JAMES WILSON

PHD CANDIDATE

SCHOOL OF CHEMICAL ENGINEERING

UNIVERSITY OF ADELAIDE

"DESIGN, DEVELOPMENT AND IMPLEMENTATION OF AN OBJECT-ORIENTED DATA MODEL FOR SMALL AND MEDIUM SIZED WINEMAKING ENTERPRISES IN AUSTRALIA."

October 2013

Table Of Contents

Table Of Figures

Tables

ABSTRACT

The management of winemaking enterprises in Australia has become complex because of the increased complexity of the market and the winemaking process itself. Accurate record keeping goes hand-in-hand with complex sequencing of processes required to efficiently operate these enterprises. Unfortunately, the record keeping aspect is often regarded as an added burden to the process rather than a necessary step within it.

The ability to link obligatory record keeping with the function of controlling the winemaking and grape-growing processes offers many benefits to the enterprise management as long as the performance of these functions is able to facilitate the control of these processes and not become the added burden they fear. The benefits come in the form of enhanced and useful records for the winemaker to truly understand the cause and effect of decision making and individual processes in their winemaking. The records capture the intent and motivation of the decisions as well as the results themselves, thereby potentially revealing the implications and relative performance of their actions.

The aim of this project was to investigate the practicality and scope of building a computer-based vineyard and winery managements system (VWMS) that is capable of fitting in and enhancing the grape-growing and winemaking process for the operator, not simply providing a means of generating reports for the tax department or AWBC auditors.

A system was developed based on extensive ethnographic research of many vineyard and winery operations as well as recorded interviews with dozens of key professionals in the Australian wine industry with vastly different roles within it. It was designed to be intuitive, thorough, and flexible enough for use by operators with very different techniques and levels of intervention in the process. It was able to take into account the relative sizes of the enterprises from the very small, up to medium sized wine making enterprises.

Several new virtual concepts were introduced to enable the data model to link and translate the real activities of winemaking into objects. These functions and processes were easily understood because they were simply formal declarations of practices that were normally carried out, but not formally named or declared in the industry.

The system was successfully piloted at a winery featured in the case studies and continues to be the primary winemaking reporting system for the enterprise.

With Thanks To…

Mr. Peter Wilson: my brother, mentor, and the inspiration for this work and for my involvement in the wine industry.

Dr Carrodus: Bailey Carrodus was my mentor’s mentor. He produced the first wine I ever tasted that actually shifted my sense of how good wine can and should be. He provided the greatest challenge to me for this thesis, and inspired me to focus on the pertinent issues; a brilliant man who is greatly missed by all those who had the great fortune to meet him.

Assoc. Pr. Brian O’Neill: My supervisor. Thanks for keeping me on the straight and narrow all the way, for both my Masters Project and this one. You allowed me to concentrate on the project and protected me from the extraneous. I am eternally grateful.

Mum & Dad: for your support, understanding, and for raising me the as the kind of person who wants to do this kind of work.

Sue, Brett , Juliana, and all my nieces and nephews. Thanks for the support, particularly while I was a reclusive, poverty stricken student.

Mrs Samantha Beck: Pure inspiration, support, and friendship. You provided the right encouragement when I needed it most.

Ms Carolyn Mooney: For your unconditional love, friendship and support.

Brett Hiemstra. Cheers Mate.

Rupert: my cat. Thanks for faithfully sitting by me for every minute I sat tapping away on my computer keyboard writing this thesis, and only requiring the occasional pat or cuddle.

DECLARATION

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution to James Alexander Wilson and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library catalogue, the Australasian Digital Theses Program (ADTP) and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

James Wilson

Date :

TERMINOLOGY AND ASSUMPTIONS

Throughout this document the term “winemaker” will be used to refer to both winemakers and viticulturists, unless otherwise stated, or unless the term “grape-grower” is also used within the context. This convention is only used for the sake of brevity, and does not imply that the role of a winemaker in a winery should be confused with a viticulturist or grape-grower in a vineyard; a concept that would offend many winemakers and viticulturists.

Similarly, the term “enterprise” is used as an all-encompassing grouping term for the individual components that comprise a winemaking or viticultural business. These components, such as a vineyard or a winery are collectively owned and managed by the enterprise or they are integral to the business.

It is also assumed throughout this document that the reader understands that this information is specific for Australian vineyards and wineries only, although it may be applicable in other countries to a greater or lesser extent. Of major concern to any management system for Australian winemakers is the Label Integrity Program (L.I.P.) which mandates the ability to be able to justify claims made on a label with respect to, amongst other things, the proportions of each wine’s source fruit for regionality, vintage, and variety. This is discussed further in *9.2.2 AWBC Label Integrity Program* (page 130). It is interesting to note that the term L.I.P. has come to mean the description of these proportions for a wine as well as its original meaning as the name of the AWBC regulation. In this context the term LIP has come to have another meaning, although closely related to its original meaning. Many winemakers refer to a wine’s L.I.P. to mean its provenance.

There is nothing sinister or alien implied by the term “compute” which has been used in this thesis as a general term for when the data model program works out a value or detail by using an algorithm, structured query language, or other relational database or programming technique to manipulate data.

1. Introduction

"Winemaking is essentially so simple....it's just that the fine print goes on forever."

Dr Bailey Carrodus, August 2008

"We must beware of needless innovations, especially when guided by logic".

Sir Winston Churchill, Reply, House of Commons, Dec. 17, 1942.

The aim of this thesis is to investigate the practicalities of designing and building a computer-based data model and associated peripheral scaffolding to provide a useful tool for winemakers to manage their enterprises. It is a practical winemaking study, not an oenological study, nor a computer software engineering study. Oenology and software engineering are important considerations for the overall success of the project, but are not the primary considerations because, as will be explained later, they tend to confuse the primary goal of the thesis. The data model needs to accommodate the oenological aspects of winemaking, and also needs to be compliant with software engineering standards. It is based on extensive consultation with various professionals within the Australian wine industry. The intention was not to ask what they want, or offer ideas as to what they should need, but rather observe and question the rationale and nature of the various tasks they perform and attempt to translate these tasks into a computer data model. This ethnographic approach is essential to making the data model relevant and powerful. The repercussions of not using such a methodology are discussed in *2.2 Software Development*

The notion of a software tool to provide such functionality may seem relatively simple because the process of grape-growing and winemaking appears to be a relatively simple concept from a scientific and technical perspective. However, it is the opinion of the author that this is a fundamentally mistaken assumption made by software engineers who have designed such applications in the past. To trivialise the process into simple steps or unit operations is to underestimate the degree of planning that goes into this process and the care and thought exercised by the professionals in this industry. It also shows a lack of understanding of the way in which winemakers approach their work. In addition, there are aspects of the management process that are often neglected by winemakers, in favour of concentrating on the core grape growing and vinification processes. By neglecting these activities, winemakers tend to resent and procrastinate performing them, leading to missed deadlines and missed opportunities. The benefits of developing an accurate and realistic model to aid winemaking are greater than simply ex post facto data control, but rather that thorough and comprehensive data control will enhance the process by making it more easily understood and perceived, as well as more efficiently and economically performed. Wines and the winemaking and grape-growing processes are ideal candidates for being modelled in a computer data construct context. Wines can have a complex ancestry after blending and re-blending. Their "behaviour" as an entity is not too remarkable to make it so complex when it comes to assigning events and reactions in an object-oriented data model, but many subtly different processes are applied and can have profound effects on the end product, at least from a subjective analytical point of view. As a result it is not necessary to simulate or model any intrinsic or abstract behavioural characteristics of wines, but rather their relationships and external processes and actions applied to them might become better understood and managed if a well founded structure was provided as a basis for their presentation and manipulation, and at the same time, provide an insight into the more subjective analyses that the wine industry seems so keen on.

From an engineering or process-flow stand-point, winemaking appears to be quite a chaotic process. The raw ingredients fed into the process have qualities that vary greatly, from subtle characteristics based on maturity of the fruit, to more basic parameters such as mass or volume. To an engineer this might seem unwieldy because such important information only becomes apparent at the last minute, thereby making adequate preparations and plans very difficult to pin-down. The same can be said for the initial processing and vinification process because these same variations have fundamental repercussions for subsequent processes. As an added complication, winemakers will often choose very different techniques, even when presented with raw ingredients with identical parameters. Clearly, a formal structure or recipe is inappropriate. It is this very variation that drives the industry. Winemakers do not wish to make a consistent or standardised product. They respond to the fruit as if it is a living entity, with personality and character all its own, and make many decisions based on wholly un-scientific principles. This is the platform from which this thesis is launched. It is an attempt to apply order, power and control of a process to the people who bear the burden of responsibility for its success.

2. JUSTIFICATION

How does one justify the need for such a system to be created? What is the knowledge gap? The answer to these questions can be summarised in three main points.

1) Managing wineries and vineyards has become more complicated over the years as markets have become more sophisticated, regulations have become more bureaucratic, and relationships between entities within and between enterprises have become more intricate.
2) There seems to be no imperative to develop a system to allow the management of such enterprises without becoming convoluted, inflexible, irrelevant or onerous for the user.
3) There is potential in capturing all the relevant data associated with winemaking to not only allow the simple recording of the process but also to enhance the process through greater insight into the machinations of the enterprise, by manipulating the data to reveal fundamental and complex relationships between apparently disparate data sources. This also becomes a matter of broadening the scope and nature of the data collected in the model.

2.1 THE COMPLEXITIES OF MANAGING A WINEMAKING ENTERPRISE

The first point has come about as commercial expectations have increased for investors in the industry, and the government has sought to ensure that the industry is responsibly managed. The actual cultivation of grapes and the vinification process has changed little over the centuries. Certainly we have come to a far greater understanding of the process on a physical and chemical level, and most advances have been on ensuring quality. This issue is more about the practical management of the processes of wine production rather than the intimate details and decisions of winemaking. There are three main forces to be reckoned with bureaucratic, economic, and logistical. Perhaps even more important is the apparent lack of skills in these areas for smaller and medium sized wineries that do not necessarily have dedicated specialists. Often these roles are performed by the same person, or at least by people whose expertise may be solely rooted in winemaking.

2.1.1 BUREAUCRACY

There are a number of state and federal laws and regulations that must be acknowledged by winemaking enterprises in Australia. In general, these laws apply to levies that must be paid as well as laws regarding labelling, exporting, and the general operations of a business. A list of most of the Federal laws to be considered follows.

- Trade Practices Act 1974
- Corporations Act 2001
- Australian Wine and Brandy Corporation Act 1980
- Australian Wine and Brandy Corporation Regulations 1981
- Australian Wine and Brandy Corporation (Annual General Meeting of the Industry) Regulations 1999
- Commonwealth Authorities and Companies Act 1997
- Primary Industries and Energy Research and Development Act 1989
- Grape and Wine Research and Development Corporation Regulations 1991
- Primary Industries (Excise) Levies Act 1999
- Primary Industries (Customs) Charges Act 1999
- Primary Industries Levies and Charges Collection Act 1991
- Primary Industries (Customs) Charges Regulations 2000
- Primary Industries (Excise) Levies Regulations 1999
- Primary Industries Levies and Charges Collection Regulations 1991

(Australian Government, Department of Agriculture, Fisheries and Forestry, http://www.daff.gov.au/agriculture-food/hort-wine/wine-policy/domestic/laws-regs, 22/9/2008)

On top of these, there may also be state laws and regulations that apply for issues such as liquor-sales licensing, occupational health and safety, building requirements, environmental protection, and waste water management. Most of these regulations require reporting of some kind, and guidelines recommend thorough record-keeping as a means of complying with them. Perhaps the most daunting prospect for a winery manager is an L.I.P. audit from the AWBC, which requires an easily auditable trail of data that justifies label claims, under section 39 of the AWBC act. This trail needs to go back for 7 years. Growers also need to maintain a spray diary which provides compliance with several policies involving environmental and water protection and pesticide codes of practice.

2.1.2 Economic Sophistication

It is clear from the anecdotal evidence collected for this report that the traditional model of a small winery that grows its own grapes, makes it own wine, and bottles and sells it within one site is rare these days. The investment required to set up such an enterprise and maintain its stability is difficult to justify, particularly given the fact that a return from investment takes several years after setting up a vineyard. It is far more common for any enterprise to consist of relationships between grape growers and wineries, even if these entities have the same owner. They are treated separately on a legal and economic basis in order to separate management and economic considerations. It is common for a winery to process fruit from its own vineyards, and also receive fruit from external vineyards to make its own wine and wine for other enterprises, under contract. These relationships cross over to such an extent that without proper record-keeping it is difficult to know who actually owns the wine and who is responsible for it. It is perhaps even more significant when one considers that winemakers often make or participate in decisions concerning harvest dates, and therefore the need for appropriate communications between disparate entities. This also means there are variable forms of income and expenses to be incurred within the enterprise that would whet the appetite of any accountant. The implications for this research and the model are that all contingencies need to be considered and that there will need to be a degree of flexibility and configurability in order to accommodate these contingencies. It also has repercussions for the scope of this project as discussed in Chapter *4. Scope*.

2.1.3 LOGISTICAL COMPLEXITY

Hand-in-hand with the expansion and sophistication of the commercial aspects of these enterprises is the logistical complexity that inevitably comes with the need to rely on the external sources of materials and services. Issues, such as the perceived glass shortage in 2007, show that a “just-in-time” approach to logistical considerations has inherent risks that can jeopardize the efficient performance of a winery, not to mention the possible benefits for bulk ordering and piece-of-mind that comes with reliable and safe ordering of crucial materials and services. Most vineyards and wineries do plan for processes that they know are inevitable for the production of fruit or bottled wine; however, there tends to be little consideration for contingencies when these plans go awry. The flow-on effects can be very damaging when one considers the number of actions that need to be sequenced to bottle a wine, not just for that wine but also for other wines in the enterprise, the reputation of the enterprise, and the loss of other opportunities. There are similar examples for vineyards, for irrigation, fertigation, pruning, dropping fruit, spraying etc. When one considers the lack of certainty of weather conditions, and the important role this plays in timing appropriate processes, one can understand how important it is to manage factors that are within the control of the vineyard manager.

2.2 SOFTWARE DEVELOPMENT

2.2.1 THE NATURE OF THE I.T. INDUSTRY

The Information Technology industry, by its very nature, tends to simplify processes into discrete steps. It is often very difficult to translate the complexities of the real-world into a data model without making the model unwieldy, or simplifying it into more fundamental and generalised steps. The problem with this approach is that a lot of the subtlety and detail can be lost in the translation, and this can defeat the purpose of the exercise. Unfortunately, this is often the norm for the I.T. industry. While playing lip-service to the needs and desires of their clients, it is notorious for changing paradigms in order to simplify the model. The resulting changes alienate the user by forcing them to change the way in which they perform their work. Computer programmers and system engineers are naturally more comfortable dealing with concepts they can understand, but it is this lack of ability to empathise with the end-users of a system that often leads to the utility of the system being compromised. The key to translating reality into a data model is to truly understand the complexity, and the rationale behind the complexity, of the process trying to be modelled.

Software tends to behave and needs to be controlled with a particular mentality. This has a lot to do with the nature of the way that humans interact with computers. Programmers tend to make programs that behave and need to be controlled in a logical procedural way, much like the way they go about their own work. We usually see an object that can have functions and procedures applied to it. Just look at Microsoft Office with programs like “Word”, “PowerPoint”, and “Access” etc with functions and utilities listed across the top on a menu.

For advanced and complex functions to be performed, the user simply needs to combine and sequence several simpler functions. This mentality seems simple and logical to IT people. Not surprisingly, this also appeals to Engineers, Accountants, and scientists in general, but when it comes to other industries or professions, this is not necessarily appropriate. Why should the program dictate how the user thinks about or manages the business? On the other hand, the logistics of designing and building a system should not be the responsibility of the end user. Simply asking the “business owner” what they want is not going to achieve an adequate result. More often than not they will describe a system that has logical flaws that they cannot perceive until the system is in front of them. Nor should it be their responsibility to provide a solution, only to provide the ethnological evidence to base it on.

Computer software programs are generic tools that need to be configured and tailored for specific uses by the user. The point is that the common ground should be confined to the industry professionals using the system, not the common ground that may or may not exist between the wine industry and the software development industry. The likelihood of there being an existing solution coming from industries or businesses outside of the wine industry is remote to non-existent, particularly given the unique nature of the process ,

2.2.2 The technical and business backgrounds of winemakers

Part of the problem for writing a system for winemakers is that they do not necessarily have any motivation for using a model if it does not improve or enhance the management process. This raises the question of what does motivate winemakers and what does a system need to take into account to overcome this barrier? It was decided to perform a straw poll to provide some anecdotal evidence about the technical backgrounds of winemakers, and their roles and aspirations in the wine industry. A brief survey was conducted on about thirty (30) Oenology and Viticulture students at the University of Adelaide. It was believed that students would be just as familiar with computers, if not more so, than professional winemakers. A copy of the questionnaire appears in Appendix 1.

There was a broad range of ages, from 19 to 54, with an average age of 26.5. Two thirds were male, and two thirds were oenology students. Just over half were 2nd year students and seven were Grad. Dip. students.

The most significant points to note were...

- 21 of 30 (70%) students aspired to management or ownership of a winery or vineyard
- Only 4 of 30 (13%) students had any background or experience in business management, economics or accounting.
- Only 4 of 30 (13%) students had any background or experience in using computers for purposes other than word processing or lecture presentations. Several students cited an open hostility to computer technology.
- All but one student cited "Love of wine" as a strong influence for their studying in the industry. For 11 (37%) it was the primary influence.
- If one applies a score of 1 for No influence, up to 5 for a primary influence, the least influential factors for students were (in ascending order) Accounting (average 1.3), Marketing (2.1) and Business Management (2.4).
- Of the 5 students that had family involvement in the industry, 4 of them cited this as the primary influence for them studying in winemaking.

In summary, this provides good anecdotal evidence to support the notion that winemakers are driven by passion for the industry more than the practical aspects of operating them as a business. This is not to say that they ignore these more pragmatic factors. In fact, it was surprising to note the high level of aspiration to management and ownership. The point is that the acceptance of a data model or a system to help manage an enterprise needs to overcome a perceived barrier to the perceived irrelevance or comprehension of such systems.

2.2.3 Literature review

Perhaps the most telling point discovered during the literature review process is the apparent dearth of published information from any of the relevant disciplines. Searches were carried out in the fields of Computer Science, Engineering, Oenology, and Viticulture as well as commercial literature. Many articles were rejected because they were irrelevant, outdated, or lacked objectivity because of their commercial nature.

Wollan, Kelly, and Baxter (2002) discuss the "WineFile" system, its development, and some of the difficulties and lessons learnt from this process, as well as future directions. This article occasionally reads like an advertisement in that *"It was developed by winemakers to satisfy the needs of other winemakers who wanted a robust, logical record system that worked the way they did."* However, the authors make many pertinent observations, particularly about the way that these systems tend to alienate winemakers from their true-calling as "artisans and creators". It also places some of the blame on winemakers themselves because of their lack of understanding about what data is necessary to record in order to be able to keep good financial records. Although this might be true in some cases, most small and medium wineries do not have the luxury of having an accountant to oversee this aspect of a winery. Nor does it seem necessary that this process should be difficult or should interfere with the way in which winemakers perform their job. The blame is unfair and it makes the author appear quite arrogant, and this seems to reinforce some of the ideas cited previously about the attitude of the I.T. industry

The article mentions that, as well as the basic functions that a management system needs to perform, it should also be a planning tool to allow the user to plan and generate work orders, and record the results. It should also be flexible and allow data to be changed retrospectively. It does not mention that the tool should be intuitive to use, which seems strange given the fact that the previous arguments seem to imply that this was the reason that winemakers have so much difficulty using the system.

The use of a double linked list in the data structure is put forward as a technical solution to the complexity of tracking the cellar operations applied to wines. Put simply, a double linked-list is a computer science concept that allows objects to be sequenced by linking them to their predecessors and antecedents in the list using linking fields so that each object points to the previous and next object in the list without having to physically arrange them in order. This is quite plainly a massive over-simplification of the issues of representing cellar operations as records in a database and does not represent a thorough appreciation of the complexity of the problem at hand. Although the use of a relational database is essential as a means of storing the data, structures within the database are not necessarily part of the solution. A data model does not and should not necessarily be reflected in the structure of the underlying database. Indeed, the benefit of using an object-oriented data model is the fact that the complexity of the interaction of the data can be contained within the model and not in the data or its structures. It is also an indication that an engineering approach has been used to attempt to capture what is, in reality, a deeply idiosyncratic phenomenon, where simplification leads to a loss of integrity and meaning for the data.

The final phase of this article discusses what needs to be done to improve these systems. Ironically, it states that "user training" is the key. This is incredibly obstinate and contradictory to statements made earlier in the article. One of the main problems the I.T industry has is its inability to empathise with its users. For a developer to say that a user needs to get used to the way the system works and adopts its own idiosyncrasies, shows arrogance and contempt for the fact that winemakers and viticulturists are consumers and customers of the software industry.

The next article (Michalewicz, Schmidt, Michalewicz and Chiriac (2007)) proposes the concept of "Adaptive Business Intelligence" software which uses optimisation and prediction techniques to convert current knowledge into process improvements in the form of recommended actions. This article highlights the fact that data is of no use if it cannot be used in a meaningful way. However, using data to feed decision making algorithms in a vineyard or winery context is problematic. The decisions made by winemakers and viticulturists are generally more a question of quality rather than efficiency or economy, and within the context of this thesis, the emphasis is more on interpretation of data that has been manipulated to provide knowledge on which to base decisions, rather than seeding the decision making process. The article, however, does expound the sort of thinking that is beneficial for this thesis. It is one of the few engineering based articles that acknowledges the need to consider the reality of the process, although it still resorts to a simplification of reality. This stance is justified because of the complexity and nature of the case study it employs and the outcome it seeks. A prediction and decision based on this is very different to a need to just represent the data in a meaningful way.

The E. Allen Gleazer (1990) article supported the use of computer software as a potentially useful tool in the vineyard or winery. This was at a time when the I.T industry that we know today was in its infancy. The now ubiquitous PC was just becoming common, but user graphical user interfaces were a relative novelty. He posed the following questions...

Can you use computer software to 'faithfully support your intentions without introducing difficulties?'

What are the benefits of software which has this kind of quality?

How can you find or create this kind of software?

He then described an example of a software program and then sought to answer these questions. Although many of the answers he came up with are factors we now take for granted, such as speed of processing, and the need for the software to be customised to the industry and the enterprise, it is refreshing to note that he believed the solution was to have software compatible with the way the core business of the enterprise is performed, and that the results ought to be understandable and provide illumination of the situation to help make decisions.

An article entitled “Extending Ubiquitous Computing to Vineyards” (Burrell, Brooke and Beckwith (2003)) introduced two new concepts for consideration. The first term was “Ubiquitous Computing” in which software seamlessly interacts with other distinct networked devices. This will need to be considered in the final data model because it is common in modern wineries for PLC control of devices such as heating or cooling mechanisms, providing yet another source of data. Secondly, and probably more relevant, is the use of “ethnographic” techniques to design software. Ethnography in this context is the use of fieldwork observation of the various roles and activities in an enterprise to create a holistic description of a system. In conjunction with an interview process to get the participants’ opinions and philosophies, the authors were able to then go off and design a system based on these results. Key to this process was the observation that the participants were able to tell them what data was important, how the data was arranged, and how the data was associated with the activities and roles in the process. In this way they were able to not only define the process but also work out how the data should be used. In conclusion, the authors developed three guidelines.

- The need to consider not just gathering data but how it will make a difference to users
- Understand the work practices in question and design the interface to support users now and in the future.
- Tailor the system for the various users so that they are only required to input data commensurate with the data it returns to them.

Iland et al. (2004) emphasise the need for statistical analysis and multivariate analysis as a tool for predicting wine style and quality from pre-harvest parameters. They mention that it is necessary to carry out these analyses on the enterprise’s own data to <u>find</u> relationships rather than relying on any industry standard algorithms applying to this data. In addition, it is clear that subjective data like this need to stored and used, and is just as valid as more objectively measured parameters.

In summary, the literature review provided more inspiration to perform this thesis. It illustrated that there was a need for a system to be developed that took the core business at heart, and that it should provide a tool to manage all aspects of the operations, including the functions that have so far been neglected by existing systems, such as planning and data mining.

2.3 DATA MINING AND THE POTENTIAL FOR ENHANCED MANAGEMENT TECHNIQUES

The third major justification for this thesis is the potential advancement of a tool to provide enhanced insight into the management of wineries and vineyards. Any self-respecting winemaker or viticulturist will always have an eye out to improve their process. Without exception, there are always situations that occur in the vineyard or winery that cause the manager to question their practices, whether they are prepared to admit this openly or not. When these situations occur, the first instinct is to find the cause so that the situation does not occur again. The corollary to this is that sometimes an unforeseen situation occurs that leads to an outcome more beneficial than expected. Whichever the case, there is a potential for the real cause to be hidden or discovered somewhere in the data associated with the wine.

Of particular interest is that there may be benefits in using data specific to the winery or vineyard instead of relying on industry-wide algorithms to predict outcomes. “Terroir” is often the catch-cry used in the industry to explain the different characteristics in fruit and wines. But what about the difference between wines within a given region, or even between different blocks within a vineyard? And what about variations between the vintages of wines produced from the same block? It is common to see vintage summaries produced by industry groups or wine writers that say “Vintage X was a particularly hot vintage” or “Rain in February led to a later vintage which produced wines with more complex and softer tannins.” This is all well and good, but it does nothing to help a winemaker explain or predict outcomes for their enterprise; why one wine took longer to ferment than it did the previous year or perhaps even more importantly, help to predict how long a wine might take to ferment this vintage. For example, if rainfall is an important factor in predicting harvest date, then it is no good using state-wide, or even region-wide rainfall statistics to help in this process. What about irrigation? What about drainage? What about soil condition, humidity or mulching? Localised phenomena and practices are the key to understanding the cause and effect of the parameters used to measure the success or failure of an enterprise. Without a precise measurement and understanding of these variables, the winemaker cannot plan or comprehend likely outcomes, make decisions and justify winemaking techniques or know or understand the effect of their decisions. The aim of this thesis is to investigate whether a tool to manage data associated with these variables will provide meaningful and insightful information to aid winemakers rather than can be achieved by intuition, self belief, blind faith, passive acceptance, or alchemy.

This is, of course, most important for small and medium sized enterprises which tend to be more experimental, proactive, flexible, and susceptible to external influence.

3. Structure of Development

This thesis draws together information gleaned from ethnographic research of several case studies , interviews with professionals within the industry, previous research done by the author in a Masters project, common business models and structures, common data modelling tools and techniques, data collected from seven years of records from one of the key case studies, and perhaps most importantly, the development of several concepts that are new to the industry in the sense that they have never been formally proposed and structurally defined as tools in the winemaking industry. It is important to present this information carefully in order to make sure that all relevant information is processed in the correct manner and presented logically and thoroughly. It is proposed that the following structure is used.

1. Scope out the scale and nature of the project with appropriate terms of reference.
2. Define the commonly known data modelling and application software development issues and how they affect the proposed development.
3. Define the commonly known models and techniques currently used in the industry.
4. Create a general, high level model of the industry based on these concepts and previous research.
5. Introduce findings from interviews, case studies, and ethnographic research.
6. Introduce some new concepts to address perceived short-comings in the model.
7. Use this information to build a new model.
8. Build the model and populate with data from key case studies.
9. Test the model to prove the validity of hypotheses and new concepts.

Any kind of software development must be performed in a logical and careful sequence and there are industry standards that attempt to ensure the integrity of the process. In essence, these standards impose a top down design phase, followed by a bottom-up implementation. The design phase involves gradually defining more and more detail, whereas the implementation involves building up functionality from simple functions and objects into greater intricacy and encapsulation. Each step in the structure defined above is based loosely on these basic standards. However, throughout this process, the emphasis will be on ensuring that there is no compromise of functionality for the potential beneficiaries of this model just because the computer-software application paradigm demands that the system needs to adhere to I.T. industry standards.

4. Scope

4.1 Scale

First and foremost, this thesis is aimed at the smaller end of the scale of wine industry enterprises, from micro wineries and vineyards up to medium scale wineries that might have an annual crush of 1000 tonnes or vineyards of up to 300 acres. The choice here is arbitrary, but the rationale is that enterprises of this size have very different issues compared to larger enterprises. They are usually owner operated, and employ only up to a dozen people at peak, and perhaps only 4 or 5 staff off-vintage. As a result, staff members tend to have several roles and responsibilities. Laboratory facilities tend to be minimal and usually there is only one specialised laboratory technician. It is quite safe to say that very few if any of these enterprises will have an I.T. department or even an I.T. specialist. However, with the advent of advanced computer networking and internet access, it is not deemed necessary to restrict an enterprise to any geographical proximity boundaries.

4.2 Multiple entities within each enterprise

The model will not be restricted to assuming the enterprise consists of one vineyard and one winery. It may be one or more vineyards and/or one or more wineries. The repercussions for the model here is that it will need to be configurable so that it does not assume that because there is data for a vineyard, there will also be data for an associated winery.

4.3 Functions

The model should confine itself to tasks concerned with the functional operations of the enterprise, and not necessarily provide a full functional accounting system. There are a plethora of accounting systems available on the market, and so the data model and associated scaffolding need only provide an interface to commonly available accounting systems rather than incorporate these functions internally. This does not mean it should exclude accounting information. An important function of any winemaker is to be aware of the costs involved in decision making. The demarcation and sharing of information between the two systems is very important. Duplication and synchronisation of data will need to be considered.

4.4 Sophistication

Towards the smaller end of the scale spectrum, there is great diversity with respect to the levels of sophistication and intervention employed in grape growing and particularly winemaking. The model should not assume that any particular functions are mandatory. Similarly, the emergence of new technologies should be easily added to the model by the user, as individual enterprises become more sophisticated over time and benefit from economies of scale with increased automation and/or improved equipment and techniques.

5. Data modelling and some basic design principles, independent of user specifications

This chapter is primarily concerned with establishing some architecture and design guidelines on which the data model is built. A data model, in this context, can be described as an n-tiered system of programmed objects.(see Figure 1) Each layer functions independently of the other, has fundamental structural differences and roles, and serves as a means of separating responsibility for the behaviour, function, and appearance of the data model. These decisions and principles do not impact on the core issues of this thesis, but simply act as a scaffold for the data model to reside.

5.1 Architecture

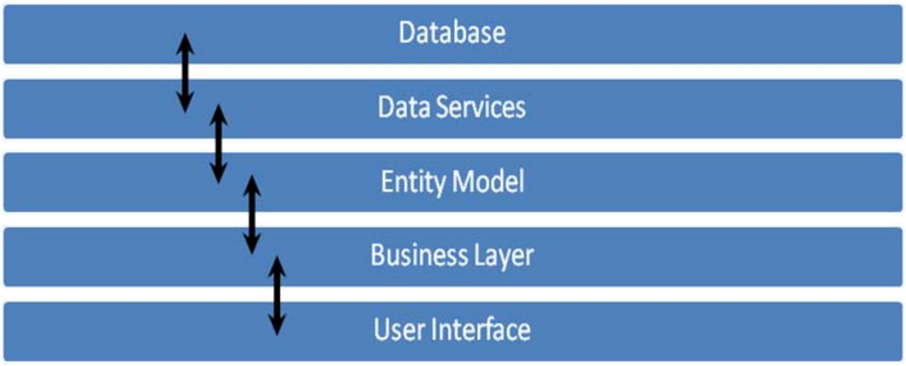


Figure 1. Standard n-tier architecture

The Database is simply a Microsoft Access database, used to store and retrieve the data for the model. Its structure is heavily influenced but not determined by structures and entities in the Entity model. As noted earlier, it is important that the database structure avoids over complication in order to provide efficient performance. It is generally accepted that relationships and structures in the database avoid going beyond third normal form, thus forcing complex relationships into the entity model.

The Data Services layer is used as a means of collecting all the selection and editing structured query language (SQL) into a single component, to avoid any dependency of the entity model on structures within the database.

The entity model uses object-oriented concepts and structures to contain the data from the database to provide virtual entities with properties, methods and behaviours analogous to the real-world objects they represent. These objects may be actual objects like tanks, wines, people, or abstract objects like plans, vinification processes, and timetables. By carefully designing the entity model, it is able to retrieve data from the database and provide populated instances of these objects for manipulation by the business layer. The entity model will be built using classes in Microsoft Visual Basic version 6 and incorporated into an ActiveX module. The choice of Visual Basic 6.0 is based purely on availability, stability, and familiarity.

The Business Layer will also be an ActiveX component written in Visual Basic. It is responsible for providing business logic and business related rules to manipulate data objects defined in the entity model and present data to the User interface. The demarcation between business logic and entity behaviour needs to be carefully designed to ensure that behaviours and parameters specific and inherent in an object remain in the entity model, and that the business layer is responsible only for providing interpreting requests from the user interface.

The user interface is just used to provide user access to the model. Rules about how it behaves can be partly hardcoded into the interface, but also have restrictions and behavioural rules applied to it by the business layer. This includes aspects of disabling certain functions based on the status of entities or user permissions.

5.2 RECORD LOCKING

When a system allows concurrent access by multiple users, it is necessary to have a strategy to make sure that data is not edited by more than one user at a time. Record locking tools are provided by database systems, but this causes complications for n-tier systems which need to bubble these changes through various layers to maintain data integrity. It has been decided to provide higher level record locking by employing user-defined system sector locks. When a user enters a certain area of the interface that allows the user to modify the data, subject to user permissions, a flag will be set so that subsequent access to that data by other uses will be on a read-only basis until the original user exits that data area. Deciding what data records need to be protected cannot be prescribed before the database structures are designed. It is more than likely that some data, particularly reference data, will require many tables or records to be locked, so this comment will require clarification once the database and entity model have been designed. It seems unlikely that this will have severe impact on the performance of the system given the scale of the enterprises specified and the number of users likely to be using the system at any given time, This locking mechanism also allows the possibility of notification of other users when a record is updated so that information can be automatically updated on the screens of other users without requiring the user to refresh the data manually.

5.3 PERMISSIONS

A hierarchy of permissions that reflect the responsibilities for decision making can be used to allow or restrict access or editing rights to the data model. Typically, these permissions are...

- No Access
- Read Only
- Read and Write

The permissions granted will depend on the role the user has in the organisation and the need to allow that user access to the specific data. These rules typically include...

- Winemakers
- Assistant Winemakers
- Winery Owners
- Marketing Personnel
- Accountants.
- Legal and Industry Bodies
- Cellar Managers
- Team or Shift Managers
- Cellar Hands
- Laboratory staff
- System and Network Administrators

(Wilson, “Winery Management System”, 2006)

It should be noted that these roles will provide an important starting point for consideration of the design of the user interface. In small enterprises, it is common for an individual to perform many if not all of these roles. The User Interface will be responsible for acting on the permissions by enabling or disabling fields or forms throughout the system, based on functional calls made to the business layer, which will in turn be based on entity settings in the entity model, populated by data from the database.

5.4 AUDIT LOGGING

Every time a significant event occurs in the system, an entry will appear in an audit log table in the database. This will allow error tracking as well as verification of functionality during the testing phase.

5.5 DATA TYPES

Apart from the usual data types used in Visual Basic, the data falls into 4 categories based on their source and their nature.

Fundamental data are the objectively recorded data from the vineyard or winery. Rainfall, Sampling, Harvest Dates, Weighbridge dockets, Vessels, wines, operations, analyses are all directly measurable empirical data.

Subjective data are the opinions, observations and interpretations made subjectively by the user such as tasting notes, wines scores, taste, flavour, aroma, appearance. This kind of data will rely heavily on industry standard scales which may or may not relate to fundamental data types. However, the prospect of linking and interpreting the aesthetic aspects of winemaking with the empirical data is something that might be useful to investigate within the data mining component of this project. For example, sample tasting during ferment could indicate the onset of malolactic fermentation (MLF), even though a wine is not expected to go through MLF during primary fermentation, and therefore laboratory tests have not been scheduled to detect this. Such observations show a potential means of linking such subjective sources of information to fundamental data types such as lab testing or other operations.

Strategic data are the data concerned with planning and contingencies, such as marketing, sales and vintage plans, as well as vinification flowcharts, and any other plans or strategies. It also includes information about justifying decisions made in the winemaking process. This aspect of winemaking is similar to the fundamental data, but is more to do with the aspirational and motivational aspects of the enterprise. Strategic data is important in linking the business cycles of the enterprise to the operations. By associating strategic plans and decisions with other forms of data we are able to collect a flow of data that links cause and effect.

Configuration data is data associated with fixed but modifiable parameters that describe the specifics of the enterprise. These are commonly called code-sets and will allow the model to be data-driven for preferences and default actions and parameters. It is expected that these tables will be set once during the configuration process, and rarely modified, only when new instances of users or equipment are introduced. They also offer a means of controlling the sophistication of the system as the user gains experience and familiarity with the model, and as the enterprise expands and becomes more sophisticated over time.

Data may be sourced from several places, regardless of its type.

- Direct input from users
- External systems
- External sources
- Internally generated by the model using internal or user defined algorithms.

The distinction between these data types is purely technical and descriptive. In fact, throughout the case study phase of this project (*7, Case Studies*) it was interesting to notice a lack of distinction between subjective and objective information, and descriptive tasting notes were often given as much credence and significance as any other measurable parameter, if not more. This can be regarded as beneficial for the sake of this project because it tends to solidify the perception of relationships between causes and effects as perceived and described by the winemaker. For example, if the intention of an action (say acid addition) is to impart a subjective change in a wine (which is quite common), then it makes perfect sense to "measure" and judge the effect in subjective terms. This may be perceived as spurious in scientific or engineering environments, but these actions are not scientific or engineering issues but rather part of the human aspect of the industry. This is discussed further in *8.2.3 Subjective data*.

6. A HIGH LEVEL DATA MODEL AS A STARTING POINT

There are aspects of the winemaking industry that are well understood, and just like any other agricultural process there is a cyclical and seasonal nature to such enterprises. However, winemaking can be distinguished from other processes because there is an extended period between the growing and harvesting steps and the final packaging process, which may take anywhere between a month and several years. Accompanying this is the complex nature of the planning process, the variations in the process definitions for the different products, and the need to deal with different products at different stages in their development at any particular time in the year. Above all, there is great emphasis placed on “vintage”, the harvest and vinification process typically occurring between February and May each year, because of the workload and strategic importance.

The diagram below represents a simple, high level, data flow model for a typical winemaking enterprise, consisting of a single vineyard and winery. It is worth noting here that for enterprises consisting of just a winery or just a vineyard, or multiple vineyards or wineries, the model is just as applicable.

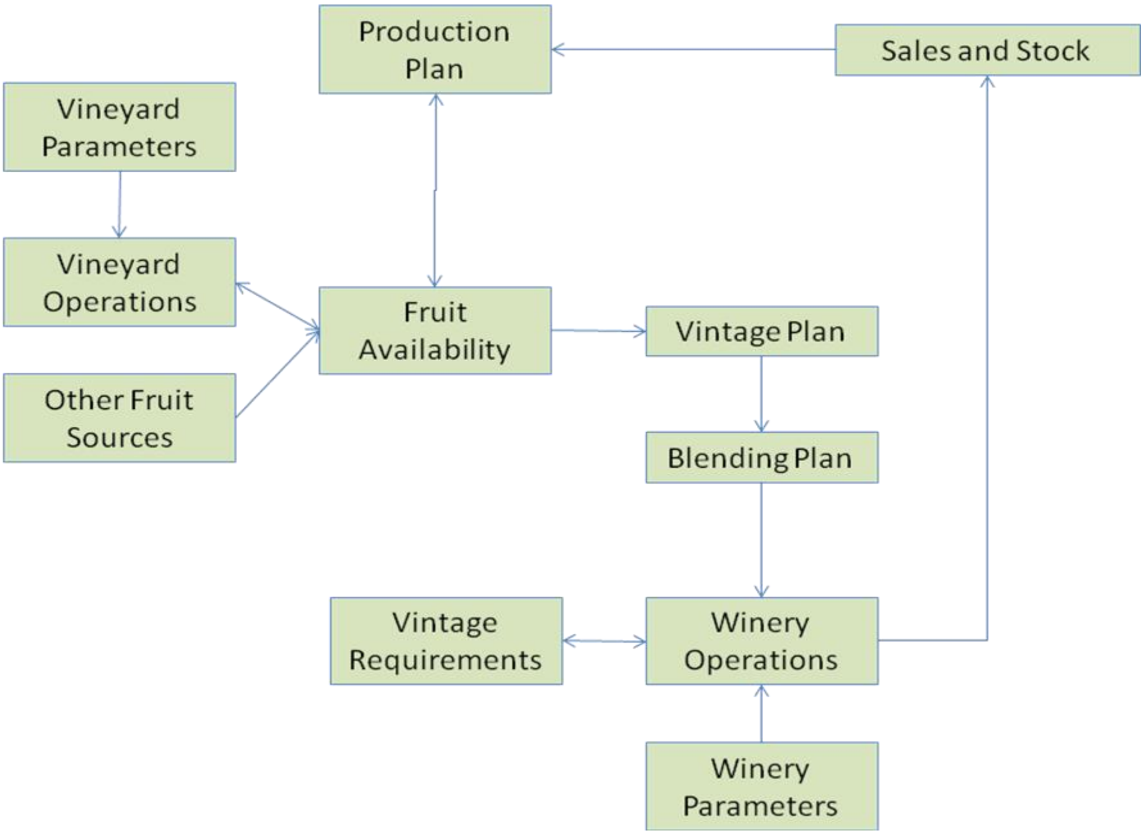


Figure 2. A simple high level data flow model

This plan is based on early discussions held with several of the professionals referenced throughout this document and is an amalgamation of many versions that came about because of the different business models used by the various enterprises. It serves as a useful starting point for the model because it is close to what the winemakers themselves see as the essence of the model. Each element is a general area or concept that links with others to describe the operational cycle.

6.1 PRODUCTION PLAN

The Production Plan sets the whole process off. It combines parts of what might be considered a Sales and Marketing plan, along with prerequisite quality parameters concerned with winemaking. It describes the intended end products, their proposed volume, packaging, production cost, and intended market. As stated earlier, it is not intended to provide financial functionality as this is better served in financial packages commonly used in the industry. However, this information may well be useful for this model because it is important to be able to track costs as they are incurred and ensure the targets are met. This highlights the need to provide appropriate interfacing with financial packages, although this will be optional. This plan provides seeding information to establish the fruit required for vintage.

6.2 Fruit Availability

The production plan implicitly indicates the quantity and quality of fruit required for the vintage. However, the fruit available from the enterprises own vineyards may not necessarily fulfil the requirements. Some wineries also do contract work and process other sources of fruit for themselves or others. The Fruit Availability component is used to summarise the fruit required and its sources, and reconcile the Production Plan with the Vintage Plan. You will note that there is a two-way flow between the Production Plan and the Fruit availability, reflecting a need for negotiation between the two.

6.3 Vineyard Operations

This entity encapsulates all the operations occurring in a vineyard. It responds to and imposes upon the requirements of the Production Plan manifest in the Fruit Availability. The operation of a vineyard is usually the domain of a viticulturist who needs to ensure the health and viability of the vines under their control. This is a year-round task, involving pruning, spraying, irrigation, fertigation, sampling, staff management, materials management, equipment management, as well as the need to ensure that elements of the production plan are met. Viticulturists tend to think in terms of blocks of vines, and occasionally break down these blocks into particular sub-blocks for special treatment, whether this is due to issues of vigour, disease, damage, topography, or environment.

6.4 Vintage Plan

A Vintage Plan describes process definition details of how each batch of fruit is to be treated, with the aim of producing the wines required for the production plan. In a sense it is more than just what the winemaker wants to happen during the vintage, which is typically defined as the period from the receipt of the first batch of fruit until the last ferment is complete, but rather it details the ongoing processes involved until the wine is ready for blending. Obviously, things get a little complicated when we start to consider blending issues and the use of an individual batch in more than one finished product, and this is discussed later. Each process definition considers many factors at all stages of the process including...

- Crush/destem/whole bunch press
- Extraction
- Fermentor allocation

- Yeast
- Sulphur Dioxide regime
- Heating/cooling
- Additives
- Maturation
- Fining

The aim of the vintage plan is to predefine, as much as possible, the ferments to be done to ensure that there is sufficient winery capacity, materials, and labour (Vintage Requisites).

6.5 Blending Plan

The Blending Plan works in concert with the Vintage Plan to define the processes required to produce the wines for the Production Plan. The Blending Plan takes the wines produced from the ferments and allows the user to specify various scenarios for blending these wines into the final products specified in the Production Plan. The reason they are separated is that blending decisions are often made much later in the process. There may be several batches of a particular variety, which could be used in any of the wines requiring that variety, but the differences between batches will not be known until they are produced and can be tested and tasted. It is also commonly regarded as quite a different and distinct process deserving quite separate attention from the more pragmatic role of vintage. It is a time when winemakers feel back in control of the process and can make careful and considered decisions without the pressure of external factors. It is worth noting that this precludes co-ferments which obviously need to be organised before vintage.

6.6 Winery Operations

This is more than likely the most complex aspect of this project, given the extraordinary diversity in the way wineries are constructed, staffed and managed. Winery Operations covers all aspects of recording actions and events in the winery. This includes cellar work such as crushing, pressing, filtering, transferring, racking, additions, barrel work etc., as well as laboratory testing, tasting and fining trials, bottling and packaging, and other processes too numerous to mention in this summary.

6.7 Sales and Bottled Stock

Once the wine is packaged it is usually kept in a storage facility prior to sale. Rather than providing a sales system for “point of sale” (such systems are well established and outside the scope of this project), the sales and stock component of this model is required only as a means of keeping track of the whereabouts and quantity stock still within the control of the enterprise. This becomes useful for creating subsequent production plans, to provide evidence of how well products are being sold, and the impact current stocks have on setting targets for subsequent vintages.

So, as simple as this model is, it illustrates the way data flows around and represents a summarised version of the inner workings of such enterprises. The aim now is to fill in the details, and attempt to convert this into a realistic entity model. It is certainly true that the devil is in the detail. One only needs to visit a few enterprises to realise that there are more exceptions than rules.

7. Case Studies

Given the basic model cited previously, we can now turn attention to developing it and enhancing it based on real-world, ethnological studies. Throughout, this project, I was fortunate to consult with several professional winemakers and viticulturists who provided a wonderful insight into their own particular perspective of this industry. They came from very different enterprises, ranging from the manager of a single vineyard supplying fruit to a couple of wineries, up to much larger enterprises with a single winery processing fruit for some thirty (30) growers. What follows is a selection of the most significant case studies that have provided the bulk of the ethnological evidence used in this thesis. The first case study provides the bulk of the common aspects of grape growing and winemaking enterprises. To avoid repetition, only unique aspects of each of the subsequent enterprises are highlighted. The description of these case studies is used to simply report the ethnological evidence. The analysis and implications for the data model appear in italics, in separate boxes throughout the descriptions. A more complete summary of these thoughts is presented later.

7.1 Case Study 1 – Medium sized winery with two vineyards

This company stands out in many respects as a classic medium sized winery within the Australian context, and is probably close to the upper end of the spectrum within the scope of this thesis. It is equipped with all of the common facilities found in a modern winery, and exists in the market place as both a domestic producer and exporter, with five (5) separate labels targeted at different price-points within both domestic and export markets. It operates two vineyards, one in the Yarra Valley and one in the Heathcote region of Victoria, and a winery adjacent to the Heathcote vineyard.

7.1.1 The Winery

The winery is capable of small batch processing of 1 tonne, up to fermentors of 30 tonne capacity. It annually processes about 1000 tonnes of fruit from not just its own vineyards, but also does contract work for other local winemakers. The general philosophy of the winery is to allow gravity feeds of ferments from raised fermentors down into a mobile press. This allows some space saving because the area under the fermentors is used for storage tanks. All major fermentors in the winery are stacked on top of storage tanks. As a result, there is a great deal of raised walkways throughout the facility. There is a general flow in the design, with the fruit reception area at one end of the winery, the fermentors and main operations in the middle, and the barrel hall and bottling facilities at the other end.

7.1.1.1 Laboratory

The modern laboratory (Figure 3) is capable of performing all common analyses, using modern equipment such as a spectrophotometer, turbidity meter, Dissolved Oxygen meter, and the ubiquitous aspirator for sulphur dioxide analyses. It also accommodates a tasting area.

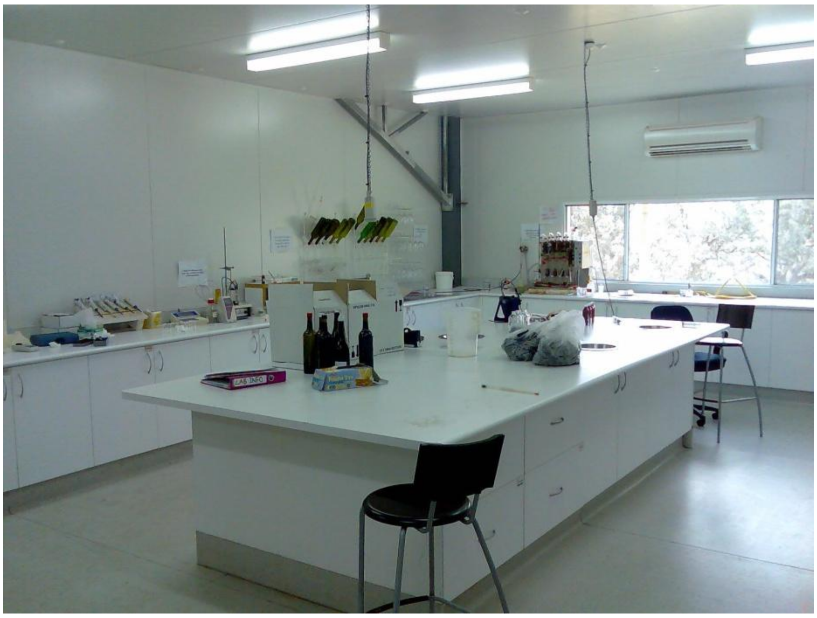


Figure 3. The Laboratory at Whitebox Winery

7.1.1.2 Forklifts

There are two forklifts and a scissor lift for use throughout the winery. A large diesel powered forklift is used to lift large 2.5 tonne fruit bins into the reception bin, and a smaller LPG powered forklift with various special attachments is used for tipping smaller 500kg bins and general pallet work throughout the winery. The scissor lift is used for high level maintenance work on the tanks

7.1.1.3 Fruit Reception

The fruit reception facility (Figure 4) consists of a weighbridge, 25 tonne reception bin, and crusher/destemmer with a must bin and must pump connected to 6 inch stainless steel hard lines to pump directly into any of the fermentors. White fruit can be diverted into a large membrane press that can then direct the juice straight into any of the storage tanks for fermentation. The Press rides on rails and can be positioned under any of the fermentors to press off any red ferments. The reception bin is capable of receiving 25 tonnes of fruit so that a B-Double transporter can be emptied completely and sent on its way, without the fruit needing to be processed immediately, although this has never happened in practice. The must tank underneath the crusher/destemmer is open so that additions of yeast, Potassium Metabisulphite (PMS), acid, enzymes, or other additives can be made prior to pumping to the fermentor or the press. The pumping action of the helical must pump provides added mixing to any additives to ensure reasonable distribution of the additive in the must.

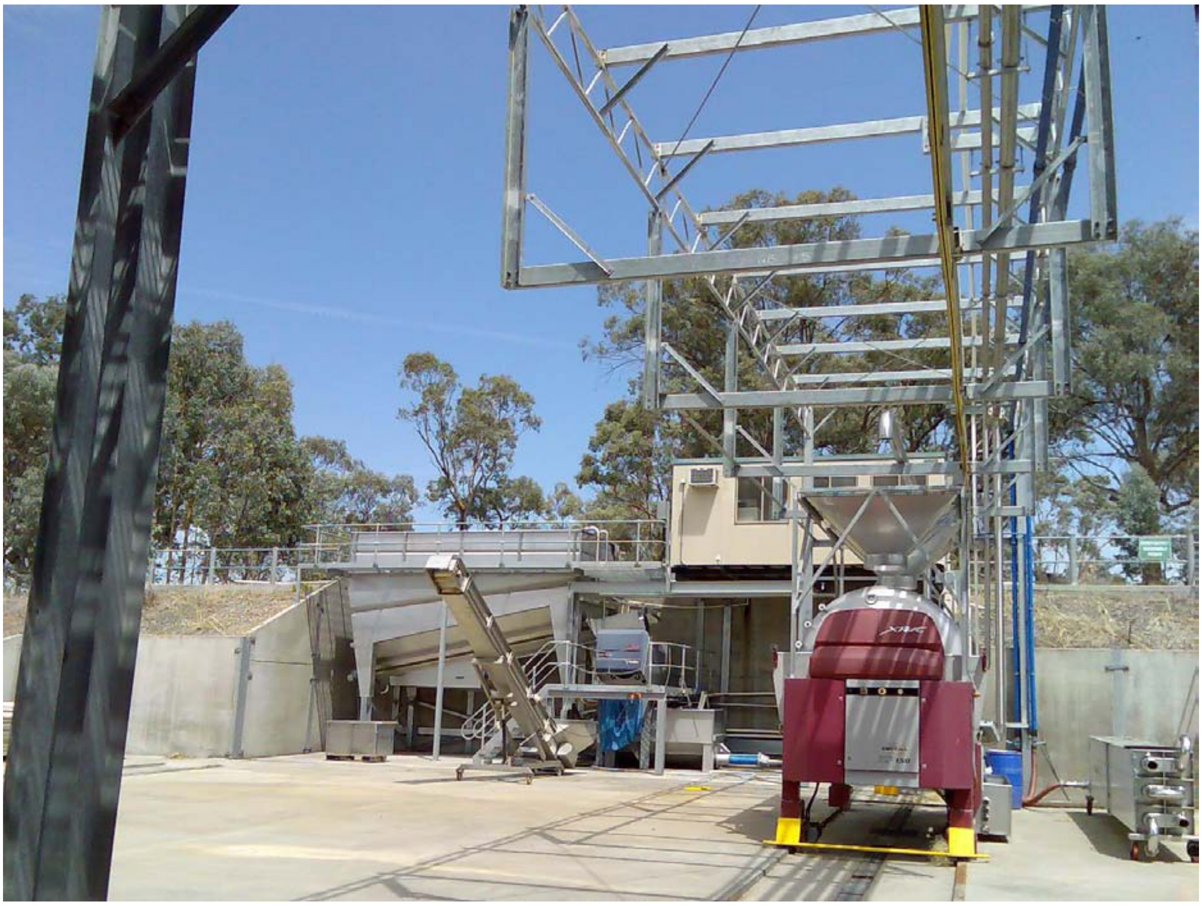


Figure 4. Fruit Reception at Whitebox Winery

7.1.1.4 Tanks

The following two tables describe the basic parameters for the fermentors in the main winery and the storage tanks in the soc-called "tank farm". (Figure 5)

Type	Capacity	Number
Open	12 tonnes	18
Open (Variable Capacity Lid)	6 tonne	5
Open (Variable Capacity Lid)	4 tonne (4.5kL storage)	1
Open (Variable Capacity Lid)	3 tonne (3.5kL storage)	3
Open (Variable Capacity Lid)	1 tonne (1.1 kL storage)	3
Static	25 tonnes (32kL storage)	6
Ganymedes	25 tonnes (32kL storage)	4

Table 1. Summary of fermentors at Whitebox WInery

Capacity	Number
92kL	4
46kL	4
37kL	10
27kL	18
15.3kL	12
5.3kL	4

Table 2. Summary of storage tanks at Whitebox WInery

The 37 kL and 27 kL storage tanks are the bases for the Static and Ganymedes fermentors and the Open fermentors, respectively. As a result, they have no lids, and employ side pots to provide access for dip readings and breathing functions during transfers in and out. The open, static, and Ganymede fermentors have pneumatically controlled tun doors to aid opening these heavy doors when wines are being pressed. It is common for harvesting bins to also be used for fermenting smaller batches of fruit up to 3 tonnes. These bins are manoeuvrable, mobile and easily plunged manually.

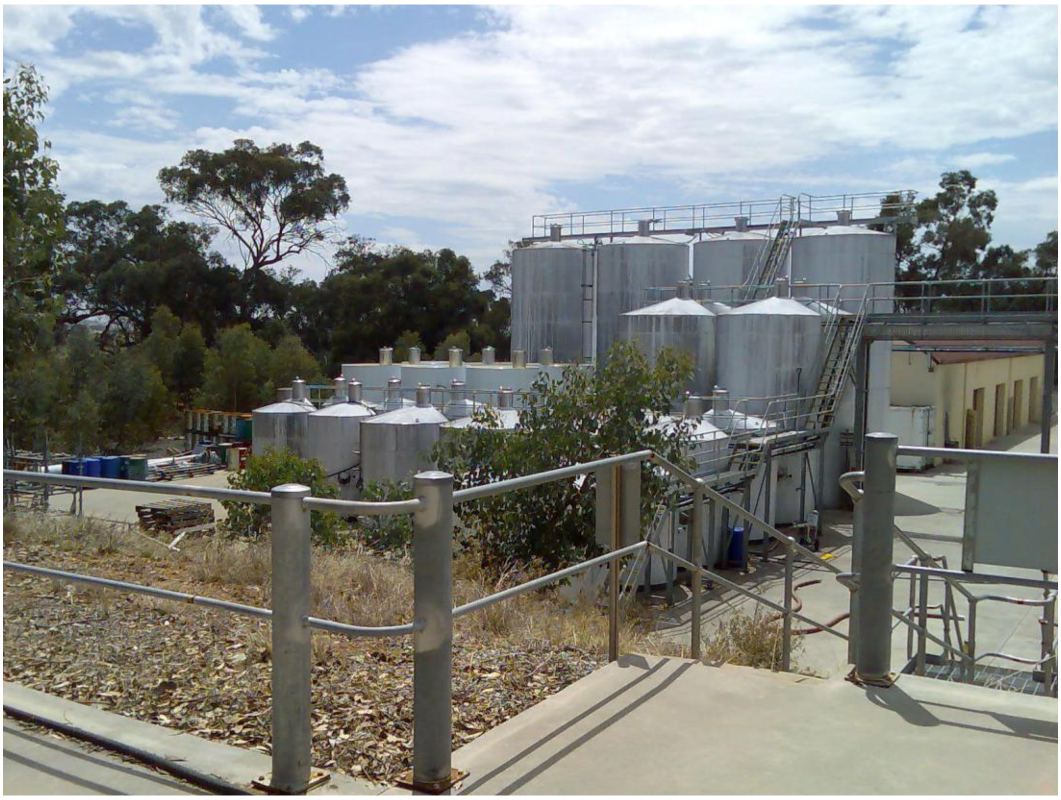


Figure 5. "The Tank Farm". External storage tanks at Whitebox Winery

7.1.1.5 STAFF

The winery is staffed by a Chief Winemaker, Senior Winemaker, and Assistant Winemaker, and a Laboratory Technician. During vintage this is expanded with 6 or 7 extra vintage staff. They typically operate in two shifts; a day shift from 6 am to 4pm, and a night shift from 4pm to 2 am. This means the winery is staffed for 20 hours per day, ensuring that ongoing processes are uninterrupted as much as possible. There are always at least two OH&S trained staff members with first aid certification and at least two licensed forklift drivers working at any one time.

7.1.1.6 Barrel Hall

There is a large insulated barrel room within the winery, capable of holding about 2000 barriques on racks and maintained within an appropriate range of humidity and temperature. The barrel racks are arranged in rows and their positioning is recorded on a whiteboard within the barrel hall. Each barrel is labelled with the name of the wine and all barrels are kept together in the one area. Each batch has a mobile 200 litre variable capacity stainless steel topping tank to allow top up of the barrels periodically. When this is empty, another barrel is transferred into topping tank and is used to maintain the levels in the barrels. The barrel hall also doubles as a storage area for bottled wine.

7.1.1.7 Filtration

The winery is equipped with a small but sophisticated cross-flow filter (Figure 6) enabling the filtration of wine to be performed relatively unattended once it has been set up for a particular job. It is programmable and self cleaning and performs regular back-flush operations to ensure the filtration process continues reliably and automatically.

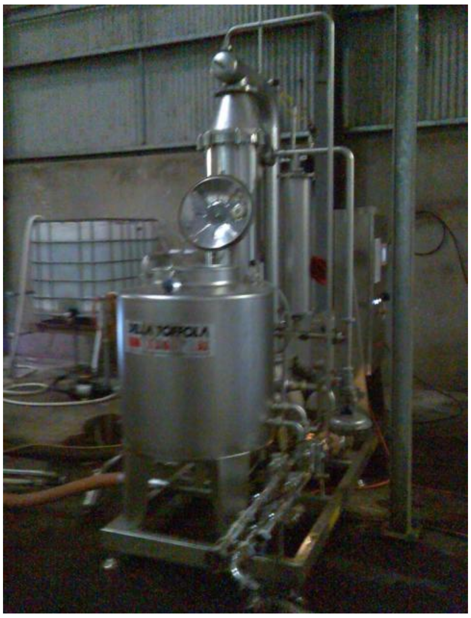


Figure 6. Cross-flow filter at Whitebox Winery

7.1.1.8 Presses and Pumps

Large ferments or white fruit batches are pressed in a large membrane press that can take up to 20 tonnes of fruit or 30 tonnes of must at a time. The wine or juice comes out into a 500 litre open buffer tank allowing tasting and additions to be made prior to pumping to the storage tank. It also allows high and low float detectors to be used to automatically trigger the pump used to transfer the wine or must from the buffer tank to the target tank. The must/ferment can be fed in through open doors on top of the press tank, or through an axial feed at the end of the tank. The axial feed and buffer tank are also used to recirculate during cleaning and rinsing the press.

There are also smaller scale basket presses used for batches under 5 tonnes, although for border line cases, the larger press can be used by simply keeping the skins from a previous large pressing in the press.

There are several pumps available for use in the winery, both centrifugal and positive displacement types, and the choice of pump depends on the task at hand. Positive displacement pumps are generally slower but more controllable and reversible, which makes them ideal for barrel work. Centrifugal pumps are faster and can be opened for free flow, but pump only in one direction. A flow meter is also available and can be put in line to measure flow rate and volume very accurately. This is always used during racking processes when it is important to have an accurate reading of losses due to lees, and also when the volume of wine needs to be re-measured after filtering so that subsequent additions can be made at appropriate levels.

7.1.1.9 Micro-Oxygenation (MOX)

The use of micro-oxygenation is now commonplace in many medium and large wineries. It involves the slow introduction of oxygen to the wine via a sinter dangled into the tank just above the lees layer. Oxygen is dissolved directly into the wine rather than bubbled through it, and it is believed that this enhances tannin polymerisation. Any tank within the winery can be connected to the MOX apparatus but only 6 tanks can be connected at any one time.

7.1.1.10 Heating and Cooling

Effective and timely temperature control is paramount in any commercial winery, particularly during fermentation. This winery has most of the tanks connected to an automated temperature controlling system that controls the flow of refrigerated “brine” through the cooling jackets incorporated on the tanks. (Figure 7). They are also thermally insulated with polystyrene and aluminium jackets covering the whole of the tank. A temperature probe on each tank sends the temperature to the controlling system which then automatically opens or closes a valve to allow or stop the flow. Heating blankets are attached to some of the tanks (Figure 8) to allow controlled heating using the same automated system. The winemaker can set a range of acceptable temperatures for a wine depending on the phase of the production cycle. The smaller scale variable capacity fermentors are not connected to temperature control, nor are they insulated. However, their smaller size means that cooling is not usually necessary and heating can be provided using a heating plate.

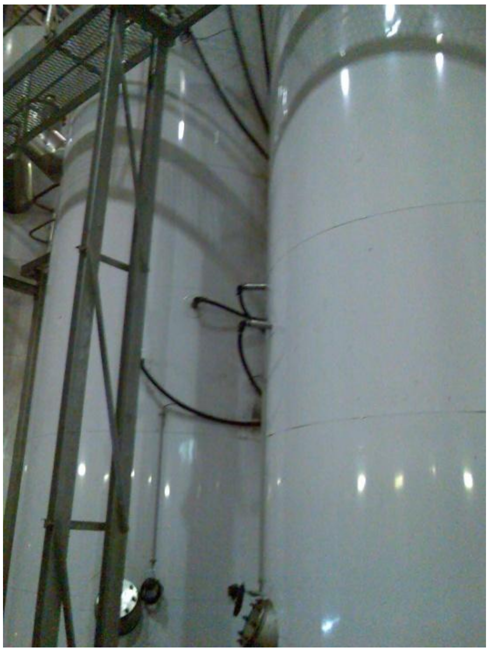


Figure 7. Cooling lines attached to vessels at Whitebox Winery

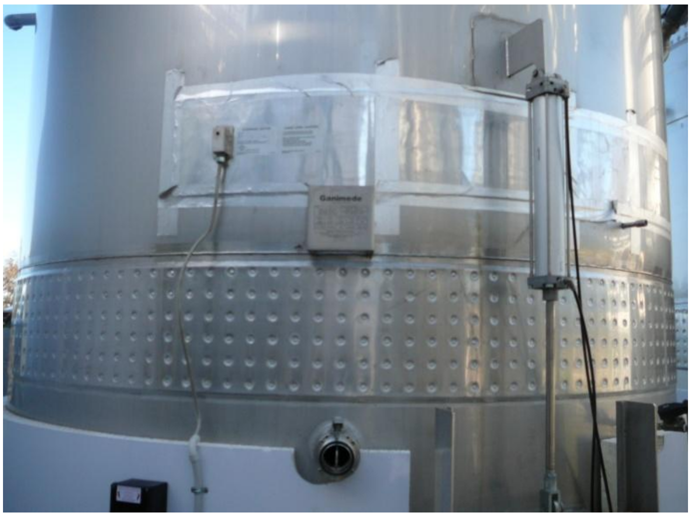


Figure 8. Heating Pad installed before bring covered by insulation at Whitebox Winery

7.1.1.11 Water in the Winery

Clean water, both ambient and hot, is available throughout the winery. The hot water is heated to approximately 70°C which is sufficient to provide sanitising capabilities. Water is required not just for cleaning but also for transferring wine and juice through the winery. A “plug” of water is sent immediately in front and immediately behind wine or juice being transferred through hoses or lines to provide a barrier to exposure to the atmosphere. This requires very high pressure to be able to overcome significant head pressure when “pushing through” a wine into a large, tall tank. Water management facilities are discussed in *7.1.6 below*

7.1.1.12 Plunging and cap maintenance

There are several techniques available to the winemakers at this winery for cap management of wines fermenting on skins. The tendency for skins to float on top of ferments means that it is important to keep the so-called “cap” wet, to maintain protection for the wine, avoid the development of aldehydic taints, and to enhance extraction from the skins. The open fermentors have a dedicated pneumatic plunger (Figure 9) that runs on a rail system. The head of the plunger is a 40cm diameter star shaped disk. The plunger is operated manually, aided by electric motors used to position the plunger head and pneumatics used to drop and return the plunger through the cap. This process is not physically demanding, but requires a certain amount of skill and coordination to perform the task correctly. It takes approximately 5 minutes to plunge a single ferment. The head is sanitised using hot water and a PMS/tartaric acid solution, in between plunging each fermentor.

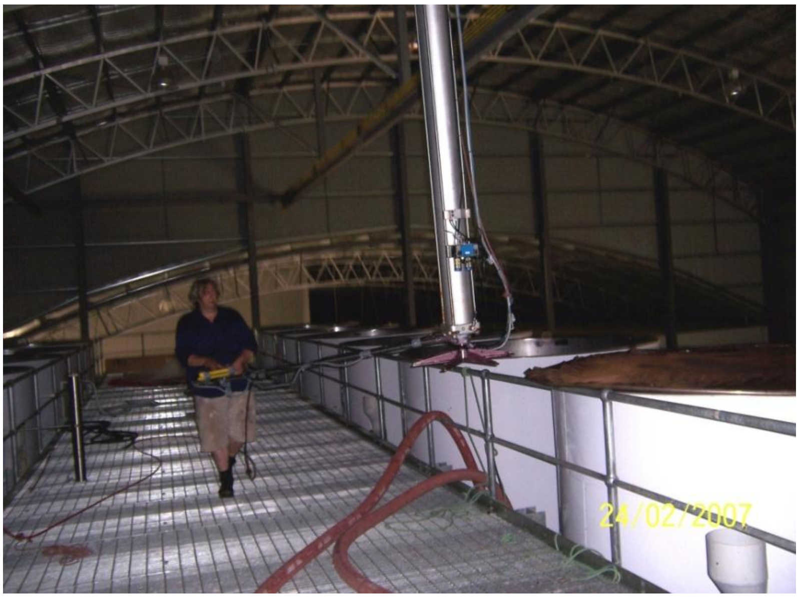


Figure 9. Open fermentor plunger in action at Whitebox Winery

The static fermentors each have a dedicated centrifugal pump and hard lines from the base to the top of the tank. Irrigators are used at the top to spray the wine over the cap to keep it moist. The pump switch is connected to the irrigator switch so a single switch can controls the whole operation.

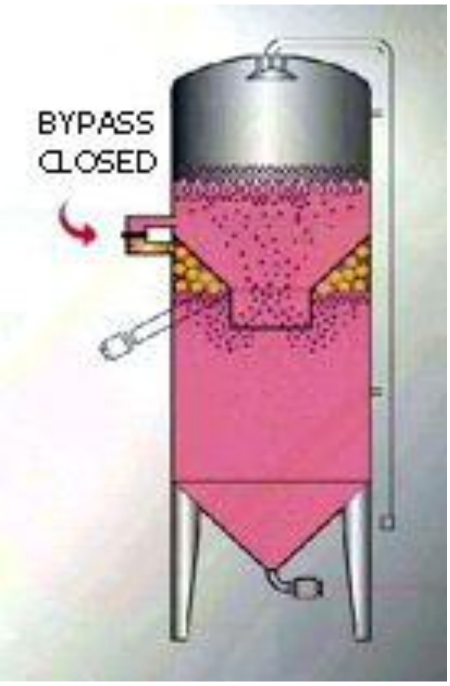


Figure 10. A Ganymede fermentation vessel

Ganymedes fermentors (Figure 10) have cap management built into their structure. The vessel has an inverted, open-ended cone within the main body of the tank, which traps and collects CO_2

generated from the fermentation process. This accumulated gas is periodically released from the cone and rapidly bubbles up through the ferment to turn the cap over. The release is managed by a programmable control system which manipulates the pneumatic bypass valves to release the gas. All other vessels used for fermentation are plunged manually.

7.1.1.13 Bottling Line

The bottling line at this winery is a recent addition. Before it was introduced, wine was either transported off-site for bottling or mobile services were brought in. The system also corks and spins capsules or spins closures, labels, and packages the wine. This allows far more flexibility for the winery when it comes to scheduling bottling sessions, and substantial cost savings and quality control by negating the need to transport the wine off-site.

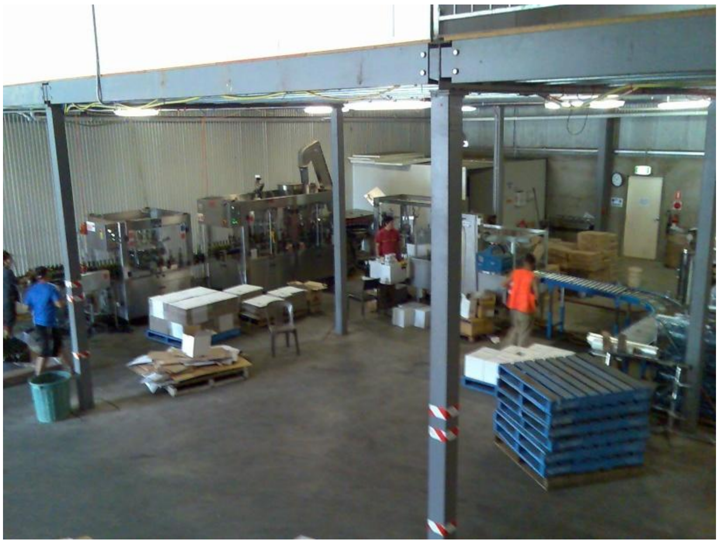


Figure 11. The Bottling Line at Whitebox Winery

7.1.1.14 Nitrogen Generator

Nitrogen is commonly used as a means of protecting wine from exposure to the air. This site has a Pressure Swing Adsorption (PSA) nitrogen generator that accumulates 99.9% pure Nitrogen in a tank which is then distributed throughout the winery via food grade air lines. Nitrogen is used with diffusers which float on the surface of the wine to continually provide a blanket of protection for wines left under ullage. It is also used as a means of gently mixing tanks and as an aid during SO_2 additions via sulphitometers. This generator has replaced the expensive use of commercial Nitrogen packs delivered every few days.

7.1.1.15 Consumables and Materials Storage

The winery has storage facilities for the multitude of consumable materials required for wine production. There is a chemical shed used to store bulk chemicals that simply require isolation from the rest of the winery for protection and safe-keeping. These materials include tartaric acid, PMS, oak chips, citric acid, bentonite, diammonium phosphate, and other commonly used additives. Tracking of these materials is done by stock take rather than on an individual usage basis. LPG, used for fuelling the forklift, is stored in a large tank away from the rest of the winery and is also used as required but no record is kept. A bunded 1500 litre diesel tank is also placed an appropriate distance from the winery and is used to fuel the larger forklift and other winery vehicles. A diesel record is used whenever any is taken. Refrigerated Containers ("Reefers") are used to store any consumable material requiring storage at lower temperatures, such as yeast and nutrients. The reefers can also be used for cold stabilisation of small wine batches.

7.1.1.16 Cleaning

There are two main strategies for cleaning and sanitising equipment in the winery. The first is the use of caustic cleaning agents. The waste water treatment facility on site (*7.1.7 Waste Management*) demands that a non-sodium based agent is used so that treated water subsequently used on the vineyard does not contain excessive levels of sodium. All equipment cleaned using this agent is then rinsed with citric or Tartaric acid to neutralise the alkali, then rinsed with clean water. Sanitisation is achieved by adding PMS to the rinsing solution. Alternatively, hot water can be used as a cleaner and sanitiser and is often necessary when dealing with bi-tartrate deposits left on tank walls. The hot water is about 75°C which is sufficient to provide adequate sanitising results. The barrel washers use this hot water under pressure through their rotating spray heads to ensure complete and thorough cleaning of the internal surfaces of the barrels.

7.1.2 The Vineyards

Each vineyard has a vineyard manager, 2 permanent staff and a viticulturist consultant. They also have their own harvesters and tractors that provide self sufficiency during vintage and appropriate spraying and barrel pruning accessories. Sample testing before vintage is usually carried out in the laboratory at the winery.

Each vineyard has a diverse range of fruit varieties (see Table 3 and Table 4) divided into blocks. They are often treated quite separately, and there are even sub-blocks defined so that different areas of vigour are given treatments throughout the year. Barrel pruning is nearly always used as an initial post vintage pruning treatment, allowing faster subsequent spur or cane pruning. All the vines are irrigated, although the irrigation zones do not always coincide with the designated blocks.

Variety	Acres	Percentage
Shiraz	93.6	57.2
Nebbiolo	21.1	12.9

Tempranillo	19.5	11.9
Merlot	9.9	6.0
Viognier	9.9	6.0
Cabernet Sauvignon	9.7	5.9

Table 3. Heathcote Vineyard Blocks

Variety	Acres	Percentage
Pinot Noir	53	34.2
Shiraz	24.3	15.7
Cabernet Sauvignon	22	14.2
Merlot	14	9.0
Chardonnay	13.1	8.5
Sangiovese	10.3	6.6
Sauvignon Blanc	5.8	3.7
Pinot Gris	3.2	2.1
Mataro	3.2	2.1
Petit Verdot	2.8	1.8
Viognier	2.2	1.4
Pinot Meunier	1.1	0.7

Table 4. Yarra Valley Vineyard Blocks

The vineyard managers, viticulturists, and winemakers consult on a regular basis throughout the year. These discussions include short, medium and long term plans for the vineyards and usually concern cropping rates and fruit quality issues. The viticulturist and vineyard managers have a closer relationship, and are concerned with irrigation, pruning, fertigation, and spraying regimes. These regimes reflect the discussions with the winemakers to ensure that winemaking concerns are catered for. Some fruit is often sold to other enterprises, and it is common for the winemakers for these enterprises to also consult with the viticulturist to ensure quality and quantity parameters.

Viticulturists and vineyard managers tend to have a conservative approach to their vineyards. This is because there tend to be many variables that are out of their control to a greater extent. Factors such as rainfall and pest control tend to make them less likely to experiment with radical solutions. Nevertheless, this does not mean that there is no annual variation. Most notable variations over the period from 2004 to 2008 were harvest dates and water availability.

Harvest dates gradually got earlier and earlier for many varieties, but not all, over this period. 2009 saw the harvest date shift back to 2004-like dates. Although this phenomenon was consistent across the region, there were also pruning labour scheduling issues that caused large variation in pruning dates over this period, along with large rainfall variations and availability of irrigation water that might also have contributed. In a sense, the viticulturists have a constant battle to maintain the status quo, let alone radically modify or experiment with vineyard practices.

Pruning and fruit-dropping labour is usually performed by contractors, whereas all other management tasks are performed either by specialised equipment such as the barrel pruners, or handled by the permanent vineyard hands, if not the viticulturist or vineyard manager themselves.

7.1.2.1 Seasonal Phases

From a viticultural point of view, the year was split into three phases...

Dormancy (winter): is the period from just after the last fruit is harvested until the first signs of growth. This period tends to be mainly concerned with cleaning up the vines and then setting them up for production for the next vintage by pruning them appropriately.

Growing (spring and summer): is mainly concerned with managing the development of the vines and the structures for the fruit. The growing phase is diverse in the range of activities and functions performed, depending on the nature of the vines, and when they reach the various milestones along their development path.

Harvest (autumn): Although there is no discernible change over date from growing to harvest, this phase tends to be where spraying and vine management stops, and simple tasks like sampling and perhaps only some basic irrigation is performed.

It was interesting to note that the dormancy phase was regarded as the time when most impact could be made to achieving the goals of the fruit production for the following vintage. This was reinforced by the fact that they tended to regard a new vintage as beginning the moment the last of the fruit comes off the vines for the previous vintage. This is because all the work that is to be carried out from that point on is deemed to contribute to the outcome of the next vintage and the fruit that will result in the next year.

7.1.2.2 Vineyard Activities

The following table is a summarised timetable for the various activities carried out in the vineyard during the seasons. It is important to note that this is only a guide, and that some tasks may be performed outside of their assigned periods. In other words, the model should not restrict the types of tasks possible at certain times of the year.

Action	**Winter**	**Spring**	**Summer**	**Autumn**
Slashing	✓	✓	✓	✓
Equipment Maintenance	✓	✓	✓	✓

Vineyard Monitoring	✓	✓	✓	✓
Spot Spraying	✓	✓	✓	✓
Pre-pruning (Barrel Pruning)	✓			
Pruning	✓			
Wire Dropping	✓			
Take cuttings for replacing vines	✓			
Herbicide	✓	✓		
Post Replacement	✓	✓		
Fertigation		✓		
Disbudding (Shoot Thinning)		✓		
Vine Training		✓	✓	✓
Foliage Spraying		✓	✓	
Irrigation		✓	✓	
Canopy Management		✓	✓	
Water Shoot Removal		✓	✓	
Fruit Sampling			✓	✓
Bunch Dropping			✓	
Harvesting				✓
Fertilising				✓

Table 5. A list of the common vineyard activities with indications of when they are performed.

7.1.3 WINEMAKING STYLE

It was identified quite early on that this winery represents the upper extremes of winemaking technicality within the scope of this project. This comes about because of a distinct set of circumstances.

- The enterprise is well established but relatively new to the industry.
- The winery is modern and of sufficient scale to justify a diverse range of techniques and products.
- The winemaking personnel are constantly looking for technical and aesthetic improvements in the wine.
- They are prepared to use and often seek-out new technologies and techniques to aggressively seek these improvements.
- Underlying this modern approach remains a healthy respect for traditional methods and minimalist techniques.

The personality and motivation of the winemakers are given full-reign over the winemaking style and this creates a hands-on, tailored treatment of each wine in the facility. The approach tends to be to create each wine in its own right, even if it is going to be combined with other like-batches later, so that there are no issues with trying to cover up perceived faults. Inevitably some wines are better than others, but if there are few faults then the blending exercise becomes a positive experience rather than a pragmatic one, and often makes assigning wines to labels much easier. This is particularly noticeable with wines in barrel, where each barrel is tasted before a racking or refreshing process is carried out. Any faulty barrels are removed or treated separately, and indeed, occasionally outstanding barrels are set aside for reserve status. The diversity of fruit available from their own vineyards is also a contributing factor, because various varieties require different approaches by the very nature of the fruit. Nebbiolo fruit tends to be very tannic and difficult to treat, and has provided a great deal of attention from the winemakers in order to produce a viable wine. They also produce a small amount of sparkling wine base each year which is transported to another facility for processing into sparkling wine.

The diversity of techniques employed here occurs at all stages of the vinification process.

Harvesting: Some blocks or sub-blocks are deliberately harvested at two or more distinct stages of ripeness. Obviously, some Pinot noir and Chardonnay blocks are harvested earlier if they are to be used for sparkling base wines, but it is also common for some wines to benefit from proportions of early and late harvest fruit. This technique tends to evolve over several vintages as the winemakers become more familiar with the characteristics of the fruit at different sugar and acid levels and the resulting wines and their blending characteristics.

Co-fermentation: Combining two or more varieties of fruit into a single batch ferment is said to have aesthetic benefits for the resulting wine. This is occasionally used here but is regarded as problematic because the proportions of the varieties become fixed and rarely result in a better wine than can be created with post fermentation blending.

Crushing/Whole bunch pressing: The way fruit is processed can have a profound effect on the final product. Whole bunch pressing, where hand harvested bunches are lifted directly into the press can significantly reduce the amount of colour extraction from the skins because the skin is not actually broken until the pressing begins, and the juice is forced away immediately. This becomes important for white wine production from red-skinned fruit such as Pinot Noir or Pinot Meunier used in sparkling wine base. The crusher rollers are adjustable to vary the degree of breakdown of the skin structure, but this is only varied occasionally and is used more as a response to berry size rather than a winemaking technique. The destemmer separates out stems and so-called “Matter other than grape” (MOG) from the fruit. The stems contain many bitter and tannic compounds that are not usually desirable, but it is not unusual for some winemakers to deliberately add some stalks into the ferment to enhance these characteristics. Pressing regimes are pre-set on the large membrane press but the system is programmable for special situations.

Cold soaking: Sometimes, the winemaker wants to increase the extraction of colour and flavour compounds from the skins before the fermentation process begins and the alcohol levels increase. Cold temperatures inhibit fermentation but not the extraction of soluble compounds from the skins into the juice.

Yeast choice: The winemakers take a great deal of care in their choice of yeasts to enhance the quality of the wine and ensure complete fermentation, particularly for very ripe fruit with sugar levels up to 15 Baumé. There are usually about 5 or 6 yeasts available to choose from, each with different strengths and qualities. Selection of yeast is usually made well before vintage starts to ensure that there is sufficient supply of the yeasts at the required time.

Nutritional additives: Additives are used to aid yeast viability at inoculation and during fermentation as part of an aggressive move to avoid stuck ferments that have occasionally occurred in previous vintages. The winemakers have also instigated strict procedures for the rehydration of the active yeast.

Enzymatic additions: Enzymes are commonly seen as an effective means of affecting positive outcomes and avoiding negative outcomes during and after fermentation. Acting on a biological level, enzymes have been used to enhance colour and flavour profiles of wines and aid in filtration by settling out components of juice or wines that might otherwise clog filters.

Cap Management and Oxidative pump-overs: As described earlier, cap management techniques depend to a certain extent on the nature of the fermentation vessel. However it is not unusual for more aerative or oxidative techniques to be employed during the earlier stages of fermentation in order to increase oxygen levels in the must, which can aid the viability and strength of yeast in the fermentation process. Typically, a process of “dump and return” is used for many red wine ferments on skins, where juice at the bottom of the vessel is rapidly released out and stored in another tank, then sprayed back over the top of the cap. This is usually done as an occasional alternative to plunging. It is preferable that ferments about to undergo this process are left unplunged for several hours before, and so the usual standing orders (of plunging every 6 hours) is varied for vessels assigned this treatment.

Separation of free run and pressings: This process is inevitably always done with white wines, particularly with Viognier. The pressing process causes the release of material from the skins and can cause the resulting juice to have harsh, bitter characteristics. The increase in cloudiness due to small particulates add to this phenomena and so it is advantageous to treat this fraction of the juice separately from the so-called “free run” juice that comes out of the press before and during the early, lighter stages of the pressing cycle. Once these wines are produced, they can be blended back into each other, perhaps partially, or treated as separate wines. It all depends on the quality of the wines produced.

Lees: Lees consists of particulate matter and solids that settle out of a wine as it ferments and matures, but only after a wine has been pressed and therefore separated from the skins. It can consist of many materials including particulate grape matter, residual fining material, tartrate crystals, polymerised tannins, protein and metal precipitates, and yeast and bacteria. After racking a wine, the lees is evaluated by the winemaker to determine whether it might be useful for other wines in the winery. It has been found that lees can add favourable characteristics to a wine in terms of flavour profile, and also aid in the fermentation of wines that are sluggish toward the end of the fermentation process. Lees is only ever sourced after the first racking of a wine, after fermentation

As far as the L.I.P. is concerned, lees is still regarded as wine and needs to be accounted for in the data model. When lees from one wine is added to another, this is regarded as adding that volume of wine to the original even though only a very small component of it will remain in the finished wine. Such losses during the subsequent racking or filtration process are regarded as losses spread across the whole of the wine, not just the lees component. This can impact on variety, vintage and regional label claims of the final wine.

Oak treatment: Oak barrels are used extensively in the winery for most wines. The winemakers prefer to use a mixture of old and new barrels in order to get a mixture of levels of oak treatment, allowing blending later to reach an appropriate balance for the wine. A large range of barrel types are used, from various manufacturers, countries and regions. Over the years, the winemakers have determined their favourite combinations for varieties of wines and barrel types. The cost of barrels varies greatly, from approximately $500 to almost $1500 per unit. The cost and benefit of each type must be matched and justified against the eventual price of the wine being produced. Some wines are treated with oak chips instead of barrels, being a far cheaper means of adding oak compounds to the favour profile of the wine. This is usually combined with MOX treatment to even out the effect of the "artificial" oak addition. Some barrels are used for white wines in their first year, and then used for red wines in subsequent years. Once a barrel is used for a red wine, it is not possible to use it with white wines.

Accounting for oak use is not a straightforward process. In pure accounting terms, the cost of using oak is calculated in much the same way that capital equipment is depreciated over the period of its *use. Industry standards suggest that a barrel loses 80% of its oak value per annum. This is of course a spurious figure given the fact that barrels have a two-fold effect on wine. Apart from the release of oak compounds into the wine, the nature of the vessel also allows the slow ingress of oxygen into the wine, which aids in maturation. This effect is not significantly affected by the age of the barrel and the amount of use it has sustained. So even old barrels used for 3 years, which have apparently imparted more than 99% of their oak value, still have winemaking value. Wine only penetrates millimetres into the surface of the wood, so there are also processes available to have barrels renewed by shaving the insides of the staves and headboards, and revealing fresh wood, and then re-toasting. Clearly some method of depreciation will need to be adopted within the model. Whether this will be a constant or proportional depreciation methodology is up for debate. Clearly, the model will need to record barrel use on a barrel by barrel basis, including the time it is used, and the colour of the wine. It is also feasible that the type of wine should also be recorded so that the winemaker can choose which barrels might be suitable for further use with different varieties. Keeping individual barrel records will also allow a more detailed history of how the barrel performed, with respect to integrity (leaks or splits) and also the effect of the barrel type and age on the variety of wine.*

Malo-Lactic Fermentation: This is the process by which malic acid in the wine is fermented via various metabolic pathways, into lactic acid. Malic acid typically tastes of green apples and can provide a fresh and crisp profile to the wine, whereas lactic acid is typically milky, leaving the wine softer and allows more subtle characteristics to come to the fore. It is commonly known as secondary fermentation. MLF is used selectively in this winery. Although this process can occur naturally, the winemakers prefer to deliberately inoculate wines that require MLF, and inhibit natural MLF in wines that they do not want to go through this process. The inoculation process also ensures that they get to choose which bacteria cause this process because some bacteria may have detrimental effects on the wine.

Laboratory analyses: Laboratory testing of wines is used as an oenological technique to prevent and diagnose wine faults as well as understand the chemical profile of the wine. It provides most of the objective data for a wine and can be used to understand more subjective aspects of a wine's appearance, nose and palate. Above all, it can be used to see the effect that other actions, deliberate or otherwise, have had on a wine. This winery has dedicated laboratory personnel who are required to perform many tasks as a matter of course. For example, dissolved oxygen analyses are mandatorily taken after a wine is transferred or racked from one tank to another, and sparging is sometimes required if the D.O. level is too high. All wines are tested for sulphur levels at least once a week. As well as these standing order analyses, the winemaker will often request specific analyses to be carried out on an ad-hoc basis, as a means of justifying perceived faults or as precursor to subsequent tasks. As discussed earlier, many of these analyses are part of a sequence of tasks to achieve an overall processing task, such as processing fruit, or refreshing a wine in barrel. It is also common for a wine to be tested for several parameters at the same time so that fewer samples are required to be taken.

Laboratory personnel not only respond to ad-hoc requests, but also maintain a list of standing orders as well as monitoring other processes in the winery that may require subsequent analysis of the wine. This provides an important opportunity for the data model to help in coordinating sample taking and analytical tasks to ensure efficiency, accuracy and timeliness.

Clarification, Stability and Fining: There are five aspects of a wine that need to be achieved before it can be bottled. The wine needs to be clear, cold stable, protein stable, fined to taste, and filterable in the sense that a coarse filter is used in line in the bottling process and this will clog if the wine is not free of compounds or materials that block this filter.

Red wines are usually cross-flow filtered which generally achieves clarity. Cold stability is only usually sought for the more commercial wines where consumers are less likely to see tartrate deposits as a positive quality. Protein stability for reds is not regarded as an issue, so it is only fining and filterability that are of significance. Extensive fining trials are carried out to ensure that a satisfactory flavour profile is achieved. Filterability is tested a few weeks before bottling is to be done, and the wine can be cross-flowed or otherwise clarified in time for the bottling process.

White wines go through a similar sequence, but greater emphasis is made of protein stability. Bentonite trials are used to ensure protein stability and clarity, although the former usually assumes that the latter will be achieved. Fining trials are performed before and/or after this process, but each wine is treated separately and there is no consistent formula used across the range.

The winemakers have a wide range of materials and techniques to refine the wines, although the cross-flow filter is the tool of choice for clarity, and bentonite is nearly always used on white wines for protein stability. Cold stability is achieved in tank by increasing the cooling to get the wine to 4°C for a period of time. Several fining agents are used, although the winemakers tend to have their favourites for particular wine varieties. It is not uncommon for fining to be performed in steps because fining can be regarded as removing a negative as well as enhancing a positive aspect of the wine. Sometimes more than one agent is used at a time.

Clearly there is no fixed sequence or timeline for any of these processes in the development phase of wines. Each of these steps can radically alter the organoleptic nature of the wine, and therefore each process needs to be regarded as optional, although it is rare that no processing of this nature takes place, particularly fining. The opportunity exists for the model to link subjective data such as tasting notes with fining trials and additions, just like sulphur tests can be linked to sulphur addition work-notes to verify the effectiveness of those additions.

Experimental Batches: It is common for experimental batches to be made in order to explore any means of improving the products. The success or otherwise of these batches often has implications for the subsequent vintages. These batch tests need to be carefully designed to ensure that the desired effect can be attributed to the technique being applied. Several experiments have led to significant changes in the way some of the wines are produced.

Sparkling Wines: Small batches (1-2 kL) of sparkling wine base are produced each year and then sent off-site to be processed into sparkling wine using traditional methods.

Sanitation: Water quality has been a significant issue at this winery. During the first few vintages (2004-2006) significant problems occurred for several wines during their fermentation and it was later discovered that these issues were most likely due to spoilage yeasts and bacteria that were discovered in the water supply. The winemakers have taken major steps to remove this problem by employing HACCP (Hazard Analysis Critical Control Point) techniques. Yeast preparation is now strictly controlled during vintage. More disciplined sanitation and preparation of vessels and equipment is part of far stricter SOP's (Standard Operating Procedures). The water supply has been changed and steps are in place to include ozone and filtration processes as well as regular testing of water storage facilities.

HACCP principals are now recognised as an essential part of the winemaking process. This does not necessarily mean that a winery needs to be certified according to ISO standards, but rather that the techniques used provide a foundation for the way wine production occurs, and can aid winemakers avoid expensive supplemental processing and quality degradation as a result of poor or negligent techniques. The data model should be capable of providing analytical tools to monitor the critical control points in the manufacturing process, and enable investigative research into the probable causes of problems that occur. It is not suggested that the model should carry out the HACCP process, but it should be able to provide functions that facilitate this process.

Blending: Which wine ends up under which label and how these wines are blended is down to a complex process of tasting, fining and blending trials. This process is complicated by the fact that although the winemakers have a specified range and volume of wines to produce according to the sales plan and vintage plan, opportunities may exist for variations of these plans to take advantage or to compensate for contingencies that inevitably occur. The winemakers will usually have prepared well in advance to ensure that they have an appropriate range of components available in order to provide a satisfactory blend. For example, some components may have received no oak treatment, or perhaps pressings and free-run wines have been kept separate so that a desirable blend can be made later once the individual wines have reached a suitable level of maturity.

Just like the clarification, stability and fining processes, there is no fixed sequence or timeline that can be defined for the blending of wines. Nor is there a clearly defined process for blending trials, or even defining a desired outcome. However, by observing the winemakers and analysing how they have gone about rationalising the decisions they have made, it is clear that one can conceive a set of tools that would aid winemakers in this process. These scenarios are discussed later.

7.1.4 Wine Tasting, Amelioration, and Fining Trials

Tastings are held regularly, particularly during non-vintage times, to ensure the wines are maturing and progressing as expected. The idea is to look for early signs of problems as well as suggest ways to improve the wine, or prepare it for finishing. Generally, wines are tasted in varietal batches to compare the various lots. Some wines may then be nominated for fining trials, or filtering or any number of other processes. Brief tasting notes are attached to each wine, but more specific notes regarding the rational for subsequent processes are not necessarily recorded. Subjective data does not feature highly in the recording structures. The winemakers tend to rely on their own memories to judge the success or otherwise of amelioration techniques implemented to achieve subjective outcomes. It is common for fining decisions to be quite conservative, so that fining is performed iteratively and the ideal level is approached in steps.

7.1.5 Operations

The term "Operations" is an all-encompassing concept for the way the enterprise carries out the task of producing wine. In this case, there is a distinct hierarchy of roles and responsibilities to ensure the micro and macro management of the enterprise. However, because the enterprise consists of so few staff members compared to the plethora of tasks required to produce the desired outcomes, there is naturally a requirement for staff to show flexibility within these roles. The distinction between defined roles, the definition of responsibilities for these roles, and the actual performance of the tasks becomes blurred. In general, roles and responsibilities at each level of the hierarchy match the level of significance and implications for the tasks they perform.

The Chief winemaker is responsible for wine styles, and ensuring the production plans are met. His decisions have a fundamental effect on all other decisions and tasks within the enterprise, for the vineyards as well as the winery.

The Senior winemaker works closely with the Chief winemaker and is responsible for ensuring the Chief winemaker's decisions are realised in a practical sense. There is a shift of emphasis from the theoretical to the practical as the tasks become more functional, and the basis for decisions rely more on pragmatics than ideals. This role is responsible for scheduling harvest dates, staff, equipment and materials, as well as making most of the ad-hoc, practical winemaking decisions based on the parameters set out by the chief winemaker. It is worth noting here that there is some blurring of roles as the Chief winemaker and assistant winemaker often perform these tasks too, after consultation with the senior winemaker. The senior winemaker usually starts the day by defining and assigning all the tasks required for the day.

The assistant winemaker role is generally defined as being in charge of cellar staff and coordinating the tasks set out by the senior winemaker. This role also provides feedback to the senior winemaker about what is going on in the winery, and any problems that are occurring. It is usual that cellar staff have significant experience in winery operations, however during vintage this may not be the case, particularly as it is common for the enterprise to engage staff from overseas; student winemakers wanting to experience winemaking from an Australian perspective. The assistant winemaker is responsible for monitoring all cellar staff in the winery, to ensure that tasks are carried out safely and correctly.

Most tasks performed in the winery are planned, and performed using "Standard Operating Procedures" (S.O.P). The senior winemaker has developed a system of "work-notes" which are simply printed sheets of paper describing details of the task to be performed, SOP's for the task, and fields allowing the entry of feedback parameters for the task, if any. Having said that, it is also common for there to be exceptions or modifications to the SOP's, and these need to be explained clearly on the work-note. Other tasks are assigned on a standing order basis without formal documentation. These tasks include laboratory analyses, cap maintenance, barrel topping, routine cleaning, equipment maintenance, and so called "Scum Runs" which entail the inspection of all wines in the winery for signs of problems such as oxidation or microbial contamination, checking that cooling or heating is correctly set, and any ullage issues.

Here is a list of the most common tasks performed at the winery.

- Process Fruit including weighing, crushing, pressing for white wines, and delivery to vessel
- Inoculate with Yeast or Bacteria for MLF
- Press red wines
- Mixing or Cap Management such as pump-over, plunge, battonage, valve to valve, drain and return
- Additions to wines such as tartaric acid
- SO_2 addition
- Laboratory Analyses such as Temperature and Baumé, SO_2, pH, TA, Volatile Acidity, Alcohol malic acid, dissolved oxygen, Ammonia, primary amino acid, CO_2, turbidity, residual sugar, cold stability and protein stability.
- Laboratory Trials such as addition, tasting, blending, or fining
- Filter
- Fining
- Rack or Transfer a wine
- Cooling/Heating adjustments
- Bottling wines
- Scum Run
- Diffuser
- Micro Oxygenation set up or adjustment
- Clean and/or sanitise a vessel
- Sparge a wine
- Dump a wine
- Load a wine for transport
- Clean/Sanitise equipment such as a press
- Transfer skins from press to a ferment
- Top barrels

Many tasks consist of a set of individual tasks that need to be performed in a sequence, with feedback parameters from some determining the nature and necessity of subsequent tasks.

Example 1: The senior winemaker may wish to periodically "refresh" a wine in barrel by pumping it out into a single tank, cleaning the empty barrels, checking the SO_2 levels and pH of the consolidated wine and adjusting them accordingly, then returning the wine to barrel. Each of these steps might be considered a single work-note themselves, and obviously the need to add acid or SO_2 depends not only of the levels that currently exist but also on other winemaking and oenological decisions that need to be made at the time.

Example 2: The fruit reception process needs to be carefully coordinated. The primary work-note sets out details of how the fruit should be processed and in which vessel(s) it should end up. However, at the same time, the winemaker may want additions to be made such as yeast, acid, sulphur and enzymes. The acid addition needs to be made based on the pH and titratable acidity levels of the fruit, but it is preferred to make additions to the fruit during the processing, in the must bin, well before it ends up in the vessel. This is because the act of pumping the must actually mixes the additive well, and means subsequent additions and further mixing are not necessary. So the pH and T.A. (Titratable Acidity) analyses need to be made immediately the fruit arrives, and the additives need to be prepared before the processing begins. This is particularly necessary for the yeast, which can take up to 20 minutes to prepare before pitching.

So we can see that the processing of the fruit can consist of several individual types of work-notes, with a degree of interdependency for some of them. For example...

- Process Fruit
- Inoculate with yeast
- Analyse pH
- Analyse T.A.
- Analyse SO_2
- Add Tartaric Acid
- Add SO_2

The fruit is weighed on arrival. The analyses can be carried out immediately, and a malic acid analysis is usually ordered at the same time, but this is performed after the pH, T.A. and sulphur aspiration. These results are then given back to the winemaker who decides whether additions are necessary and at what rate. Usually, the SO_2 addition specified to achieve a certain level of free sulphur in the must, whereas the acid addition is made to achieve a certain pH although the T.A. level is taken into consideration too. There are general rules of thumb for addition rates to achieve both although it is also common for the winemaker to get analyses done a few days later to verify the effectiveness of these additions. Given the mass of the fruit (stalks and MOG {Matter Other than

Grapes} are assumed negligible) and the stipulated rates of addition, the additives are prepared and added during the processing.

This whole situation is also complicated by the fact that fruit coming from the local vineyard may arrive in 2 tonne bins over a period of several hours. So decisions need to be made as to when these additives are prepared and added.

Example 3: There is a concept of standing orders for plunging ferments. Usually four times a day at 6 hour intervals, each ferment needs to be plunged or pumped over, depending on the nature of the vessel. If the winemaker has decided that an addition needs to be made, such as an acid addition, then it makes sense that this should be coordinated with the cap management regime so that the acid will be mixed immediately it is added to the ferment. On the other hand, if a wine is about to be pressed, or a “dump and return” has be ordered, then it may be important that this ferment is not plunged.

Example 4: Preparing a wine for bottling includes several analytical tests to ensure the wine is heat and cold stable as well as filterable. The results of these tests may lead to subsequent clarification processes being applied. These processes make take several days to perform, so it is clear that some of these work-note sets may take a week or two to perform.

Similar sets of work-notes exist for other stages of wine production. So we can see that although these work-notes are individual tasks, it is common for them to be combined into a set of work-notes, with a critical sequence of steps that may have impact on other tasks in the sequence.

The strategy and plans for each wine are well understood by the winemakers; however this information is not necessarily explicitly understood by the other staff in the operation. The cellar hands tend to act on instructions from the winemakers but do not necessarily always understand why they are performing the various tasks. This has occasionally caused some confusion when seemingly contradictory tasks have been performed without the staff really understanding the circumstances, and possibly because they lack the technical and oenological training to comprehend the purpose of the tasks. This is not a fault in the system, but to some degree, particularly on the scale of this enterprise, a little information could be useful. This is particularly relevant when many of the staff are actually in the middle of tertiary training to become winemakers themselves, and might provide useful feedback with a better understanding of the reason for performing a sequence of tasks.

7.1.6 Water

The quality of the water is paramount in wine production. It is estimated that wineries use about 2.5 m^3 per tonne of fruit crushed (http://www.wfa.org.au/PDF/Environment2003.pdf). This volume is insignificant compared to the 364.5 m^3 per tonne of grapes required by vineyards.

Water has always been a major issue for this enterprise, particularly for the Heathcote region vineyard and winery. It is sourced from three major suppliers; channel water, rainfall, and delivered town water. The main supply is from so-called "channel-water" supplied via a pipeline and pumped into dams on the property. Each year the vineyard is allocated a certain volume of water. The water is of sufficient quality for use in the vineyard, but not for the winery. Up until recently the winery relied on rain water collected from the roof of the winery, and supplemented with "town water" delivered by truck. There has always been a major concern with the possibility of chlorine in the town water, and so the winery has recently embarked on a process of cross-flow filtering and ozone sanitising the dam water to generate appropriate quality water for the winery.

7.1.7 Waste Management

Waste from the winery comes in two forms; namely marc and water. Solid marc such as stems and skins is transported using a tip-truck, to a marc pit and mixed with manure and used as a fertiliser/mulch on the vines. Water is treated using a series of small dams to firstly aerate the water for microbial and oxidative action on the waste followed by a settling pond (Figure 12) to allow the

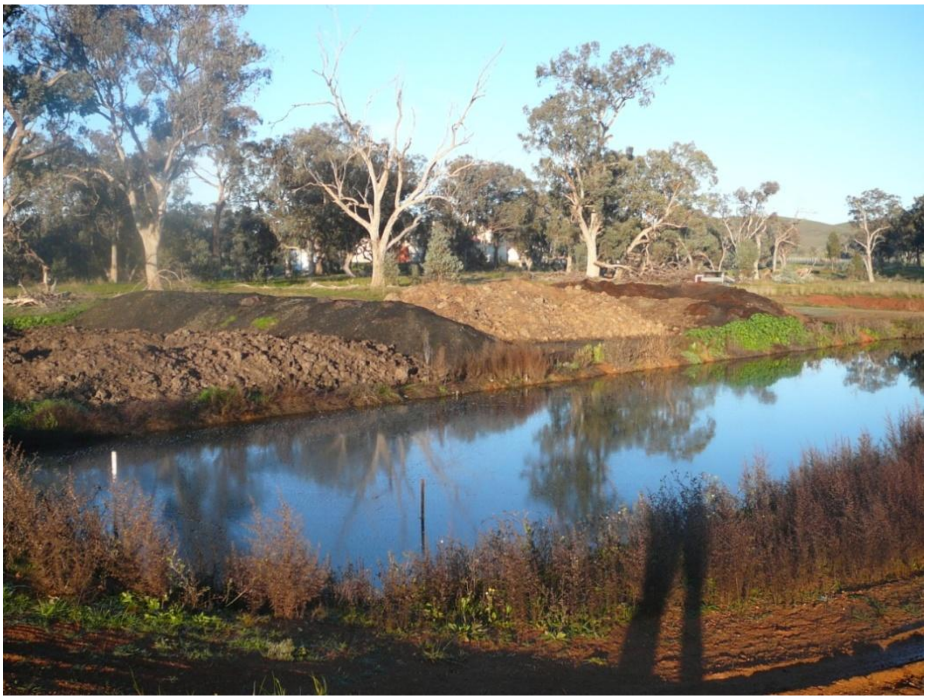


Figure 12. The settling pond for waste water treatment and marc heaps at Whitebox Winery

solids to settle and to then decant the relatively clean water into the dam used to store the channel water. This process is monitored by winery staff throughout the year, apart from vintage when it is supervised by the vineyard staff. The D.O. level and pH of the aeration pond is carefully monitored each day, particularly during vintage when waste water is generated at a far greater rate. The pH varies greatly during vintage and tends to be higher at the start due to cleaning operations, and drops to quite low levels during racking and pressing operations later in the vintage period. The system can cope with these variations very easily and the "cleaned" water coming out at the end is consistently between 6.5 and 7.5 ppm O_2.

7.1.8 Safety

Regular visits and audits by WorkSafe are carried out, and it is the responsibility of the Chief and Senior winemakers in the Winery and the Vineyard managers in the vineyards to ensure that the various sites are compliant with any requests. Most permanent staff are trained in first aid and there is always a trained staff member present when any operations are being performed. The winery appoints a permanent staff member as an OH&S officer who is responsible for monitoring safety issues within the winery, including organising regular safety meetings, addressing safety concerns from staff, and maintaining Material Safety Data Sheets records and standard operating procedures

7.1.9 Storage Facilities

This enterprise has a wine storage warehouse located in the north eastern suburbs of Melbourne where almost all of the wine produced by the winery is stored for easier distribution to the market. Wine is also stored at the winery, in the barrel hall. Both facilities act as staging points for exporting and domestic wholesale sales. The enterprise has no retail sales apart from mail order or internet sales.

7.1.10 Labels

As part of the marketing strategy for this enterprise, there are four different brands used, each targeted at a different price-point in the market.

Label 1: This is the premium brand used to market the best wines produced by the enterprise. It tends to be the most highly protected label, and only wines of premium quality are put into this label. These wines tend to have higher quality oak used and receive the most care and attention. The batches tend to be smaller, and are usually sourced from the Yarra Valley vineyard, although exceptions to this do occur. The bushfires in the Yarra Valley region in 2009 adversely affected most of the fruit, and as a result only very small amounts of wine were produced under this label for the 2009 vintage.

Label 2: This label is the mainstay label for the enterprise and is the most recognisable in the market. It is generally associated with the Heathcote Shiraz/ Rhone blend wine style and represents about 70% of the wine production for the enterprise. The popularity and demand for this label tends to dictate production volume rather than the requirement for the wine to meet the standards of the label's reputation. It is common for the enterprise to buy fruit from other sources within the region to supplement the production of wine for this label.

Label 3: This label represents the more exotic and alternative wine varieties produced by the enterprise, at least when compared to typical wine varieties in Australia. These wines are at a slightly lower price-point than label 2 wines.

Label 4: The fourth label is an innovation and a response to the market place that required a new label at the same level as label 2 but allows a broader range of wines to come under the same price-point. It is a combination of labels 2 and 3.

Label 5 is the lowest price-point and is used to provide a means of selling wines that do not meet the quality requirements for the other labels, or are surplus to their production requirements. These wines may or may not be given specific "Geographical Indication" (G.I.) reference or even vintage status according to the "Label Integrity Program" (L.I.P.) regulations. It is common for these wines to be of far greater quality than the price-point might dictate, and can often serve as a means of attracting consumers to the higher price-point wines produced by the enterprise.

7.1.11 Exporting

This case study highlighted issues associated with exporting wine; in particular, the need to understand the technical regulations and requirements which are briefly discussed in *8.2.5 Managing Paperwork*

This enterprise treats the exportation of its wine in much the same way as the domestic sales, but with an extra layer of complexity with respect to licensing, labelling compliance and transportation. Wines are identified for export well before they are bottled, but it is a general policy that labels and packaging are designed for compliance for all major exporting targets. There are exceptions of course, and when new opportunities occur after bottling, new back labels or over-stickers are ordered. All wines for export are immediately sent for export approval, and continuing approval is also sought when necessary, on an ad-hoc basis. Export approval for individual consignments is often managed by export managers working on behalf of the enterprise, but often this task is left to staff within the enterprise.

7.1.12 Use of Barrels

If a wine is selected for barrel treatment, the winemakers typically select a range of barrels of different ages and oak specifications. This decision is based not only on what is deemed appropriate for the wine, but also the availability of the barrels themselves. It is also common for only a proportion of a wine goes into barrel so that blending later can achieve a suitable balance of oak in the final product. Splitting a wine into individual barrels also offers the opportunity to minimise the danger of spoilage to a small proportion of a wine rather than the whole batch being tainted.

The winemakers commonly treat a set of barrels containing the same wine as, conceptually, a single wine. Analysis of a wine in barrel usually means that small samples are taken from 4 to 6 barrels in the batch, and this accumulated sample is then tested. However, when the wine is subjected to some form of homogenisation, during a racking off lees, or a sulphur and acid adjustment, the winemakers usually taste each barrel before this process, to ensure that that small proportion is comparable to the others. If it is significantly better, it may be taken out of the batch and treated as a separate wine. If, on the other hand, there are problems, it may also be isolated and given supplemental treatment in an attempt to get it up to quality. Variation is always expected, because of the individual characteristics of the barrel, and it is a deliberate act to seek this variation, but occasionally a small number of barrels may stand out in some way that is beyond the scale of variation expected.

Although strictly not barrels, the enterprise has also purchased approximately 60 Flextanks which are 1m^3 plastic tanks. Looking much like pallecons, they are made from special polymer compounds that permit a similar ingress of oxygen like real barrels, thereby approximating the aging benefits of oak vessels.

It is common for a balance of new and old barrels to appear in a set of barrels for an individual wine. For this reason, it is common for this set of barrels to be kept together and used as a set, with the occasional introduction of a new barrel or the retirement of an old barrel. It also means that the added complication of a conceptual single wine being stored in many individual vessels can be avoided. This is beneficial from a data and accounting point of view, as well as from a winemaking point of view because the data model would become overly complicated by having to reference multiple instances of barrels in any data manipulation, and the winemaker does not want to have to refer to barrels on an individual basis unless the circumstances deem it appropriate. A barrel set does have some disadvantages, but this is discussed later. (11.4 Barrel Sets)

7.1.13 Contract Work

On occasions, the winery is asked to perform some processes for other enterprises on a contractual basis. This can be in the form of the initial processing of fruit, to enable the must to be transported across regional borders, or the complete fermentation, maturation, and bottling of wine using either the fruit from the enterprises own vineyards or other vineyards in the region. These contracts usually involve separate rates for materials, and use of the facilities and labour components. The winery also offers laboratory services to perform many common analytical tests for local winemakers.

The data model needs to treat these wines quite separately, with respect to precise records concerning the use of oak, the amount of materials used such as fruit, yeast and additives, time spent processing, and any other contractually stipulated parameters. It is not the intention for the data model to handle the accounting side of such arrangements, but rather to keep track of the data in such a way that this information can then be fed into an accounting system.

7.1.14 Transport

The nature and geographical dispersal of this enterprise and its constituent facilities necessitates extensive transport requirements. Not only does fruit need to be transported from the Yarra Valley to the winery in Heathcote, but also bottled wine needs to be dispatched to the warehouse facilities near the city, and other ad-hoc transport jobs.

Obviously, trucking suppliers need to be kept informed about such details because harvested fruit has a very short life-span when left unprocessed for extended periods of time.

It is worth noting here that the enterprise relies heavily on mail and courier services for all smaller items.

7.1.15 Budgets, Accounting, and Costs

Each year the winemakers prepare a proposed budget for the coming vintage. This proposed spending plan estimates the processing costs likely throughout the year. It provides a seed for hiring vintage labour, and pre-ordering vintage requisites such as yeast, barrels, and additives. These are not necessarily all the costs associated with operating a winery and represent only part of the overall budgeting process for the enterprise. In a sense, it is the part of the budget that has the greatest impact on how the winery performs and what functions it wants and needs to perform at the busiest part of the year. It also defines spending during the most critical phase of the annual cycle of operations in a winery.

The issue of costing is significant to the winemaker when there is a degree of discretion as to how money can be spent and which wines can be given a greater proportion of the available funds; typically the higher price-point wines are given greater discretionary funding. However, not all costs that are discretionary, or are part of the vintage budget, can be attributed or assigned to an individual wine because the monitoring of the allocation of these costs becomes tedious and uncertain. In general, these costs are averaged out per litre of wine still in production over the period in question, regardless of the market point or relative quality of the wine. They are nevertheless important to budget for because the scale of the vintage inevitably affects the scale of use of whatever these costs involve.

Labour: The cost of labour must be regarded as a cost to be averaged out over all the wines in the winery. Salaries to permanent staff are regarded as constant over the period of the budget, whereas vintage labour (paid by an hourly wage) is estimated based on the scale of the vintage, and estimated peak workloads, and the length of the vintage.

Utilities: such as LPG, electricity, water, or diesel fuel are also non-attributable, but must still be estimated because they are not necessarily constant from vintage to vintage as circumstances change. A good example of this is the change in cost and usage of water between different vintages as sources and usage regimes were changed. Although there is a distinction between electricity which is paid for after it is consumed, and the other materials which are delivered and stored, paid for, and used when required, it is common sense to attribute all the costs when they are paid for rather than trying to meter out costs as they are consumed.

Equipment Service and Repairs: All the major equipment in the winery and vineyards is serviced at regular intervals, and particularly before vintage. This cost is predictable, but repairs are not. Although the winery is remote from significant engineering resources for repairs, the managers have established a sound relationship with an electrician who is available on call at an hour’s notice. It is common knowledge among winemakers that good trades-people are the most critical external people with whom a good relationship is essential. Such costs can be significant, but usually pale into insignificance compared to the untimely breakdown of critical equipment such as a press or a pump, which can leave products exposed and prone to adverse conditions for extended periods of time, if not complete loss of product. Without fail each year, one or both of the harvesters have broken down at critical times, even though they are maintained and serviced to very high standards. This has caused major disruption to vintage plans, staffing, and harvest scheduling. It is a prime candidate for HACCP analysis. Generally repair costs are part of a contingency budget maintained as a buffer.

Consumables: This is an interesting category because some items can be attributed to a wine without undue burden on record keeping. These sorts of items include fruit, yeast, MLF bacteria, yeast nutrients, and packaging supplies (dry goods) which need to be used specifically for wines in exact amounts. Other items such as tartaric acid and PMS are used both in wines and throughout the winery generally to clean and sanitise vessels and equipment. Monitoring use of these items would definitely be a burden. They therefore fall into the same category as cleaning agents, laboratory chemicals and other items used in a non-wine-specific way throughout the winery. Fortunately, they tend to have a long expiry date and are used throughout the year so the winemakers tend to be generous when budgeting for consumables like tartaric acid and PMS.

Barrels: As discussed earlier, barrels are "consumed" only when they actually contain wine, and a system of depreciation is used to account for the cost. This rightfully means that unused barrels do not deteriorate if they are kept in a reasonably humid and cool environment. So, as far as winemakers are concerned, barrels have not contributed to the cost of winemaking until they actually have wine in them.

Transport: Transport costs are perhaps more significant for larger scale wineries that source fruit from other vineyards. In this case study, a significant amount of fruit comes from their Yarra Valley vineyard, and so this expense is anticipated. Harvest tonnage estimates and dates are used to indicate the number and scale of the transport costs required.

Contingencies and Exigencies: Experience from previous vintages reveals that it is always wise to budget for likely contingencies. An example of this occurred in 2008 when smoke taint affected some of the fruit harvested from the Yarra Valley as a result of bushfires a few weeks before vintage started. It was decided to attempt to remove this taint once the wine was produced because it was felt that the level of taint did not make the fruit a lost cause. Taint removal requires a specialised reverse-osmosis filter available by a mobile contract service. The early realisation of this problem enabled appropriate budget modifications to accommodate this added expense, and justify a reduction in oak requirements for the vintage.

It is not proposed to offer a budget designer function within the model. However, there are several tools that will be useful. The function of this budget is not just as a planning mechanism to have a better understanding of what is likely to occur during the year, and the vintage period in particular, but also provides means of seeding a tool to allow winemakers to "play" with processing regimes for individual wines before and during vintage as well as comparing plans to outcomes after vintage.

7.1.16 PHYLLOXERA

The detection of the grapevine pest phylloxera in the Yarra Valley Region of Victoria led to the establishment of the Maroondah Phylloxera Infestation Zone (PIZ) in 2006. This zone was extended in late 2008 as a result of the detection of further infestation outside the original PIZ, and consequently this enterprise's Yarra Valley vineyard was made subject to the Victorian Department of Primary Industries phylloxera management protocol. In a nutshell, this prohibits the transfer of grapes up to the Heathcote facility for processing, including maturity samples. As a result, the enterprise had to make arrangements for the 2009 vintage to have the fruit processed at a facility within the extended PIZ before it was able to get the material to the winery in Heathcote.

From a logistical point of view, this was an expensive but surmountable problem. However, from a winemaking point of view, it was almost intolerable. The decision was made to get the fruit crushed and transported immediately to the Heathcote winery so that the winemakers could take control of the fruit as soon as possible. However, two complications arose. Firstly, it was very difficult to transfer the must from the transporter tank, taking some 6 hours to empty each time, and secondly, with the danger of smoke taint from the recent bushfires, there was a severe danger of exacerbating the issue by extending the skin contact time, and added processing through must pumps. This caused the winemakers to decide to have subsequent batches of fruit crushed, destemmed, and pressed prior to chilling and subsequent loading onto tankers for transferral to the winery. A very careful pressing regime was used allowing free run and light pressings to be taken with very little smoke taint detectable. This of course, also meant that the red wines had very little colour extraction, but this was accepted as inevitable anyway, given the smoke taint issue.

The idea that this enterprise is made up of a winery and two vineyards is too simplistic from a data model point of view. External vineyards, and now wineries, might be deemed to be simple external entities that just need to be referenced in the same way that a supplier is referenced; that is as a non-descript object used for reporting and accounting purposes, but in reality, they are intrinsically involved in the details of the model. In this example, the two vineyards and the winery are major entities in the overall scheme of things. Obviously they will serve as core entities in the model, and will be represented in great detail and referenced by most of the other entities in some way. It is equally clear that external entities such as the winery used to process the fruit, represents a second tier of such entities, albeit a simpler version. This will allow work-note objects (for example) to refer to the appropriate winery object, and rely on equivalent references assumed available for work-notes performed at the winery. In object-oriented programming terms this is known as polymorphism.

7.1.17 NAMING CONVENTIONS

The name of a wine is intended to give an abbreviated summary of the wine's origins. Each fermentation batch is given a name using a common syntax YYVINVARDDMMX where...

- YY is the vintage year (This would certainly raise a few eyebrows in the software development community for those who experienced the Y2K phenomenon)
- VIN is a two or three letter vineyard descriptor used to distinguish fruit from different vineyards.
- VAR is a three letter abbreviation for the fruit variety
- DDMM is the two number date and two letter month for the harvest date
- X is an optional letter used to distinguish otherwise identical ferments from fruit harvested on the same day from the same vineyard of the same variety.

Later in the process when wines are perhaps blended or designated to a particular label, the harvest date is dropped, and the vineyard designation becomes a label abbreviation or perhaps a region designation, depending on the status and history of the wine. The variety part of the name also changes to allow for blends, and usually adopt six letters for two or three variety blends, such as CABMER for Cabernet sauvignon-Merlot blends, or even SHMAVI for Shiraz-Mataro-Viognier wines.

The naming convention is not fixed, and tends to adhere to ad-hoc standard that is used to ensure wines are named uniquely, but not necessarily descriptively. Exception have occurred when part of a wine is put in barrel and given a "B" extension to keep it separate from the rest of the wine remaining in a tank. Small test batches are given letter extensions too, so if a wine is given three different treatments, they will typically be given extensions "A", "B" and "C". Clearly, this shows how misleading it can be to read too much into the name of a wine.

Although it will be nice to automatically give a wine a name based on the appropriate harvest parameters, a wine name must be the winemakers' decision, and therefore cannot be used to store any meaningful information from a data model point of view. The model will need to ensure that names are unique, as an aid to the user of the model only. The temptation, from an engineering point of view would be to take over control of wine naming protocols, but this would alienate the user and paradoxically cause confusion.

7.1.18 CURRENT DATA SYSTEMS

This enterprise has a simple network consisting of server and several client personal computers. All computers have internet access. The system also allows remote access via VPN (Virtual Private Networks) which allow access for other entities in the enterprise such as the vineyards, and for remote users.

The enterprise originally adopted a paper based system of record keeping after it was found that the commercial record keeping system they purchased was unusable. Eventually, the paper system was transferred to a simple electronic system using Excel worksheets. These files are used for all record keeping including...

- Vintage Plans
- Work-notes
- Wine parameters
- Barrel Allocations

These worksheets are used to record all the data necessary for reporting and auditing. The vineyards still employ a paper-based record keeping system including the obligatory "Spray Diary".

7.1.18.1 VINTAGE PLANNING

Below are some examples of some of the vintage plan documentation prepared by the winemakers before the vintage begins. The fruit is not yet divided into production of specific wines or even ferments, but it shows how each variety is expected to be treated, not just during the fermentation but also post ferment. The significant point to note here is that labels are roughly assigned, but the specifications are vague and some are left up to the blending process after vintage.

The first table (Table 6) is a yield estimate for the various blocks. It includes external vineyards to the enterprise as well as those under the enterprise's control. Each block is assigned to a general wine, but not a label. The Tonnes per acre column is based on the estimates provided by the viticulturist, but is updated closer to vintage as data comes to hand.

Vineyard	Zone	Block	Variety	T/Acre	T fruit	Sold	Winery	Wine
Heathcote	7	1a	Chardonnay	0.0	0	0	0	HECHAR
Heathcote	3a	5	Cabernet Sauvignon	2.8	27	0	27	HECAME
Heathcote	3b	5	Merlot	2.8	28	0	28	HECAME
Heathcote	4	2a, 3a	Nebbiolo	0.9	15	0	15	HENEBB
Heathcote	1	1	Shiraz	3.4	37	0	37	HESHIR
Heathcote	1	2	Shiraz	3.4	38	0	38	HESHIR
Heathcote	1	3	Shiraz	3.4	36	0	36	HESHIR
Heathcote	5	4a	Tempranillo	2.7	45	0	45	HETEMP
Heathcote	6	5a	Viognier	1.9	21	0	21	HEVIOG
Yarra		CC	Chardonnay	3.4	22	20	2	YVCHAR

Yarra		BC	Chardonnay	4.3	24	0	24	YVCHAR
Port Phillip			Cabernet Sauvignon	2.3	11	0	11	YVCAME
Port Phillip			Merlot	1.7	3	0	3	YVCAME
Yarra		CE	Cabernet Sauvignon	2.9	36	0	36	YVCAME
Yarra		MA	Cabernet Sauvignon	2.8	32	0	32	YVCAME
Yarra		MA	Merlot	2.35	33	0	33	YVCAME
Yarra		MA1	Cabernet Sauvignon	2.8	1	0	1	YVCAME
Yarra		MC	Petit Verdot	3.2	9	0	9	YVCAME
Yarra		MC	Pinot Gris	2.9	9	0	9	YVGRIS
Yarra		CA	Pinot Noir	2.35	35	0	35	YVNOIR
Yarra		BA	Sangiovese	2.9	30	0	30	YVSANG
Yarra		CB	Sauvignon Blanc	4.1	24	0	24	YVSAUV
Yarra		CD	Shiraz	3.1	52	23	29	YVSHIR
Yarra		MC	Mataro	3.3	11	0	11	YVSHIR
Yarra		BB	Shiraz	2.8	25	0	25	YVSHIR
Yarra		BE	Pinot Meunier	3.6	4	0	4	YVMEUN
Yarra		BE	Chardonnay	3.6	4	0	4	YVSPAR
Yarra		MC	Viognier	2.7	6	0	6	YVVIOG
Yarra totals				2.81494	464	63	401	
HC totals				2.87397	475	0	475	
Vintage				2.84449	938.5	63.0	875.5	

Table 6. Vintage Yield Estimates

This table shows that although each variety is associated with a general wine, scope exists to break these wines down to more specific batches. The winemakers choose to keep the wines generic at this point in order to remain open for contingencies. The next table (Table 7) shows how each general wine is to be treated during and after vintage. There is space to fill in actual parameters used to aid in subsequent vintages. You will note that these tables show more detail about possible destinations for the wine, but also reveal further indecision concerning pre and post fermentation specifics.

Code:	HEVIOG	
Volume:	10,000 Litres	
Destination Brand(s):	Best 5000 L for Brand X remainder for Brand Y	
Vineyard treatment:	Thin out to 2 tonnes per acre in late December, if necessary	
	Planned	**Actual**
Harvest:	Hopefully slightly earlier than 08. possibly two batches? One small hand-pick?	
Pre-ferment:	gentle press, skins to shiraz. Cold settle (or trial flotation). Rack and inoc. Bento? Pressings?	
Ferment:	cold ferment, start at 18C, down to 12 gradually, back up to 15C to finish. Yeast dependent on baume?	
Post-ferment:	partial malo?	
Elevage:	old barrels.	
Blending:	shiraz may need some.	
Finishing:	lots of oxygen, lots of bento, light fining.	
Bottling:	Nov-09	

Table 7. Wine Plan – Vintage Plan Example

The next table (Table 8) is a summary of likely yeast requirements based on the previous tables.

wine	HECAME	HENEBB	HESHIR	HETEMP	HEVIOG	YVCHAR	YVVIOG	total	rate	reqd	stock	order	cost	cost
total volume	38400	10800	237100	31200	12700	15500	3600	349300	mg/L	kg	kg	kg	$/kg	total
EC1118	0	0	49100	0	0	0	0	49100	300	15	34	0	39	$0.00
L2226	0	0	44000	5500	0	0	0	49500	300	15	6	9	61	$549.00
BDX	15000	0	48000	25700	0	0	0	88700	300	27	5.5	21.5	61	$1,311.50
UV43	23400	0	75000	0	0	0	0	98400	300	30	7.5	22.5	61	$1,372.50
Vin2000	0	0	0	0	12700	0	3600	16300	300	5	0	5	60	$300.00
BRL97	0	10800	0	0	0	0	0	10800	300	4	0	4	66	$264.00
QA23	0	0	0	0	0	13100	0	13100	300	4	5.5	0	66	$0.00
Vin13	0	0	0	0	0	0	0	0	300	0	3	0	60	$0.00
NT50	0	0	0	0	0	0	0	0	300	0	5	0	60	$0.00
Enoferm Syrah	0	0	21000	0	0	0	0	21000	300	7	0	7	66	$462.00
RC212 Pinot	0	0	0	0	0	0	0	0	300	0	0	0	66	$0.00
total	38400	10800	237100	31200	12700	13100	3600	312800		107	66.5	69		**$4,259.00**
to allocate	0	0	0	0	0	2400	0							

Table 8. Yeast Requirements

The yeast strains here are just a tiny proportion of those available to the winemaker. Notice how the excess of EC1118, the cheapest and most common variety used in this enterprise, is ignored and specific strains are used for specific varieties, regardless of cost or potential wastage. It should be noted that these decisions are not necessarily just a matter of wine style. Often specific strains are used to avoid problems such as high alcohol levels or sluggish ferments. Some of the wines, not shown in the table, did not have yeast selected at the point this snapshot was taken. Note too how the HEVIOG yeast has been selected, but the Wine Plan example *Table 7. Wine Plan – Vintage Plan Example* above had not been updated. Of greater significance is the "to allocate" row at the bottom of the table. This is an acknowledgement that the design of the plan is ongoing. Not all the decisions are made in a single sitting, because there are budgetary, marketing, and winemaking decisions that need to be reconciled.

The next table (Table 9) illustrates how oak allocation is planned, not just barrels but also chips.

Wine	volume	barrel vol	new %	new barrels	new vol	old barrels	old vol	tank	unallocated	chip type	chip rate	chips (kg)
HECAME	38400	1400	0	0	0	7	1400	37000	0	DCA	0.5	20
HENEBB	10800	800	0	0	0	4	800	10000	0	DC180	0.25	10
HESHIR	237100	15000	20	14	3000	54	12000	218000	4100	DC210	0.5	110
HETEMP	31200	4200	0	0	0	19	4200	27000	0	DC210	0.75	30
HEVIOG	12700	12700	8	5	1016	52	11684	0	0	none	0	0
YVCHAR	15500	3100	0	0	0	14	3100	10000	2400	DC180	0.2	10
YVCAME	87000	15000	33	22	4950	45	10050	64000	8000	DCA	0.5	40
YVGRIS	5600	7300	5	2	365	31	6935	0	0	none	0	0

YVNOIR	77200	15000	25	17	3750	50	11250	54000	8200	DC180	0.25	20
YVSANG	20900	5600	0	0	0	25	5600	15300	0	DC180	0.25	10
YVSAUV	15500	200	0	0	0	1	200	15300	0	none	0	0
YVSHIR	44600	15000	25	17	3750	50	11250	27000	2600	DCA	0.35	10
YVMEUN	2400	0	0	0	0	0	0	2400	0	none	0	0
YVSPAR	2400	0	0	0	0	0	0	2400	0	none	0	0
YVVIOG	3600	3600	20	4	720	13	2880		0	none	0	0
total		98900		81	17551	365	81349		25300			260
		440 barriques							112 barriques			

Table 9. Oak Requirements

The volume readings come from the yield estimate table, whereas all other parameters are entered directly into the table. Interestingly, this table reveals an insight into how the winemakers think about the process of allocating oak, and the sequence of the decisions they make. For each wine, of estimated volume...

1. How much will go to barrel?
2. What percentage of this amount will go into new barrels?

This establishes the number of old barrels required.

3. How much of the remaining wine can we allocate to tank?
4. What, if any, chip treatment do we wish to apply to this wine?

The remaining wine is categorised as “unallocated”. This allows for contingencies for variations in expected yields and eventually, actual yield deficits or excesses, and effectively makes the calculations for new oak requirements somewhat immune from these variations. As long as the volume of wine allocated to barrel or tank is consistent with the production plan, the excess becomes another issue in the vintage plan separate from the production plan. The calculations for available tank space are based on current availability and bottling dates for current wines. Similarly, the requirements for used red, used white and new barrels are calculated and compared with existing stocks and those that will be available in time. Deficiencies are then added to the new requirements list for the upcoming vintage.

The interesting thing to consider here is how an accountant or logistician might have approached this problem. The winemaker’s focus was on wine style and took an oenological approach to the problem by asking certain questions first, before considering what space and barrel resources were currently available. An accountant might take a very different point of view, by concentrating on current stocks of barrels (new and old) and likely available tank space, and allocating accordingly, *then deciding on what needs to be purchased for any excess wine. The difference might appear to be subtle and the eventual outcome might end up the same, but the paradigm is very different, and this has implications for the data model and the user interface in particular. Winemakers have a distinct motive for their decisions, and it is primarily an issue of wine treatment and quality. Economics and budgets are secondary in this case. If the end result is that too many barrels need to be purchased to accommodate the plan, beyond the budget, then the winemaker will go back and tweak the plan to see if a compromise can be made, but the emphasis here is on quality rather than expedience. The focus is on getting the correct balance of barrels versus stainless steel maturation, and the appropriate proportions of new and old oak, regardless of what is currently in stock for barrels and tank space.*

The next tables (Table 10 and Table 11) show expected use of common additives.

Wine	Tonnes	L/tonne	Litres	PMS (kg)	H2T (g/L)	H2T (kg)	Go ferm (kg)	Fermaid (kg)	DAP (kg)	MLF (g)
HECAME	55	700	38400	2	1.5	58	12	10	8	230
HENEBB	15	700	10800	1	2	22	3	3	2	65
HESHIR	339	700	237100	12	1.3	308	71	59	47	1423
HETEMP	45	700	31200	2	2	62	9	8	6	187
HEVIOG	21	600	12700	1	1.5	19	4	3	3	76
YVCHAR	26	600	15500	1	1	16	5	4	3	93
YVCAME	124	700	87000	4	1	87	26	22	17	522
YVGRIS	9	600	5600	0	1	6	2	1	1	34
YVNOIR	110	700	77200	4	0.5	39	23	19	15	463
YVSANG	30	700	20900	1	1	21	6	5	4	125
YVSAUV	24	650	15500	1	1	16	5	4	3	93
YVSHIR	64	700	44600	2	1	45	13	11	9	268
YVMEUN	4	600	2400	0	2	5	1	1	0	14
YVSPAR	4	600	2400	0	2	5	1	1	0	14
YVVIOG	6	600	3600	0	1	4	1	1	1	22
	876		**604900**	**30**		**710**	**181**	**151**	**121**	**3629**

Table 10. Additives requirements

Parameter	**Value**	
PMS rate	50	mg/L
Goferm rate	300	mg/L
Fermaid rate	250	mg/L
DAP rate	200	mg/L
MLF rate	6	mg/L

Item	Unit (kg)	Required	Stock	Order	Cost/Unit	Cost
PMS	1	31	11	20	3	93
Tartaric	25	29	20	9		0

Go-Ferm	1	182	0	182	44	8008
Fermaid	1	152	0	152	16.5	2508
DAP	25	5	8	0		0
MLF	0.15	25	0	25	900	22500

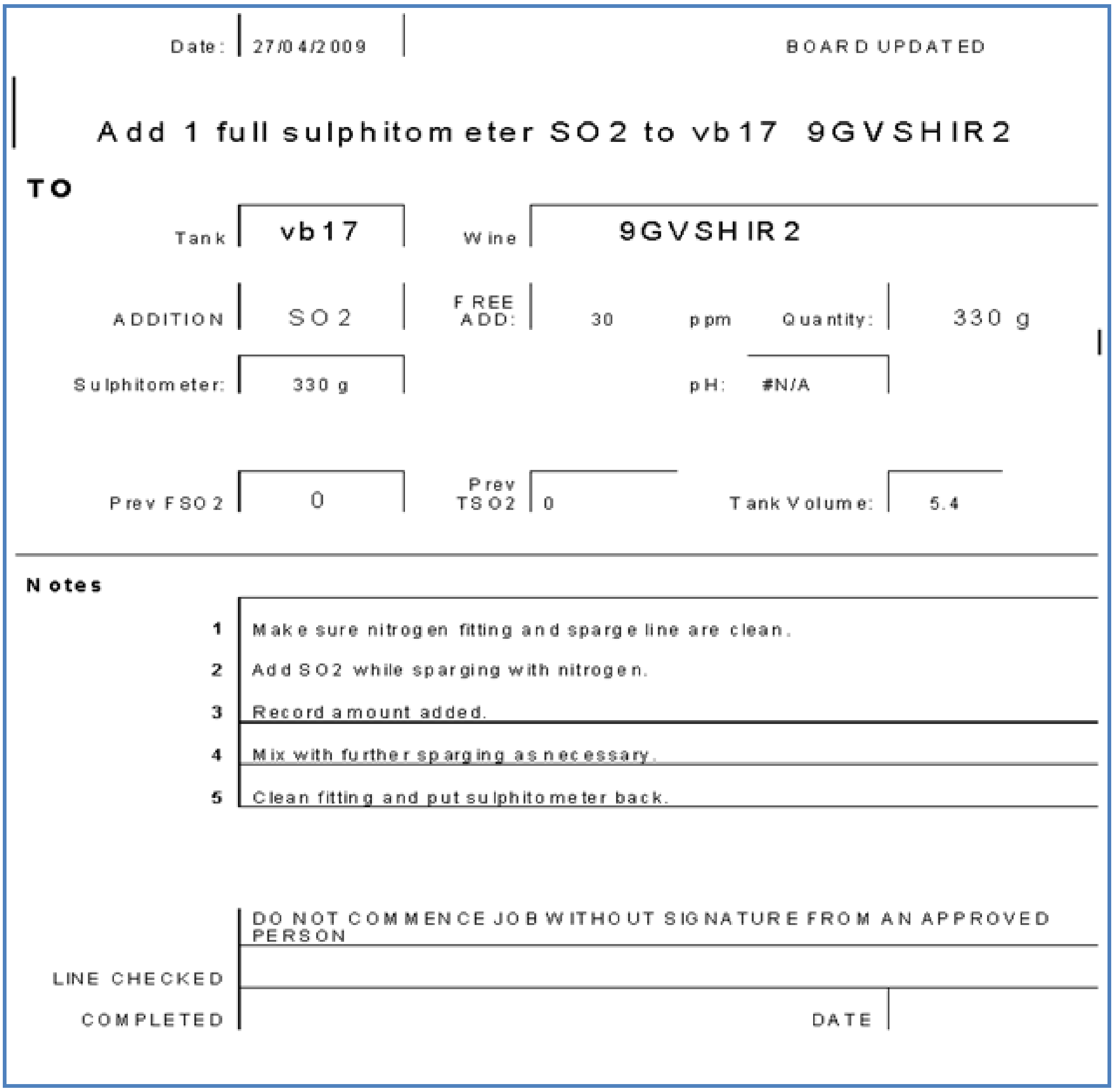
Table 11. Additives to purchase

The amount required of each additive is calculated based on flat rates appearing in the second table. Similar tables were constructed for enzymes and other additives on a wine-by wine basis.

It was interesting to note that even after these calculations were made, they were used only as a guide and significantly more PMS and significantly less Malolactic bacteria was ordered. So clearly there was more to it than just simply carrying out the calculations and following them blindly. Malolactic fermentation can occur naturally in wine, as long as sulphur levels are sufficiently low to allow the bacteria to perform. It has been found that recommended inoculation rates are excessive and this was the basis for dropping the required amount of bacteria to be purchased for vintage, particularly given the price. PMS on the other hand, is quite cheap and used for sanitising as well as an additive to wines. The final order placed was double the amount required according to the calculations. The flat rate of 50mg/l was deemed too low for some wines, given the experience of previous vintages. It would have been more appropriate to specify requirements on a wine-by-wine basis and then adding a requirement for sanitising tanks and lines separately. There is scope here for the model to take these contingencies into account. The fact that the enzymes and other additives were calculated on this basis shows that perhaps this should be made across the board for all additives.

7.1.18.2 Worknotes

The winemakers have developed a series of Excel templates to automate the generation of work-notes in the winery. They are structured in such a way that some of the fields are automatically filled in when the wine is entered (Figure 13). They also allow the creation of a simple audit trail to help keep track of the production of each wine.



Date: 27/04/2009 BOARD UPDATED

Add 1 full sulphitometer SO2 to vb17 9GVSHIR2

TO

Tank	vb17	Wine	9GVSHIR2		
ADDITION	SO2	FREE ADD:	30 ppm	Quantity:	330 g
Sulphitometer:	330 g			pH:	#N/A
Prev FSO2	0	Prev TSO2	0	Tank Volume:	5.4

Notes

1 Make sure nitrogen fitting and sparge line are clean.

2 Add SO2 while sparging with nitrogen.

3 Record amount added.

4 Mix with further sparging as necessary.

5 Clean fitting and put sulphitometer back.

DO NOT COMMENCE JOB WITHOUT SIGNATURE FROM AN APPROVED PERSON

LINE CHECKED

COMPLETED DATE

Figure 13. An example of a worknote from case study site

The example shown here is a sulphur dioxide gas addition to a wine in the tank designated VB17. The standard operating procedures are included with the work-note but space is provided to add extra notes and comments. Once the work-note is completed, the soft copy is filled in with relevant information to complete the audit trail. Fields such as pH, volume, and previous free SO_2 are automatically filled in when the work-note is created. When the free addition field is filled, the spreadsheet automatically converts this to grams of sulphur, which is the scale used on the sulfitometer.

The major criticism for this system is the fact that it is very cumbersome to use and difficult to glean useful information. The simple partial transfer of wine from one tank to another necessitates the creation of several work-notes so that the system is able to function properly. In practise, the only *person who really understands how the system works is the winemaker. The system relies on a crude algorithm to match names in order to match records. It utilises the primitive facilities of Excel to perform as a pseudo database. The winemaker acknowledges the weaknesses in the system and the need to constantly keep it updated in order for it to function. The problem seems to have come about because it was developed ad-hoc, and modified as the required functions became more sophisticated. It is a reflection of the fact that winery operations are never as simple as might be assumed by an outsider. Quite appropriately, the winemaker has to adapt the system to cope with reality rather than restricting himself to simpler operations for the sake of the system. The implications for the model are quite simple. It must be designed so that it easily and properly accommodates the actions and intentions of the winemaker. It was refreshing and revealing to hear the winemaker say that the system needs to be centred on the wine not the operations and actions. This would seem logical given the fact that from the outset, the vintage and sales plans focus on the wines and all the actions in the winery are based on achieved a desired effect on the wine. However, one must also acknowledge another requirement to allow these work-notes to be applicable to more than one wine in any given instant. As mentioned earlier, one can envisage a general standing order for all ferments to be plunged or pumped-over every 6 hours. This is an action- centred work-note, and the structures within the model will need to reconcile this shift in focus from wine to action. Fortunately this is a feature capability of a well designed object oriented data model.*

7.1.18.3 Wine Monitoring

There are two main spreadsheets used to store the essential parameters for each wine; one for wines in tank, and one for those wines in barrel. These parameters include...

- Wine Code - wine name
- Tank – Vessel name
- [1]Barrel Count
- Wine Volume (kL)
- tasting group (Manually assigned number to group like wines for comparison)
- [2]cold stable (Yes or No)
- [2]heat stable (Yes or No)
- FSO2 ppm
- TS02 ppm
- Date SO2 – the date the sulphur levels were last tested
- pH
- TA (g/l)
- Date PH/TA – the date the pH and TA readings were made
- VA (g/l)

[1] Barrel spreadsheet only
[2] Tank spreadsheet only

- Date VA – the date the VA reading was performed
- Sugar (g/L)
- Date sugar 0 the date the sugar reading was made
- Malic (g/l)
- Date Malic – the date the malic acid reading was made
- [2]Micro-oxygenation – Yes or No, whether or not a Micro-Oxygenation sinter is active in this wine.
- Lees - Yes or No, whether the wine is on lees or not
- [2]Filtered – Yes or No, whether the wine has been filtered or not.
- [2]Scum – a general comment field for the condition of the wine
- [2]Diffuser – Yes or No, whether a diffuser is in place to protect the wine on ullage
- [2]Temperature °C
- [2]Chips – type of chips currently in the wine
- Allergens – list of additives or fining agents used which need to be included on label
- Ferment / Malo – indicator of fermentation status, F for fermenting, M for Malolactic
- Vintage – Percentage vintage proportions for LIP
- region– Percentage regional proportions for LIP
- varieties – Percentage varietal proportions for LIP
- plan – proposed and on-going processes for the wine.
- [1]Top With – Name of wine used to top this wine

This data is updated by the winemaker and laboratory staff as they complete analyses. Other spreadsheets are used for the analyses calculations but these results need to be manually inserted into the wine spreadsheets.

The rationale for these spreadsheets is not just for recording wine parameters but they also allow a simple hazard scoring system for the wines to alert the winemakers of any issues with respect to common maintenance tasks for the wines.

Points are added to a wines score if certain criteria are met. The higher the score, the greater the "hazard level"

- 2 hazard points for free SO_2 level below 10ppm
- 1 hazard point for free SO_2 level below 19ppm
- 1 hazard point if more than 28 days since last free SO_2 analysis
- 1 hazard point if more than 28 days since last tasting
- 1 hazard point if pH above 3.7
- 1 hazard point if VA above 0.8
- 1 hazard point if VA increases by more than 10% per month
- 1 hazard point for last racking date greater than 90 days ago
- 1 hazard point for topping date greater than 28 days ago

Given the fields available within each spreadsheet, this algorithm can be modified and tailored depending on what the winemaker regards as hazardous. The system as it stands uses the same algorithm for all wines within each spreadsheet. In a sense it is a simple HACCP system that the winemakers refresh each day and use to prioritise cellar operations. In software engineering terms, this algorithm is similar to those adopted in Customer Relationship Management (CRM) systems where customers are "managed" according to a combination of known parameters and demographic data and assumptions.

The scope for improvement here is enormous and offers the opportunity to marry two major requirements for the data model. As a HACCP system it is easily adaptable in terms that are simple for winemakers because it uses parameters and situations that are constantly referenced on a daily basis. If properly designed it could also provide an intuitive interface to a Data Mining function. The idea would be to provide extended access to data for inclusion in the monitoring algorithms and at the same time offer graphing functions to plot these parameters and provide statistical information.

The main point to take from the description of these spreadsheet based systems is the lack of data integration which otherwise allows important information to be used outside the context in which it was received. The vintage plans are manually referenced in order to generate work-notes for processing fruit. This means there is no tangible link between plans and implementation, and therefore the ability to verify plans and benefit from comparing plans to outcomes. The monitoring scoring system is the sole exception to this, but it is relatively clumsy and inflexible to use. However it does provide a starting point for development into a tool that allows winemakers to prioritise their efforts and avoid serious errors.

The other significant point to note is that so much of the specifications for the plan are tentative and non-committal. This is not a criticism, because some of these decisions cannot be made until closer to the harvest, or during the fermentation or even after the fermentation is complete, if not well after. It provides an insight into the winemakers' intentions but that they are always open to contingencies. The model will need to provide for such non-committal information, and provide a range of settings for various parameters ranging from likelihoods to specific rates or settings. This implies that vintage requirements will also need to be very flexible with calculations. For a software engineer, this is annoying and difficult to cope with. For a winemaker, this is reality.

7.1.18.4 The Vineyard

Interviewing the vineyard manager at Heathcote provided the most compelling evidence for the need to build a specialised system. The manager himself was so frustrated by the fact that all the commercial systems he was able to evaluate still fell short of providing the required functionality to allow him to properly manage the vineyard, he had resorted to building his own database system. “They’re all based mainly around the finances of the industry”. There was no perceived provision for viticultural date such as recording dates for vine status milestones such as budburst or veraison. He needed something that would allow him to record and refer to activities from current and previous vintages so that he could make informed estimates and inferences of activities on fruit quality and quantity. The development of his system was an on-going process that was still in the early stages from a SDLC perspective but it was being as part of the data management system for the vineyard. However, the main system still relied on the ubiquitous white-board, filing cabinet, and newsagent-purchased diary.

7.2 Case Study 2 – Very Small winery

This organisation represents the smallest scale of the enterprise studied in this thesis. The enterprise consists of a small winery producing approximately 200 cases per year, and cellar door.

Fruit is sourced from local suppliers, although the property does have a small vineyard which may or may not be used as a source of fruit in the future. The winemaker-owner has organic winemaking techniques at the heart of his philosophy. His technique relies on a traditional, non-interventionalist, minimal-sulphur approach, but references modern technology when appropriate. He admits to having shifted slightly from this ideal lately because of the perceived high risk associated with such techniques, but has not lost sight of his goal; he “just needs to try and find a balance”.

Only producing Shiraz wines, fermenting is done in stainless steel vats, then transferred to French oak for maturation, and hand-bottling on site. Temperature control is achieved by the simple adaption of a freezer providing coolant in copper coils wrapped around the vessels; a technique common to several small and boutique wineries encountered in this study.

The scale of this operation, along with the relatively early stage of development, means that winemaking is hands-on and carefully monitored. Although the winemaking is kept simple, this does not mean that complex problems are avoided, and the implications of any problems that arise have greater significance for smaller enterprises because a taint in one barrel might mean losing a significant proportion of a wine.

There is no automation as such, and only the basic laboratory analyses are on-hand. The small batches of wine involved means that topping regimes become quite complicated because even small additions have a greater significance to LIP issues for small batches of wine, and there is less topping wine available to choose from. It also has implications for oak balance because there is less flexibility with small volumes. An interesting fact that arose was the use of barrel shaving to regenerate barrels. Costing about $70 to renew a barrel, this process can be repeated to give a single barrel three lives, at least from an oak contribution perspective.

The owner readily admits that he is on a particularly steep stage of the learning curve; necessitating a diligent monitoring of all aspects of operations, from production to sales, and planning for subsequent vintages. He also readily concedes that although he recognises the importance of using appropriate tools to perform these functions, his current techniques are only just adequate and will need to be radically improved, particularly as the business looks to mature, consolidate and potentially expand and diversify. Currently, he uses a paper and spreadsheet based system for recording activities and data in the winery, whereas planning and future strategies are not always maintained formally.

Of particular interest is the need to come to terms with factors that he has most control over. There are several sources of fruit available to him, and as his experience with each increases, the more this control becomes apparent. This is most important given the perceived lack of oenological knowledge and support provided by the suppliers.

The simplicity of the winemaking reflects the simplicity of the enterprise, and this sets a base-line for the functionality of the model and the interface to accompany it. It illustrates the need to ensure that the data model should not be too automated, or make assumptions about the winemaking process (yeast or MLF inoculation for example) or assume facilities such as a laboratory are present or utilised. If the model uses this enterprise as a core for the winemaking process, then we also need to look at an example of sole vineyard, without a winery as a base-line for the viticultural management aspect of the model.

7.3 Case Study 3 – Small vineyard

This enterprise consists of a small vineyard of 82 acres of predominantly Shiraz vines planted in 2001. All their fruit is generally sold to two main client winemakers although they are looking to occasionally make small batches by themselves, with the aid of another winemaker in the region.

The vineyard is divided into the traditional blocks and rows, but as a dedicated grower, the owner has come to recognise areas of higher and lower vigour within and across these divisions. These areas need to taken into account for accurate yield and parameter measurement for sampling, particularly with relatively small block areas.

The two major fruit customers each have dedicated blocks within the vineyard. One has simple requirements to supply a certain quantity of fruit with certain parameters, leaving harvesting decisions up to the owner. This has occasionally led to undesirable delays in harvesting because space was not available at the customers winery, leading to deterioration in the fruit. The other major customer has a more hands on approach, and is positioned at the premium end of the market. This customer has higher demands on attention to the quality of the fruit, and pays a higher price as a result. This has led to a decision to certain blocks being transferred from the former customer to the latter; which has resulted in a new regime of pruning and maintenance on these blocks. The transition from one regime to another has meant a transitional regime has been developed to allow the new block to catch up.

The owner finds that his business is getting more complex, and he says that he tends to be spending too much time performing administrative duties, and less time performing what he regards as important core-business duties. The record keeping system currently employed is a simple paper and spreadsheet based system. The owner recognised that part of his problem would be solved by having a system that integrates all aspects of his activities and plans.

The concept of block and rows is a convenient means of identifying distinct areas in a vineyard. However, in practical terms and therefore for the data model, it is inadequate to simply use these divisions as a basis for a parcel of fruit to be processed. At this and other small vineyards, winemakers and viticulturists occasionally like and need to be far more selective in the way they treat and use the vines. These decisions can be made for viticultural, oenological, wine-stylistic, or business related. For example, it is possible that more than one pass is made through a block during harvest. This might be to selectively hand harvest the best fruit before mechanically harvesting the remainder. From a data modelling perspective, this is inconvenient and complex because the basis for the identifying a batch of fruit can shift from vintage to vintage.

The solution to this issue is to make the model and interface flexible enough to cope, so that it uses more than a simple block identifier to indicate the source of a parcel of fruit. The same issue exists for other entities, such as plans, work-notes and regimes that might apply across more than a single block. The implications are that a direct year to year comparison of fruit may not be possible, but this is unavoidable unless a suitable design strategy can be engineered.

The other issue to consider is the degree of management and intervention that is applied to different blocks. The model needs to be able to cope with extreme micro-management and detailed notes for a block but not expect or rely on this for all blocks. The user should not be inundated with hundreds of questions and fields to fill in if that degree of detail is not relevant, or if they could not be bothered. The interface needs to be gentle enough not to scare away non-interventionalist users but still be powerful enough to cater for situations where detail is important. This is not just a design philosophy but also has implications for the data model in terms of flexibility of functionality.

7.4 Case Study 4 – Small enterprise utilising contract fruit and production facilities

This small winemaking enterprise produces approximately 180 cases a year of each of five varietal wines. The enterprise purchases the fruit from a cool climate vineyard and it is delivered to a winery for production under contract. The key to the success of this arrangement is the fluid communications between the winemakers, the grower, and the winery management. (Figure 14.) In fact, the enterprise in this case study consists simply of the winemakers and a storage facility. The winery and the vineyard are external to the enterprise.

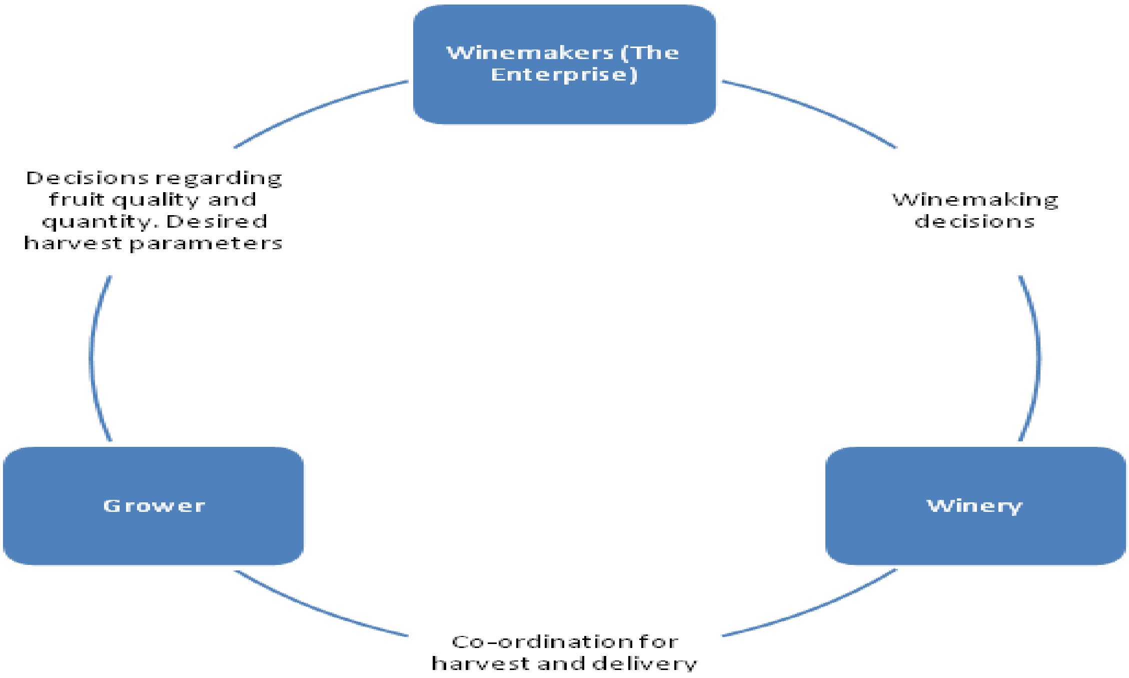

Winemakers (The Enterprise)
Decisions regarding fruit quality and quantity. Desired harvest parameters
Winemaking decisions
Grower
Winery
Co-ordination for harvest and delivery

Figure 14. Interaction between entities in the winemaking process

In general, the winemakers are across all aspects of the production process, apart from the practical logistics associated with harvesting and delivering the fruit, which is more efficiently handled by direct communication between the vineyard and the winery. The relationship between the winemakers and the winery management is based on a wine style rather than the winemakers having to micro-manage the whole process. The winemakers generally only need to communicate with the winemakers on major aspects of the process such as additions, fining, filtration, oak choice, rather than directly supervising or performing the process. Agreements are made concerning the regularity of analyses and other routine actions so that the winemakers are only concerned with major decisions about wine-style. The winemakers often take advice from the winery's winemakers anyway, because these staff are often better informed and have a broader experience with the idiosyncrasies of the equipment and materials in the winery. Tasting and fining trials are held usually with both parties present. A major advantage of this concept is that the winemaker is constantly aware of the cost of production because contract wineries tend to have a well defined cost structure. Another advantage for the enterprise is the fact that processes such as bottling can be coordinated with other enterprises using the wineries facilities, thereby avoiding some cost penalties associated with small batch processing.

As one might imagine, the very nature of this enterprise is quite different to most other enterprises, which tend to be based on at least one entity of infrastructure such as a winery or a vineyard. This case study introduces a new concept to the data model; "Contract entities"

Why do we need to make the distinction between entities owned and managed by the enterprise and contracted entities such as fruit suppliers or external contract winemaking facilities? The answer is a matter of data ownership, management responsibility, accounting, and user interface perception and presentation.

One of the core foundations of this thesis is the belief that important information and data relationships are inherent between the vineyard and the winery. If one of these entities is under the direct management of the user of the data model, then there are certain functions that need to be included in the model that will help with this management burden. Nevertheless, even entities that are not under direct managerial control still have parameters and data that have an impact on this relationship and the flow of information around the production cycle. For example, in this case study, the winemakers are not responsible for collecting and analysing fruit samples prior to vintage, but that information is very important for the integrity of the data. The difference is that the data model does not need to provide scheduling or work-note orders for the sampling, only the data that comes from it; probably only available several days after the analysis has been performed. In this case, fruit suppliers cannot be regarded simply as suppliers of materials like dry goods or chemicals. There is an intimate relationship between the fruit and the final product that means the winemakers need to know much more about the fruit than just variety and yield. The enterprise does not necessarily have *direct management control over the vineyard, but it certainly has input and needs to get feedback. The consequence of all this is that if the data model is to define the significant entities involved in an enterprise, some of the entities will be external to the enterprise. If we call these "contract entities" and flag them as such, then the data model will know to treat them slightly differently because*

- *The data associated with the entity is in effect second hand*
- *the user of this instance of the data model does not have direct management of the entity*
- *The costing for this entity does not need to be managed (it arrives in invoice form)*
- *The user interface will not need to present the entity in the same way*

7.5 CASE STUDY 5 - MEDIUM SCALE WINERY WITH MULTIPLE EXTERNAL VINEYARD SOURCES

This enterprise provides an interesting case study because there are several features to the organisation that make it more structurally complex than any other studied. As a result, it represents the upper extreme of the scale of enterprises within the scope of this project.

The winery is partially owned by 35 local growers in a co-operative. They crush up to 3000 tonnes per annum and provide minimal contract winemaking for other winemakers. They have several labels, some of which are truly iconic within the Australian industry. The enterprise has its own viticultural staff who not only manages the vineyard immediately surrounding the winery but also act as consultants to the constituent growers.

The winery started out by accepting all the fruit supplied to them by the growers, but this became problematic because of the vast quality and quantity discrepancies involved. Their premium brands require a standard of fruit that was not always achieved or possible by the majority of the growers, and so the necessity to protect these brands meant much of the fruit ended up in bulk wine and the volume of premium wine fell short of production plans. The enterprise does not regard bulk wine as core to their business, so a new understanding was put in place between the growers and the winery to match the supply and demand of premium fruit.

Fruit is selected based on consistency over several years. Over 200 batches of fruit are processed separately for some varieties, before they are blended. As a result they have developed a thorough management process to test and monitor the vineyards, the fruit, and the wines at all stages to ensure the standards are maintained.

The vast differences in soils, microclimates, yields, viticultural management styles, amongst other factors, mean that each vineyard needs to be treated individually. The viticulturist employed by the enterprise advises each grower to a lesser or greater extent based on these factors. Some growers are very hands-on and selective about pruning, weeding, irrigation, fruit thinning and many other viticultural management practices. For example, more enthusiastic growers divide their blocks into regions of high or low vigour and will use these divisions as a basis for the batches of fruit they intend to harvest. Other growers simply stick to a consistent routine year after year without much variation at all. It should be noted that the intensity in managerial style is not necessarily proportional to the eventual quality of the fruit. It was found that there was surprisingly little consultation between the growers themselves, which might be regarded as healthy competition but might also be regarded as slightly dysfunctional. The viticulturist for the winery is often utilised as a go-between for some growers, and this can be used as a means of sharing techniques and ideas.

As one can imagine, a complex monitoring system applied to a diverse range of entities requires a well defined and flexible process. The viticulturist monitors and records all the landmark stages of fruit development in the vineyards, including flowering cap-fall, bud burst, veraison, etc. Fruit is graded into one of six levels, at all stages of production from pre vintage sampling to pre crushing. A similar standard is applied to the individual wines as they are developed during fermentation and subsequent processing. As with all wineries, scheduling is very important to ensure the efficient use of resources as well as maintaining production quality. For this enterprise, scheduling is critical to their success.

As far as existing data systems are concerned, the ubiquitous spreadsheets are used to store all the data and a whiteboard is used to provide an overview. The winery uses a common commercial winery management system. Through necessity, the application of these tools is more advanced than most other enterprises studied. To a certain extent, rather than just being a means of recording data for later reporting, some of this information is used closer to harvest as winery plans are finalised, and some is used at harvest or during early ferment. For example, tasting notes of berry samples are used to assign likely grades for batches of fruit. This grade is constantly adjusted whenever the fruit, juice or wine is tasted. Sometimes these observations in the vineyard are also used to trigger specific analyses at harvest and early ferment too and cross referenced with the tasting notes. Further examples of re-using data to enhance observations were the use of bud-burst dates as an early indicator for harvest dates, as well as the use of primitive data mining and analyses to discover trends in the data to attempt to quantify and qualify various parameters and their effect on wine quality. Ultimately, using selection for premium labels as wine quality indicator, several interesting trends were observed regarding yield rates. They admit that such analyses would be performed far more often if time and simple access to data was available.

The enterprise investigated existing commercial systems to find a system that could be all encompassing as far as record keeping for the vineyards and the winery. From their perspective these systems do not have adequate viticultural coverage. To this day, they still use a plethora of spreadsheets to try to manage the shortcomings of these winery-centric systems. As far as they are concerned, the physical actions and details of growing grapes and producing wine is well established and settled, based on the industry standards and the personal preferences of the decision makers such as the winemakers and viticulturists. The problem they have is the fact they regard the input of data as a separate process, requiring a separate mindset and mentality, quite separate from what they know and think of as the winemaking process. In a sense, this achieves one of the goals of the data model. It should not impose on the core business of the enterprise by modifying the paradigm for its own sake. Unfortunately, to achieve this they have to isolate the data management aspect of their duties as a separate task, rather than it being a natural consequence of the task at hand.

This enterprise highlights several important considerations for the data model and raises a few issues.

- The importance of integrating vineyard and winery functions
- The nature and intensity of the relationship between entities in an enterprise
- The definition of entities internal and external to the enterprise
- Further justification for data mining functionality
- The use of subjective data in practical terms

The consequences for the data model from this case study are significant. Although the vineyards could be regarded as part of the enterprise, they cannot be said to be under the direct control of the enterprise. The data model design needs to recognise that, in this case, the majority of vineyards are not under direct management of the users of the model, but at the same time, they are more than just suppliers of fruit. There is a symbiotic relationship that means there is an intimate exchange of information between an individual vineyard and the enterprise as a whole. But because management is decentralised, the instance of model at the central office cannot assume it has access to data and functionality associated with the management of the vineyard, and at the same time, it does not need to provide that functionality within the context of this instance of the model. Certainly this information is nice-to-know, but the winery does not need to know because it does not have direct responsibility or management of these functions.

The differentiation between this type of relationship and that of a vineyard that is wholly part of an enterprise may be subtle, but it is also profound and the implications are significant. If the model assumes that any action performed in the vineyard must be associated with a schedule, a booking, or a plan, then it becomes cumbersome to have to enter this information if the user has no say in its creation, and is only interested in the fact that the action occurred.

In effect this situation can be reconciled by acknowledging that the data model should not assume that it is the central control point for the entities within the enterprise. Case study 6 is a good example, because neither the vineyard nor the winery is directly controlled by the managers of the enterprise. The solution to this issue is that it cannot be assumed that the vineyard owners and managers have access to the same instance of the data model. This demarcation of management and therefore of responsibility for planning and recording processes performed in the vineyard means that the data model needs to recognise a different style of relationship between entities within the enterprise. It makes no sense for the data model to be the central storage point for plans and management strategies when those decisions are not being made by users of the model. Each vineyard could have an instance of the data model and use it in the same way that other enterprises use it to manage a vineyard under their direct control. For this enterprise, the instance of the data model used at the winery would have to reference its own copy of the vineyard entities differently. The model would not need to provide functionality for scheduling spraying or pruning or irrigation, for example, although the post-facto data associated with such processes (date, time that it was performed) would be useful. But this also means that it cannot assume that data associated with this functionality will exist to be used for data mining or for functions outside the contextual functionality.

7.6 Case Study 6 – High-end Medium small winery and vineyard

Perhaps the great highlight of doing this research was to be given the opportunity to interview Dr Bailey Carrodus, owner, viticulturist, and winemaker at Yarra Yering vineyard in Gruyere in the Yarra Valley, Victoria. The Yarra Yering winery is one of the great and iconic winemaking enterprises in Australia. This is entirely due to the talents of Dr Carrodus whose insight into terroir, viticulture and winemaking, along with a stubborn determination to produce wines of such wonderful style and quality, set him amongst the elite of winemakers in Australia. Dr Carrodus died about 5 weeks after the interview.

The inclusion of a high end winemaking enterprise offers this research a broader perspective on the management, marketing, and winemaking styles used in Australia. This winery and surrounding 72 acres of vineyard has been established for about forty years. It is notable for the high quality of the wines, the simple small batch processing, and the fact that the complete production process from growing to packaging takes place on the premises, all under the control of the manager who is both viticulturist and winemaker. There are only four or five permanent members of staff, all of whom perform all aspects of the production process. All the fruit is hand-harvested from the local vineyard, and handled in the small winery using small scale presses and tanks. Reds are lightly crushed and fermented in 300 kg stainless steel open cubes and are hand plunged. There are up to 30 separate red ferments done each vintage, and the resulting wines are later blended into the various wines and put in oak for anywhere between 6 and 12 months. The white wines are lightly crushed and pressed into 3 tonne stainless steel potter tanks. They are also put into French oak and matured for up to 12 months before bottling. Vintages are left in bottle for 12 months before release.

This enterprise had originally been established in what might be regarded as a more innocent age; before government bureaucracy and economic business sophistication were given any significant priority. The single vineyard mentality was the common model of the time, and wine quality was the driving force and inspiration for the philosophy and activities of the enterprise. Australia has no appellation system that imposes restrictions on grape variety or scale, so the business was established without any bureaucratic restrictions. The founder carefully selected the site and grape varieties and clones, and set about growing Bordeaux, Northern Rhone and Burgundy styled wines with the intention of having them mature over decades, not just years.

The winemaker was quite sceptical about the use of a data model to manage what he regarded as a fundamentally simple process. However he was also the most aggrieved about the burden of record keeping for L.I.P. and other bureaucratic necessities. Certainly, for this enterprise, the winemaking process is simple, relying on tried-and-true techniques to produce the wines. However it is the peripheral business and logistics that make it complicated. The owner recited several examples of how packaging and labelling issues have arisen because of logistical errors.

This has two major implications for this project.

1. The enterprise has been able to dictate to the market both wine style and quantities which means that their production is based on available resources, not a market plan.
2. The production system is well established, and changes very little from year to year, and so stands to benefit very little from the proposed improvements of the model

The consequences of each of these points are discussed in the next chapter.

The ability for this enterprise to dictate to the market has softened over the last decade. This necessitated the acquisition of a local distributor, and expansion of cellar door activities to compensate for stock no longer selling out annually. However the enterprise under new ownership as a result of the death of Dr Carrodus, is still regarded as iconic, and remains a prime example of a business that is more quality driven than market driven.

8. CASE STUDY SUMMARY AND IMPLICATIONS FOR THE DATA MODEL

The following chapter highlights some of the main lessons learnt and issues that have arisen from the case studies. They are divided into three distinct types...

- **Philosophical** – Fundamental issues to do with the general aims, boundaries, and scope of the model
- **Practical** – Winemaking and production issues that need to be addressed and clarified.
- **Systematic** – Modelling issues to do with technical considerations for the data model's function and performance.

Many issues have already been raised within the context of the case studies and do not need to be repeated. Chapter 9 '*User requirements – from the top, down'* also addresses some more specific issues and how they are to be resolved.

8.1 PHILOSOPHICAL ISSUES

8.1.1 WINE INDUSTRY PROFESSIONALS

Although it is not within the scope of this project to go into any depth into the philosophical mindset of "The Winemaker", it is relevant to note a few observations that resulting from research and interviews. Naturally, it is improper to generalise and make assumptions about a classification of people, but there was a common thread that occurred during the interviews. These points, illustrate some of the inherent difficulties encountered during the process.

i) Winemakers were generally conservative in their philosophy, regardless of their age. This is understandable given the variability of their environment and the market. However, this sometimes created a sceptical or cynical atmosphere during discussions.

ii) Some winemakers were "institutionalised" by their existing management systems and found it difficult to think laterally about what they want and expect from a management system without resorting to requesting functionality similar to their existing systems, even with the inherent short-comings that entails.

iii) As a result, it proved to be difficult to establish a common language of discussion and debate concerning the proposed functionality and interface behaviour. It was often necessary to lead discussions in appropriate directions when they got bogged down in irrelevant details concerning interacting with a system that had not yet been designed.

iv) A similar “institutionalisation” was observed in winemaking practices where previous experience was used as justification for an otherwise contextually inappropriate or outdated action or procedure contrary to the recommendations of the designer or manufacturer of the material or equipment in question.

v) When confronted with a flaw or miscalculation in their own logic, some winemakers tended to go on the attack and try to defend their position rather than retreat and accept logical, mathematical, or geometrical certainty.

vi) There was a willingness to accept opinions with which they agreed, without scrutiny, but a dogged determination to reject ideas that they did not understand, also without scrutiny.

vii) On more than one occasion, the winemaker believed that the subject and nature of this thesis was trivial and that it should be resolved within a week.

Upon reflection, it must be said that there were only a handful of participants in this project who really had a complete understanding of the majority of the issues that were encountered as the discussions took place. Many of the significant issues that will be discussed later, are only relevant to enterprises at the larger end of the scale where more complex situations are likely, and only a few of the participants were in a position to have had to make decisions in those circumstances. It is perhaps not surprising that these people were by far the greatest contributors to resolutions and had far greater insight into the intricacies of the project.

The nature of the winemaking process is such that judgement, interpretation and decision-making is nearly always on a subjective level. Unlike truly scientific endeavours, which necessarily base their path on objective evidence and measurable parameters, winemakers commonly deal in subjective data, and perhaps this is why they are perceived as very opinionated; a quality they freely admit to. In a regime where nothing of relevance is definitively and objectively provable, winemakers must necessarily be assertive to justify their stance. Indeed, evidence from the ethnological studies suggests that subjective concepts were often confused with objective qualities. It was common to observe one opinion being tended as fact on one day and the contrary opinion being tended the next, by the same person. When confronted with this apparent contradiction, the winemakers became defensive and dismissed the criticism as being taken out of context, or misinterpretation. The issue here is that subjective opinion cannot be used in the same way that observable, objective truth is to justify a belief, but such is the nature of winemaking, subjective belief is proffered as objective and empirical evidence.

In a sense, this subjective basis for the winemaking paradigm means that the focus of effort tends to be on creating an end result rather than building a mechanism to achieve it. In other words, if the process was predictable, formulated and reliable to produce the desired result then all the effort and emphasis would be on engineering the process. Winemaking, and therefore winemakers, tend to be more focussed on an end result, and so the means of achieving it is not as important so long as the goal is achieved. At the same time, because the raw ingredients and their behaviour and responses to the treatments differ from batch to batch, the production process is more intensive and subject to intervention than most other food production processes. It is a subtle difference, but perhaps it goes some way to explaining the incompatibility between thinking styles of winemakers and system architects.

It must also be acknowledged that there is a danger of adopting too much of the winemakers philosophy to the extent that it affects the design process. We have made it quite clear that the users' requirements are paramount and the end result for the model is the focus, but it will be necessary to be vigilant when pure programming or computer issues arise, that they are dealt with in a technical way, not in the manner of a winemaker. The value of the engineering paradigm whereby primary concern is for the building process, is that it makes the result consistent, standardised, easily maintained and reliable. The last thing we want is the data model or the application to behave like a wine.

In addition, the culture of each winemaking enterprise is a product of the history and philosophy of the owners and managers, and must be treated individually. Some individuals are open and invite scientific scrutiny in an effort to improve the process, but others are not. Much of the development process for this model will need to be less of a compromise between the software development world and the winemaking world, and more a search for compromise between disparate winemaking philosophies. Concepts such as "options", "defaults", "configuration" come to mind.

8.1.2 What aspects of the process can be improved by this model?

Although this was discussed earlier in Chapter 2, the aim of this model was reinforced during the ethnological study. Above all, it is necessary that the system operate not as a distraction from the practical process of winemaking but rather as a reflection of the various mental modes of operation of the winemakers, by acting as a tool that can provide contextual insight into the state of the enterprise and offer an intuitive mechanism for instigating processes without interfering in the operations themselves. It was particularly noteworthy that some actions could be instigated by several very different causes, and that the cause had great significance but this was lost because it was not recorded. By the same token, non-action was often justified from a winemaking point of view, but decisions such as these were never recorded. There needs to be a change from merely recording actions to recording decisions and implications.

8.1.3 What aspects of the process could be adversely affected by this model?

It is all well and good to start out with high aspirations for solving a problem or improving a process, but the very act of doing anything may cause more damage. It became most apparent with some of the case studies that modelling these enterprises might be a massive over-kill if it necessitated excessive interaction. Although it has just been argued that the cause and result of an action need to be recorded, the processional framework that the model imposes may need to have optional steps. For example, an ad-hoc pH and T.A. test at a small winery such as case study 2 does not need the same level of interaction that a standing order to do a fortnightly analysis on a wine in case study 1. For a start, the winemaker and the laboratory technician is the same person in the former, so the winemaker is more than likely going to be inclined to reach for a sample container rather than go to the computer to generate a request before actually doing the analysis. In actual fact, this is more of an issue for the interface design than the model. The model only requires the appropriate data. (*8.3.1 The Role of the Interface*)

8.2 Practical Issues

8.2.1 Make what you can sell or sell what you can make?

The first issue that needs to be addressed here has important repercussions for the design and implementation of the planning component of the model. This issue is to do with sales/marketing plans, and available fruit, and which defines the other. Obviously if the enterprise is just a vineyard then this question is moot, but all enterprises that have a winery component need to have both a sales plan and available fruit available as a starting position for the model.

This issue arises because different wineries have differing positions in the market. Elite, boutique wineries are able to dictate wine styles and volumes to the market because of their exclusivity and reputation. There appears to be no relationship between supply and demand, at least not at the operating volume of these wineries. On the other hand, well established wineries that operate within the main commercial market where competition is higher, need to listen to the market and adjust their targets accordingly from year to year. It is also interesting to note that some young enterprises have to follow an identical strategy to the boutique wineries because their sales and marketing strategies are not mature enough to respond to market forces. This is not the recommended commercial strategy, but it may be necessary in order to get their foot in the door.

In practice, it appears that there needs to be a combination of available fruit defining the wines to be made, and commercial and marketing factors determining what fruit is needed to produce appropriate wines. This dictates that the model will need to deal with both, given the fact that we want to make the system flexible enough to deal with both extremes, and more often than not, a combination of the two.

So how should this impact the model and the interface? Those involved in working with these strategies inevitably think in terms of labels and how they can best utilise their facilities and resources to maximise revenue, according to the label's market position and associated price point. If the market demands are paramount then sufficient available fruit needs to be sought. If there is insufficient available fruit, or indeed insufficient capacity to process the fruit then this dictates the volume to be created. The data model therefore needs to become a tool for balancing the capacity, available fruit, and market demands of the enterprise. How the capacity of the winery affects the plan depends on not just the expected volumes of wine but also how it is proposed to make it, and of course on the nett volume of wine to be produced. It may not be possible to know if there is sufficient capacity to produce the wine until all the plans have been entered, and even then there are bottle necks in the process because it may be fermentor capacity not storage capacity that reduces the volume of wine to be produced. This then introduces complications because of harvest dates and the need for fermentor capacity at certain times. Many medium sized wineries will use an individual fermentor up to 4 times during a vintage, if not more.

It is not the intention of this model to be able to make decisions regarding how this balance should be achieved. However, it should be able to make calculations to help the user arrive at an informed decision. The user needs to be able to specify priorities and be presented with the implications of those decisions. The available fruit will have expected harvest dates, but these will shift enormously. We only need to consider the 2011 vintage which saw the drought break, and very cool conditions that shifted some harvest dates by up to 6 weeks from the previous year's dates. As these dates shift, the model should reflect the changes and alert the user to conflicts that arise.

This functionality becomes crucial for enterprises like 7.5 (*Case Study 5 - Medium scale winery with multiple external vineyard sources*.).

8.2.2 Tasks and actions

The value in performing an ethnological study is that we get to witness and experience the tasks and actions performed by the winemakers as well as the motivations and intentions behind these tasks. This has profound implications for the nature of any system attempting to model them.

The most noticeable aspect of the observations made was that the collection of data all seemed to be done well after the actions were performed. It was common, particularly during vintage, to witness the winemakers or viticulturists sitting down at the end of the day tapping away at their keyboards and sorting their way through pieces of scrap paper and triplicate books. The data collection and entering process was not regarded as part of the process but rather an addendum to it. It was not that the recording of information was not important, just that it did not fit into the flow of the process. When pushed, they revealed that this was inevitably because there was usually nothing to be benefited from doing a task, going up to the office to enter it into the system, and then going off to do another task.

There are two implications here for the data model. Firstly, can the model be designed so that it can become an integral part of the processes, and secondly, should it?

The answer to this can be partly addressed using the concept of work-note sets, which recognise that individual tasks are often part of a series of tasks designed to achieve an overall goal. If the winemaker tells the model that this set of tasks needs to be executed, then the model will appear to be operating at the same level as the winemaker, and in a sense can become an assistant. It can set out the individual tasks and pre-empt the data collection process.

Another partial solution to this is the fact that most actions are planned. A detailed plan before vintage can be used to seed these pre-ordained tasks in such a way that the winemaker might use the model as a tool to drive and instigate the tasks.

It needs to be acknowledged that some winemakers would still be resistant to such a paradigm shift. On smaller scale enterprises, this might be regarded as over complicating a simple process so the model will still need to be able to operate in post-facto mode. Date-stamping tasks are also problematic because the system will not be able to assume that a task was initiated when it was entered, or completed when the final results are entered into the system.

If not then the model needs to take into account that data cannot be date-stamped as having occurred when it was entered.

In general terms it was observed that these tasks fall into five categories.

1. Planning. Plans provide the seed for the implementation of the production cycle. The winemakers can be as optimistic or pessimistic as they want, and it is the only stage at which they can say they have complete control of the production process. The task of planning a

vintage is statement of intention rather than a structure to be locked in and faithfully executed.

2. Fundamental. These tasks are the mainstay of the winemaking process, and consist of all the basic and core tasks for making wine. They distinguish themselves from other tasks by the fact that they do not necessarily have a causative event as a justification for their execution, other than the intent to make wine.
3. Pre-emptive or Pro-active. These tasks are part of the craft of winemaking. Experience, and often education and training, inspire winemakers to perform processes to avoid known hazards that might typically threaten fruit or wine, or increase the quality of the fruit or wine. They might also be classified as "ritualistic"
4. Reactive. When something goes wrong, winemakers usually have a number of actions they can take to ameliorate the problem.
5. Investigative. These tasks include laboratory work which often instigates reactive tasks

The specification of the nature of these tasks may seem very obvious and over-analytical, however it is precisely this detail that will help achieve many of the goals of the project. The category for any specific task is not set and depends on the circumstances under which they were ordered and actioned. The rationale for having such as structure is that these considerations lead to developing functionality for the system that also takes the desired end result into consideration, and helps relate these tasks, their intentions and their impacts on the rest of the system in a realistic and sequential manner. It will allow the system to behave in harmony with the user because it will focus the function of the model on the desired end goal of the user. The concept of grouped worknotes also provides a simple structure for linking motivation, justification and context to actions performed in the enterprise.

8.2.3 SUBJECTIVE DATA

The winemaking and grape growing industry is driven by opinion and perception. Unlike most industries that require product predictability and consistency year-by-year or have a distinct set of performance parameters, the wine industry is founded upon product differentiation based on subjective criteria and interpretation, not just for the end product but also for the raw source ingredients, and throughout the manufacturing process. For most winemakers, these subjective observations affect the course of the production process of their wines. They are often the clinching arguments for harvest dates, crushing/pressing regimes, intervention to ameliorate or prevent problems, fining parameters, and in defining the eventual destination for the wine. Here are some typical examples.

8.2.3.1 Fruit sampling for harvest date and yield estimates.

Selecting and defining a harvest date for fruit is often a compromise between several competing forces. Ideally it is based on taste, a balance between acid, sugar and varietal flavour, depending on the desired wine style. In reality, factors such as fermentor space, harvest equipment and/or labour availability, berry condition and weather patterns can also shift this date earlier or later. Nevertheless, choosing the ideal harvest date is often a matter of combining pH, T.A. and Baumé readings with subjective tasting of juice for sugar and acid balance and flavour ripeness.

8.2.3.2 Pressing

Just like harvest date prediction, cut off points for free run, light and hard pressings for both red and white wines can often be dictated by pragmatic issues such as tank space or time constraints. Nevertheless, when the winemaker has flexibility with these issues, the tannic or phenolic flavour profile and associated waxiness and astringency is often used to determine cut off points, as well as objective parameters like turbidity or pH. Devising an appropriate scale for this situation is a little more difficult because qualifying a taste that is departing from an ideal is more difficult (in general the free-run is regarded as better quality), particularly as such a scale assumes that the winemaker knows exactly how “bad” the hard pressings are going to get. Added to this is the fact that yields are reduced over time, some varieties are more susceptible to the effect of pressing, and incremental additions might have more or less effect on the eventual quality of the fraction once it is homogenised in a tank.

8.2.3.3 Routine tastings

Most winemakers regularly taste their wines to make observations on the progress of the wines through the manufacturing process and ensure that no problems are developing. Often, these tastings reveal characteristics that reflect the proposed wine style specified earlier during pre-vintage planning and might provide a catalyst for ameliorative actions to be performed in the form of a trial or some filtration of purging process, in order to achieve this desired style. So the tasting tends to be more specific than assigning an overall score or assessing the general appeal of the wine.

8.2.3.4 Amelioration, Fining, and Blending

Processes such as these are prime examples of subjective data. They are usually instigated by subjective requirements, require a subjective choice of technique, and above all result in subjective outcomes. In isolation, a winemaker will often adopt a single approach to such an issue, based on past experience. However, when more complex techniques are used, such as trialling multiple fining agents and combinations thereof, recording subjective outcomes can not only be of immediate benefit to selecting an optimal solution, but can provide important reference data for future situations. The ability to directly compare outcomes of situations from year-to-year, blend-to-blend, wine-to-wine has potential to allow winemakers to compare and improve techniques laterally and vertically, and at the very least provide verification and justification for existing techniques.

For example, when fining or making additions to a wine some winemakers might take a conservative or iterative approach and make additions in stages and build up to an optimal desired level, whereas others might go for an all-out attack and make the addition in one hit. Those who dabble in experimentation need to have a valid and reliable means of measuring results. Data that covers multiple vintages, varieties, or techniques would be invaluable to provide better analysis.

8.2.3.5 Bottling

Bottle shock, dissolved oxygen and filtration stripping are just some of the issues wine makers might have to contend with during the bottling process. It can be quite disturbing to spend months and months developing a wine, only to seemingly trip at the last hurdle and lose all the subtlety and character of a wine because of the relative invasiveness and rigour of the bottling process. Typically, one might expect wines to appear flabby (lacking appropriate acidity), or bland for several weeks after bottling, then suddenly re-emerge completely recovered, perhaps better, perhaps worse. Taking values for acid balance, body, levels of positive volatile characteristics, fruit flavour and intensity or other relevant parameters before, immediately after, and weeks after the bottling process can lead to directly gauging and quantifying the qualitative effect of the process, and lead to changes of both preparative processes before bottling as well as settings for the bottling process.

8.2.3.6 Marketing and sales plan development.

This subject represents a good example of the need for subjective concepts to be in such a form that they can be transferred to seed actions and processes in the winery and be compared qualitatively and quantitatively against outcomes. It is all well and good to dictate goals and aims, but the effort is somewhat futile if all they do is represent a reminder of the success failure rather than as an impetus to achieve the goal by providing indicators and comparative parameters during the manufacturing process. If we are able to translate desired wine-styles from prose into quality parameters quantified on an appropriate scale, then this information can be linked to actions and processes throughout the vinification and maturation processes. Of course, the cyclical nature of the industry means that the marketing and sales plans can also benefit from more detailed input into their development from quantified quality parameters from previous vintages.

The issue is that subjective data cannot be quantified in the same way that empirical, objective information is. If an attempt is made to quantify the data, then the scale and base line cannot be assumed consistent enough to be reliably compared when supplied by different judges or even on different dates. Discussions were held with several interviewees on this subject, and it initially appeared that if all opinions were to be taken into consideration, subjective data would have to be ignored because each example that was cited appeared to transgress this universality requirement for inclusion into the model. However, it remained obvious that this kind of information still needed to be recorded, because it was often the basis for a key decision in the winemaking process.

The key was to recognise the scope and use of the data. Peter Gago made a particularly enlightened statement; "*Experience has to be given some sort of credibility from an objective perspective*". If one is able to accept that this kind of data only has meaning within the context of the enterprise, and it should only be used as causal link in a chain of actions (or non-actions) then the information is still relevant. Experience is part of the winemaking philosophy and can be seen in the path a winemaking process takes. The difficulty is to let go of the scientific paradigm and embrace the artistic, interpretive nature of the industry. Subjective data can provide a valuable narrative for the process and therefore warrants a place in the model.

8.2.4 Budgets versus Costs

In general, the winemakers were not so much concerned with the strict accounting aspects of the winemaking process, but they were concerned with the costs and benefits associated with the variations in the available techniques. This enables them to make appropriate decisions as to which tools, materials, and techniques are justified on a financial basis. Given a budget each year, they have a degree of freedom in the management of this discretionary funding and how this money is spent. Obviously there are running costs that remain constant, some costs vary depending on the time of year, and many costs cannot be attributed to a single wine but rather need to be spread evenly across all wines produced.

The issue here is that there is more to such decisions than purely expenditure. The price of a bottle of wine cannot be determined purely based on the cost to make it and get it to a wholesaler or retailer, plus a profit margin. Additional factors such as quality, value, reputation, consistency, perception, competition, marketing strategy all come into focus, and this topic is well out of scope for this discussion. Nevertheless, many small to medium wineries do operate on a margin, and a scale and market position that makes questions like *"...am I justified in using this more expensive barrel*?" pertinent. And certainly, none of the enterprises interviewed in this project, even those at the upper end of the market, selling their wines for a ten-fold increase compared to its production cost, believed that this issue should be ignored. Most of the winemakers showed a distinct distaste for wastage, extravagance, and reckless spending.

As stated at the start of this document, it is not the intention for the data model to duplicate the same functionality of an accounting system. However some basic costing capturing functions would be useful to not only provide data for such accounting systems, but also perform basic calculations to aid in marginal spending decisions for additives and processes.

8.2.5 MANAGING PAPERWORK

Some aspects of managing the winemaking process are more concerned with managing paperwork and bureaucracy rather than the wine. Exporting wine is a good example of this in that the only significant difference in treating exported wines compared to domestically sold wines is the associated bureaucracy and regulatory documentation required to accompany the various entities involved in the process. The previous decision to exclude accounting functionality from the project could just as easily be applied to this issue. Exporting wine from Australia is a matter of gaining approval, in principle and in the specific instance. The enterprise needs to be registered as an exporter, the wine needs to be given export approval, the label needs to be approved to comply with regulations in the destination country, and the shipment needs approval. In essence these add up to a series of ticks with accompanying paperwork. Should this come into scope later in the process, then it will simply be a matter of creating extra properties or flags for the wine object and perhaps associated functionality to attach scanned copies of these documents. In the meantime, it is justifiably out of scope for this project because it is too far away from the immediate concern of managing the winemaking process.

8.2.6 THE VINTAGE PLAN

The term “Vintage Plan” meant different things to different people, and it was noticeable that the degree of detail they gave in their description was generally proportional to their expectations of any vintage planning functions.

For some, the need to create a vintage plan was moot because it would be the same as the previous year. For others, a good and detailed vintage plan was paramount because...

1. Significant variations and contingencies appear from year to year that make cloning previous year’s plans inappropriate, or
2. Changes in key personnel meant that plans and expectations for vintage needed to be reinforced formally and explicitly in a document, or
3. Like any organisationally responsible system, it makes sense financially, logistically, administratively, and professionally, to have a heads-up and a basis from which contingencies and exigencies can be prepared.

An appropriate starting point for designing a structure for a vintage plan lies in the high level data model cited earlier (Chapter 6). The aim will be to allow parcels of fruit to be matched to proposed ferments which in turn will be grouped into user-defined batches of wines. The term “user-defined” will gain particular significance here because putting functionality in the hands of the user is a double-edged sword for system designers. It does allow greater flexibility for the user when it comes to making the system suit their purposes, and can make designing the system simpler because it effectively removes responsibility for the business layer to provide detailed functionality, but it also means that this places more responsibility on the user to infer meaning to their definition, and this can lead to inadequate type and data checking leading to compromised data integrity in the model. The important information to glean here is that it will be necessary to always consider the allowed scope and nature of any user-defined construct, and build entities without too many mandatory properties or relationships. The concept of plans versus reality is discussed further in *8.3.3 Plans versus Actions – Data Integrity*

One of the most useful functions of a vintage plan is vessel and resource allocation, whereby the user can plan which vessels will be used for the various parcels of fruit on any particular day. To create this function implies that we know the expected harvest date and amount of fruit in a parcel and the number and type of fermentation vessels required, but this is not unreasonable. This situation represents an opportunity to create an entity to define the vinification parameters for the parcel which might also include more specific details about other materials and resources that need to be applied, which in turn will also help generate vintage requirements, just like the documents generated by Case Study 1 in section *7.1.18.1 Vintage Planning*.

Extrapolating even further, we might be able to match wine batches to proposed labels. However, we also need to be weary of expecting the user to be willing or able to predict much about what is expected during vintage.

Perhaps the greatest lesson learnt through observing commercial vintage, particularly at the primary case study site, is that anecdotal tales of the need for multiple contingencies, and that something will always go wrong are real. It is no exaggeration that nothing ever goes exactly to plan. Over the 4 or so years that the operations were observed, throughout vintage periods in particular, the enterprise a broad range of situations that caused massive modifications to much of the logistics.

8.2.7 MATERIAL TRACKING

There can be hundreds of different materials used in the fermentation, finishing, fining, clarification, and maturation of wines. Often, these materials are required at short notice to ameliorate problems that occur during these various phases. Some need to be refrigerated. Others have defined and quite short use-by dates. Many are very, very expensive.

These materials can generally fit into one of eight functional categories.

- Fining Agents
- Sulphur Dioxide
- Yeast
- Nutrients
- Acid
- MLF Bacteria
- Chips
- Other

There are several important factors to consider when managing these materials. Dosage rates are critical, and mistakes in this area are potentially catastrophic. The system will need to know the appropriate units of addition of these materials in order to calculate and present additive rates in context. Batch numbers are also important for many winemakers, so the option to record and trace such data is also required. An option might also be to include inventory control, although this was seen as problematic in some circumstances because materials like cleaning agents were not used directly in the winemaking process and could not necessarily be attributed to a specific wine and it was seen as overly officious to track volumes used on a case by case basis.

It is worth noting here that utilities such as water, gas, electricity will not need to be tracked. This is for two reasons, 1) there are already adequate accounting packages available to perform such functions, and 2) the costs associated with these are not attributable to specific wines but are part of the general running costs of the enterprise.

8.3 Systematic Issues

8.3.1 The Role of the Interface for Testing and Judging the Data Model

In order to explain some of the concepts of the proposed model to the case study participants, it sometimes became necessary to outline some possible examples of the user interface. This exercise also revealed an insight into the mindset of the interviewees and formed the basis for the design of the interface. However, it was clear that in the end the user interface would need to be a significant compromise between the disparate ideals put forward. As a result, although the user interface has an important role in testing the success or otherwise of the data model, it will not necessarily be a definitive interface for the model, and should not be judged as such. It will act as a means of testing and illustrating the available functions, features, and possibilities for fulfilling the aims of the data model.

At the same time, the interface will help define many of the business layer rules because this will be where many of these rules need to be enforced.

8.3.2 Concrete and Virtual Objects

If a data model is going to be applicable to all these case studies, then the key word must be flexibility. It must be capable of dealing with all the combinations of facilities within these enterprises; from single vineyards or wineries, to wineries with multiple vineyards. It must be capable of dealing with all the different levels of relationships between the user and these facilities; from direct management and intimate control, to passive contractual relationships. It must be capable of dealing with all the different management profiles, from experimental, leading edge, modern technological philosophies to simple, traditional approaches.

It is important to point out that these concepts need not be complex. The whole point of using object oriented design principles is to use carefully designed virtual constructs that can provide broad functionality without necessarily having a real-world presence. One example might be for a wine object that has to represent several concepts of “a wine”. During the planning phase of a vintage, wines are conceptual entities that provide convenient planning strategies. One instance of a wine object may represent a proposed fermentation batch, a collection of tanks or barrels of wine, or batch of wine ready for bottling under a specific label. Later, during vintage, an instance of a wine object might represent an actual ferment, or a wine in tanks or barrel. In general there is a shift of focus from a wine being a product of sourced fruit to a wine being destined for a label. The point is that different instances of this object can represent different concepts, but by using the same object with broad functionality, these instances are easily changed from one concept to another, depending on the context and properties assigned to that instance. (Figure 15)

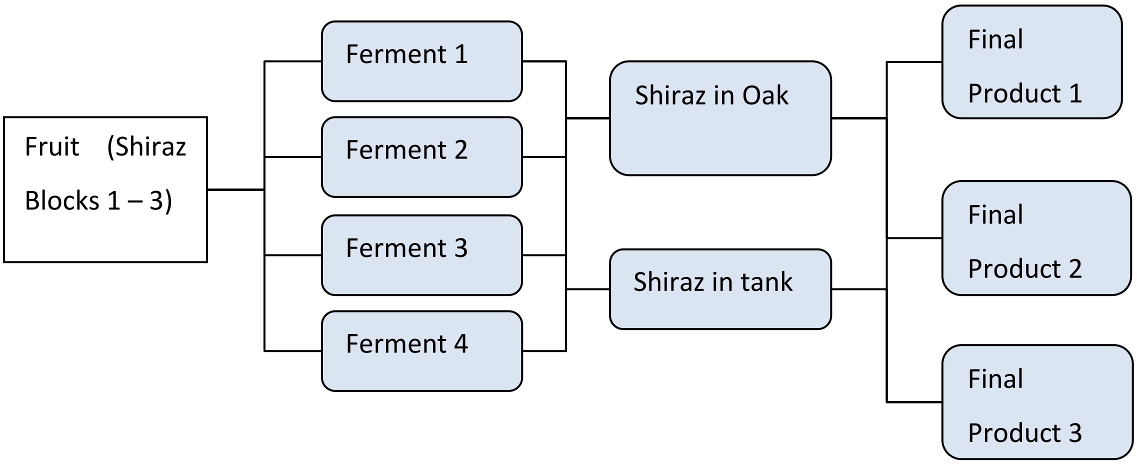


Figure 15 - Different instances of a wine object in planning phase

In addition we need to step back and look at the intent and motivation of the user of the model. It is not enough to simply mimic the actions of the grower or winemaker. If the model can somehow have objects that represent elements of the users own strategy for performing their roles then we are that much closer to defining a useful model. The case studies abound in such strategies.

The winemakers understand from the beginning what types of wines they intend to produce and where these wine styles fit within the Australian market. They also have a very good understanding of the implications that certain techniques have on the end product. This aspect of winemaking is innate and represents the cross-over point from the objective to the subjective, from fact to opinion, from viticulture to grape growing, and from oenology to winemaking. It also represents the boundaries of this project because it is ludicrous to expect any data system to be capable of even participating in such decisions. However this does not mean that the system cannot understand the objectives and provide tools to present this kind of information.

It would be easy to simply provide this functionality within the model and supporting interface, but the important facts to take from this information is to understand the winemakers' intentions and way of thinking. Clealry they understand the concept that the plan can help to derive requirements for the up-coming vintage but the methodology shown in these examples is more to do with the tools they have on hand rather than a direct reflection of their mindset. The winemakers were constantly balancing winemaking, marketing, and budgetary decisions. The model can take advantage of the fact that all the data for each of these factors will be available for presentation to the user at the appropriate time and within context.

The best way to utilise this information is to use it as both a means of inspiring concepts for the model and as a means of testing the model at the end.

8.3.3 Plans versus Actions – Data Integrity

The concept of a "Plan" implies an intention to perform a series of actions. However, when an action has been performed, the plan, or that particular part of the plan is a "fait accomplis". This raises several issues for consideration...

If the results of the action do not precisely match details prescribed in the plan, should the plan be modified to match the details of the completed action or should the plan remain a record of what was intended?

Similarly, once an action has occurred, should a user be able to modify the details later, even if the details of subsequent completed actions will be affected?

These issues need to be resolved early in the process because they have fundamental impacts on the design of the model. They also raise issues concerning the way tasks are performed in a winery, given the sometimes ad-hoc nature of performing tasks during the intense vintage period.

To explain this better, let us walk through a hypothetical scenario. A winemaker has planned for 9 tonnes of chardonnay fruit to be harvested on the 10th March. The intention is to crush and press the fruit to produce 6000 litres of juice which will be fermented in a 6,500 litre tank; the excess space is required for increased volume during the fermentation process, space for foaming or for pump-overs. It is expected that the fermentation process will take 10 days, after which the wine will then be transferred to barrel for maturing and malo-lactic fermentation. This is the plan, but in reality several things could change.

- The fruit arrives late because of a harvester breakdown.
- 12 tonnes of fruit arrives, requiring a larger fermentor.
- A stuck-ferment results in the ferment taking 15 days.

The list is endless, but these all have repercussions for the plan.

Firstly, whether the plan is to be modified to match what actually happened or whether the plan should remain a pure statement of the original intentions.

Secondly, the need to separate vintage plans from blending plans became particularly relevant during the 2009 vintage in Victoria. Unprecedented bushfires lead to many vineyards in the Yarra Valley and other regions to produce smoke tainted fruit. This caused wholesale changes to all the plans constructed prior to vintage, and caused most of the wines for some wineries to be rejected for certain labels and ended up being sold as bulk wine, after extended R.O. filtering.

So, there is a need to keep a copy of the vintage plan and the blending plan as such for the sake of purity in that the original intentions can be used to compare to the final outcome at the end of the process, but also to keep a running plan for both so that as vintage progresses, the system can inform the user of how well they are achieving the production plan, and keep a running estimate of the wine production that takes ongoing contingencies into consideration as reality is met head on.

From a modelling and data relationship point of view, this should not be an issue so long as the pure data of plan and action is preserved and maintained in the model. Questions such as those raised above only become an issue for presentation to the user, not for the contents or structure of the model and the database.

A suitable design philosophy will be adopted to allow soft relationships between entities so that, for example, a plan item can be linked to an actual event or entity representing a real item, but that the relationship is an inference that is used to seed input parameters for the real entity but does not fix the parameter. An example might be an expected harvest date for a parcel of fruit. When the fruit actually arrives, the plan may state that it was expected on specific date or that a specific amount was expected, but the actual date and tonnage may differ and will obviously need to be input and stored separately. Both sets of parameters are linked to the same object (the planned fruit parcel) but not in the same entity. What is required is a simple link between a plan entity and a real entity and that this link is to be used with caution and that no other inferences should be made.

8.3.4 FACILITIES

The diversity of function and nature of the enterprises in the case studies show that it is not sufficient to represent an enterprise as a single entity from a modelling point of view. However, even the introductory description of each of the case studies reveals that an enterprise can be seen to be made up of a set of discrete entities.

After a brief debate, the term “Facility” was chosen as a generic entity type although other suggestions like “unit” and “component” were also considered.

There was a degree of freedom available here to break an enterprise down and design various facility types that might be useful to represent any enterprise within the scope of the project. Given the previous discussion point concerning disparate winemaking philosophies, an opportunity is available to tailor facility types to include or exclude functionality deemed appropriate for the enterprise as a whole.

The case studies illustrate the two extremes of this situation. In case 1, an enterprise made up of a winery, an external winery, two main vineyards, several external vineyards, a laboratory and an external laboratory, and a couple of storage facilities. In case study 4 an enterprise consisting of both an external vineyard and winery, and a single managed storage facility.

The distinction between facilities and external facilities is of primary importance here, because there is a natural domain of control that applies to any facility within the enterprise. (This point was made in case study 4). For example, management of the winery, the laboratory, one of the storage facilities and one of the vineyards in case study 1 is performed under the one roof because of their proximity. It is relatively easy to coordinate functions between these facilities without having to rely on networked communications. However, external laboratories, vineyards and wineries are not directly managed by the users of the system. Nor are they necessarily concerned with the internal machinations of the facility; just the relevant information that goes in and out.

The terms "Managed" and "Unmanaged" were adopted at this point to distinguish between core facilities of the enterprise and peripheral or passive facilities that are just as essential to the operation of the facility but not under direct control.

By making this distinction we can restrict the functions and properties of unmanaged facilities in the model.

8.3.5 Volumes

Throughout the winemaking process, there is one key parameter that has implications for all process that occur to a wine. The volume of a wine, particularly during fermentation on skins, is often subject to error with respect to both accuracy and precision. There are also losses associated with many processes such as pressing, racking, transferring from vessel to vessel, and filtering. Initial measurements cannot always be verified or corrected because losses interfere with error margins and vice-versa. The question arises, how retrospective should volume measurements be made. If a wine is found to be of significantly larger or smaller volume than first thought, how far back should that measurement be applied to reporting results, rates of additions, and other parameters using the volume as part of the measurement?

Example

- 10.240 tonnes ± 40 kgs , fruit for fermentation on skins

- Assume 70% yield w/v = 7,168 litres ± 28 L
- Sulphur Dioxide addition @ 30ppm total = 215.04 g (this would typically be rounded to the nearest gram amount, so 215g)
- Tartaric Acid addition of 0.25g/L = 1792g (rounded to 1800g, say)
- After pressing, volume found to be 7,320 litres ± 50 L as measured by an inline measuring device.
- Racking yielded 6950 L ± 50 L and left approximately 350 L of lees.
- Transferred to 30 barriques with 100 litres left over into a 200 litre variable capacity stainless steel tank for topping. (total volume now 6850 litres)
- Topping over next 6 months lead to complete use of the 100 litres of extra wine.

This simple example shows the impact of different volume measuring methods. The volume of wine based on a yield is obviously the most inaccurate because it assumes the result from a process that has not yet been performed. However, we see that two additions have been made based on this figure. Subsequent measurements established that the actual volume of wine was greater than that assumed, so how should this affect subsequent reporting of addition rates prior to the pressing process?

In actual fact, it turns out that this is not an uncommon or even a critical situation. The additions were made at a time when they were judged to be correct and with the best available information. Secondly, the effects of the additions are not judged on an objective basis. The winemakers know that the effect of the SO_2 addition may not be a 15ppm increase in free SO_2 even if the volume was accurately known, because the wine always responds differently, and depends on very complex chemistry that is not thoroughly understood. The acid addition is made to taste and is also subjective, although there is an objective aspect to the addition to drop the pH, but this is also unpredictable.

One winemaker cited the fact that some existing data systems seem to have an obsession with accounting for losses during process such as filtering or transferring wines, or losses that become apparent during topping exercises. In practice, error margins for volume measurements are just as large as losses accounted for in such circumstances, and that the calculations seemed more for the benefit of accountants than for winemakers. It was not uncommon for the volume of a wine to increase after a process, solely because a different and more appropriate technique was used. For example, when a wine is transferred from stainless steel to barrique (all barriques are assumed to be 225 litres) then the number of barriques is used to make the volume estimate, not a dip taken from the source vessel.

Winemakers tend to be very pragmatic about this issue. It is not that they do not care; it is just that on a small to medium scale enterprise, the measuring techniques are just not accurate enough, and the consequences are not significant enough to justify elaborate or costly techniques for volume measurement.

The primary case study site utilised several means of volume measurement...

- "Educated" guesses of expected yields based on tonnages, fruit condition, and previous experience.
- An in-line flow meter.
- Equipment meters on the press and cross-flow filter.
- Dip measurements in tank.
- Assumed volumes of vessels combined with ullage estimates.

8.3.6 Define "A Wine"

Central to the core of any data model is the definition of the main objects that are being modelled. If these entities are not appropriately designed then the system is hobbled from the beginning. We need to know what is expected of the model when it comes to managing and manipulating the core entities.

As a principle, it was decided very early in the project that the wine object ought to be the primary entity in the model. This was not only an intuitive decision but also a significant conclusion from previous work by this author in the design of a winery management system which adopted a different stance and was found to be inadequate. The problem was that not enough functionality was required or defined for the application to justify putting a wine object as the focus of the structure.

The case studies and interviews conducted illustrate that the definition of a wine in the industry is contextual. Within the winery, winemakers tend to refer to a wine in terms of an individual container or vessel. However, it can also be a collection of vessels if that is the basis of how the wine is to be treated in terms of any action that are to be applied within the winery. The most common form of a wine in multiple vessels is when the wine is in barrels. It is too cumbersome to manage wine in barrels as separate entities when they are treated as a whole for most if not all actions and measurements taken within the winery. In this case, the barrels are said to be in a "vessel set" which will be adopted as a convention, although there was no common terminology amongst the case study participants.

However, from the perspective of the data model, a wine needs to be defined with functionality in mind and with considerations such as those cited in *8.3.2 Concrete and Virtual Objects*. In particular, we need to make sure that we can recreate the history of a wine and its constituent source fruit and/or wines as well as track all the events that cause these parameters.

9. User requirements – from the top, down

The following chapter is simply a statement of expectations of the system by the proposed users. It focuses more on the way the model appears to the user rather than stipulating any required structural features of the model. The first section is just a wish-list of functions and features generated by the winemakers interviewed and resulting from observations during the ethnological study of their workplaces. The second section is concerned with governmental legislation that can be accommodated and facilitated by the system to help the enterprises fulfil these requirements.

9.1 The Wish List

From the outset it was obvious that the system needed to be innovative when it came to interacting with the user. It was not enough to just provide screens that allowed the user to enter information; the information needed to be in context, not just oenologically or viticulturally, but also emotionally and in sympathy with what the user was trying to achieve. If we are able to identify the common circumstances that the user is in when they are using the system then it seems only appropriate that the system should provide contextually appropriate prompting and functions. As it turns out, there was considerable consensus about the nature of these modes.

9.1.1 The Vineyard

In the vineyard, operations tend to reactive, pro-active, or maintenance related. These activities tend to be applied to sections of the vineyard rather than across the whole property. In addition, these activities are seasonal in nature so there tends to be a natural mode setting for vineyard managers depending on the time of the year.

Reactive operations include activities like fence, post and wire repairs, and spraying in response to disease outbreak. Pro-active operations also include some spraying activities to prevent disease, as well as irrigation, fertilisation, and fertigation, crop thinning, and leaf plucking. Maintenance activities generally dominate work in the vineyard and tend to be regularly scheduled events like pruning, and mowing.

The concept of a new vintage beginning the day after the last fruit has been harvested is not appropriate for an all-encompassing enterprise-wide system. The idea of having an official vintage nominated at any given time, so that any work carried out is regarded as part of that vintage, is not a safe assumption. Legally, any vintage label claim assigned to a wine is based on the year the source fruit was harvested. It is conceivable that one day some fruit may be harvested in late December and this would necessarily have to be regarded as fruit from what would normally be regarded as the previous vintage. This would wreak havoc with record keeping if the vineyard regarded the fruit as coming from the subsequent vintage.

On the other hand, it is appropriate that any actions that take place during the dormancy and growth periods have a significant impact on the status of the fruit for the new vintage, so there needs to be some acknowledgement that the continuity exists and that a vintage cannot be regarded simply as a calendar year. The implication is that we cannot give the nominated vintage any significance other than a nominal one. The system should allow an official vintage to be specified, but this is to be controlled by the users, and they will take responsibility for the significance and consequences of controlling that flag.

In general, the wish-list for vineyard operations consists of the following...

A Diary: Operations tend to be well regimented and scheduled so there is a need to be able to operate a diary to book dates for the various actions.

Visual Overview: Vineyard managers are very familiar with the physical intricacies of their vineyards. It is essential to include a map of the vineyard to allow quick navigation to identify relevant features and specify actions. It should also allow vine status information such as Eichorn-Lorenz numbers to be entered to facilitate monitoring the seasonal aspects of the operations.

Vineyard Breakdown: Vineyards are traditionally broken down into blocks, but it is common to see further subdivision into areas of interest, vigour,

Spray Diary: This is mandatory and needs to closely mimic functions associated with a manual system. This implies a suitable materials register will also be required to monitor batch and inventory information for the various products used.

Equipment Maintenance: We also need to consider the equipment used in the vineyard which require a regular maintenance regime. The ability to schedule manufacturer recommended maintenance would allow better equipment reliability and better use of down-time.

General Notes for blocks: Viticulturists and vineyard managers need to be aware of not just each expectation for the vineyard, but the expectations of their customers; the winemakers. These tend to be in the form of philosophical regimes for the treatment and management of the vines and desired characteristics of the fruit rather than objective performance parameters. Users need to have a common place to reference and update these notes.

Harvest Parameters and Yield Estimates: Data in this area is the common language for both wineries and vineyards. The ability to share this information, regardless of who generated it, is the key to getting the two entities to function and interact seamlessly.

Fruit Parcel Definitions: Winery logistics tend to complicate the winemaking process. As discussed earlier, fruit ripeness is not the only determining factor for deciding when to harvest fruit. There needs to be confluence of available staff, equipment, winery capacity, and fruit quality according to wine-style. This is necessarily driven by the winery but naturally has significant impact on operations in the vineyard, so an appropriate structure for defining how much of which fruit and how it should be processed is essential to allowing the vineyard operations to plan and comply with the requirements of the end user of their produce.

Historical Data: Historical information has an important role in helping understand the behaviour of the vines and predict how they will behave in the future. Ideally this information should be easily accessible and presented intuitively in context with other functions.

Data Mining: Primarily inspired by the input from the vineyard manager in case study 1, it seems almost mandatory that the vineyard component of the model should allow and provide data mining functionality to investigate and provide evidence for the correlation of inputs to outcomes, whether those inputs were natural or human, and whether the outcomes were favourable or not. The important factor here is that although the system and model should not be required to interpret or have an insight into any relationships within the data, it should provide a tool to make the investigation process easy for the user.

It has to be noted that the wish-list for vineyard operations is not as extensive as that for wineries. It is possible to read any number of implications and inferences into this but in general, vineyard managers saw their job as well defined and fundamentally driven by known parameters and techniques. They seemed to be more accepting of the passivity of their role; being subject to the forces of nature far more capable of determining the eventual outcome of their endeavours. On the other hand, they also acknowledged the importance of their functionality and data being part of a greater process which stood to benefit from integration into a single system.

9.1.2 The Winery

It was found that winemakers tend to think in goal-oriented patterns rather than focussing on the process. That is to say, their actions tend to be instigated and motivated by a desire for an end result, and the process used is merely a manifestation of that requirement. (In contrast, an accountant doing taxes is more focussed on the process, and the end result has no bearing on the process). For this reason, the interface needs to have the same focus on providing logistical tools to help establish and facilitate end results rather than a pure function based system where processes are listed and executed in a predetermined sequence without any consideration for their context.

9.1.2.1 Modes of Operation

Over several months, it became apparent that the various mindsets that winemakers adopted, fitted into four different categories, and curiously, these categories could be described and perhaps even explained by the level of urgency that the situation dictated.

1. **Remedial/ Investigative**: This relates to situations where known problems are identified and a solution requires tracking the events that lead to the current state of a wine or wines. The winemaker will need to see the sequence of events that occurred, and be able to do some data mining to all forms of data to establish the cause and effect of the issue.
2. **Routine**: This is perhaps the most typical situation when the winemaker needs to be able to see what is going on in the winery and what jobs are yet to be completed. Many tasks are routine and so there is a need to be able to build standing orders that will automatically generate jobs based on a regular periodicity. Some jobs need to be completed before others can occur so it is important to see these jobs in the context of the wine or vessels they apply to as well as collectively in sequence. A diary function was requested to be able to include other non-winery job related events that also need to be considered when scheduling work.
3. **Pro-active**: It is seen as essential that winemakers are able to be proactive and avoid potential problems in their wines. The use of standard operating procedures (SOP) is seen as a first line of defence against such problems but it is readily acknowledged that this is not impenetrable. Although winemakers are all very familiar with the concepts of HACCP, adherence to its principles in small and medium wineries is often informal at best. Within the context of winemaking, hazard analysis is based on the effect of known parameters on the vulnerability of a wine. It seems logical that the functions of monitoring wines and HACCP share a common basis of evaluating a wine's exposure to these threats and evaluating the danger. For this reason, scum runs (*7.1.5 Operations*) and wine scoring/monitoring regimes (*7.1.18.3 Wine Monitoring*) from case study 1 were seen as important to replicate if not functionally enhance in the proposed model.
4. **Planning**: The issues to do with planning a vintage have been covered earlier in *6. A high level data model as a starting point* and *8.2.6 The Vintage Plan*. These functions are not only on the wish list but are also regarded as essential from a data integrity viewpoint as well as an operational relevance. The system needs to allow a simple plan simple to enter and

manipulate in the system. The ability to enter complex plans should not compromise this first fundamental. In essence, the system should present a list of the available fruit to be processed and then match them up with proposed fermentation definitions. These, in turn, should define fermentation vessel types and other parameters so that vessel allocations can be scheduled, and conflicts or bottlenecks can be identified.

9.1.2.2 Roles and Permissions

Winemakers tend to be quite reluctant to allow free and complete access to their record keeping systems to all but the most trusted members of staff. This is of course a natural attitude to have given the potential to damage vital information. On the other hand, it was acknowledged that there would be some value in allowing certain functions to be available in a restricted regime to certain people, independent of their roles in the enterprise. So rather than have permissions and functionality linked to the role of the user it was decided to have a simple hierarchy of permissions, assignable on a user by user basis.

- System Administrator
- Full Access User
- Restricted User

Each level has less functionality available to the user than the previous level. This idea means that as each function is built, it is possible to decide which aspects of that function will be available to each level of user. It must be remembered that the scope of this system is such that, even in the largest scale of enterprise, there will only ever be a few users with access to it. Having a complex permissions system is difficult to maintain and therefore unnecessary. It was noted that small enterprises will necessarily have a potentially inexperienced user as a system administrator, so any functions that are restricted to this level will also come with warnings when they are used.

9.1.2.3 Areas within the winery

Larger wineries tend to be divided into areas, often differentiated by the physical conditions as well as location. For example, oak vessels tend to prefer a range of temperature and humidity, and constancy within these ranges. Some wineries have so-called “Barrel Halls” which are heavily insulated and environmentally controlled. Cool rooms are also common place at larger wineries. Because some vessels are mobile, it is necessary to be able to move vessels from area to area, and these movements need to be recorded. In fact, some vessels are used to move wine from winery to winery, so this will also need to be taken into consideration. Obviously, other vessels are fixed in place, so it is will be necessary to differentiate between these vessel types and treat them differently

9.1.2.4 Barrel Sets

This concept was introduced in *section 7.1.12 Use of Barrels*. A Barrel Set is a concept more familiar to winemakers at larger wineries. This is because these vessels tend to have smaller capacity, but larger wineries generally deal with greater volumes of wine in a single batch. The concept may not be necessary for small winemaking enterprises, but it is essential for the larger scale enterprises. In order to explain the concept, it was initially likened to the computer metaphor of files in a folder. If one could drag and drop these entities from one collection to another then the process of managing these collections becomes more intuitive.

However, there are two perceived issues which will need to be resolved. Migrating vessels from one set to another is not the same as transferring the wine that was in those vessels, and the possible confusion about this needs to be made very clear. Secondly, individual vessels in a set cannot be treated separately on occasions, when it suits. A barrel set will only have one wine. Parameters for that wine will be assumed across all vessels in the set. This, too, is a discipline that is expected to trap some users unfamiliar or inexperienced with the consequences of such a concept.

9.1.2.5 User Definable Wine Groups and Wine Flags

Part of the discussion concerning the subjectivity of data, and the fact that winemakers tend to think in very individual and idiosyncratic patterns, lead to a resolution to allow user defined wine groups and common flags to categorise wines across various criteria. This means that the user ought to be able to define possible values for these wine groups and apply any meaning they wish. It is believed that the common flags will make grouping with respect to some more common groupings moot, but the functionality will exist. Suggested common flags include...

- Undergoing Malo-lactic Fermentation
- On Lees
- On Bentonite
- Filterability OK
- Fined
- Cold Stable
- Heat Stable
- Ready for Bottling
- Blending Wine
- Complete Wine

These flags will be set by the user and will not be linked to any actions to have them automatically turned on or off. This is a deliberate decision because the winemakers wanted these flags to remain in the control of the winemaker, not used as indicators based on actual events.

Predefined groups will include...

- General Wine Group
- Vintage Claim
- Variety Claim
- Regional Claim

9.1.2.6 Naming conventions for wines

This topic requires some clarification only because it was a source of contention for some of the participants. In some circumstances, the name of the wine was given critical significance in denoting the provenance of the wine, the nature of its source, or the intended target for its use. In these circumstances there was a requirement that each wine be given a unique name and that there needed to be a means of incrementing the name of the wines derived from a wine in some way to indicate its place in the sequence. The term wine name was often referred to as the “wine code”. In other enterprises it was used as a means of grouping wines together by giving them the same name. Some winemakers commonly used both conventions. Clearly there is no way that these two conventions can be resolved because they impose a function on the name, and these functions are so disparate.

However, the fact that the name is functional means it can be treated as a subjective parameter, just like any other. The data model needs to be able to...

- Store the name of the wine as a reading
- Generate a default name for any new wines resulting from the initial processing of fruit
- Allow users to change the name of a wine to anything (within reason).
- Offer an incremented name when a new wine is created from existing wine(s)

The important thing to note is that system cannot assume any meaning from the name. A simple default template for wine names can be standard without alienating any users, and incrementing names is also a fairly standard logical process of identifying strings or numeric sequences at the end of the name and increasing it (alphabetically or numerically).

9.1.2.7 Blending calculators

If there is any situation in a winery where the preferred tool of choice is a computer, then it is during a blending trial. This is because many blending trials are carried out in order to meet an objective and calculable end result, or that there are objective issues involved in justifying or ratifying a blend of two or more wines. In the absence of specially built tools, a spreadsheet is the most common means.

The need to trial and create blends of wines can be instigated by many different rationales. It can be for highly subjective reasons such as identifying and building reserve wines for high end labels, or might be for more pragmatic reasons like consolidating wines for extra tank space. However there are a couple of techniques that are common to most circumstances and the model will need to provide these functions in the appropriate context. Overriding factors include calculating LIP by vintage, region, and variety, as well as tweaking proportions by percentage and by volume. The user will need to be able to select wines from the winery and include them in the blend calculations. Another useful calculation is to work out the maximum amount of a particular blend that can be made.

Details for this function are mainly the concern of the user interface rather than the data model, as long as the model can provide the appropriate data and entities to allow the calculations. It is expected that more detailed specifications will be generated during the development and testing phase of the project.

9.1.2.8 A Wine Tracker

In addition to the data mining concept, there is also an opportunity to present the winemaking process in a visual manner by either using a family tree or comic book structure to display the wine as objects linked by actions and events. The advantage of a visual representation is that it allows a more intuitive understanding of the cause and effect of actions and events in the history and evolution of a wine. We will be able to track the wine, from constituent source fruit parcels to current state and isolate events and parameters at each stage of the chain.

9.1.3 Laboratories

Within the scope of this project, a laboratory can be anything from a dedicated room full of expensive analytical equipment, glassware, spittoons, and air-conditioning, to the corner of a desk in the winery with a couple of beakers and a hydrometer. Therefore, the separation of this as a unique facility within the model is dependent on the scale of the enterprise. It may not be necessary to have this available for smaller enterprises because many rely on external laboratories for all but the most basic analyses. Even at well established small-medium wineries, a dedicated laboratory was regarded as somewhat of a luxury. Nevertheless, in enterprises where they do exist, there is a special relationship between the winemakers and the laboratory staff and this can be aided and enhanced by appropriate functionality in the model. The idea is to allow the winemakers to generate jobs that include laboratory analyses worknotes, and these worknotes should automatically appear on the laboratory operator's screen.

- The screen should lay out the required analyses in a logical sequence and allow for the different analyses types to be displayed with appropriate calculators and parameter fields. Where possible, the calculations for the results should be automatic.
- There should also be an allowance to enter the results directly, without having to fill in the parameter fields in the event that only the result is known.
- Previous results (if known) should appear in context to allow direct comparison with new data.
- A report should be available to help summarise the sample collection process.

The following test categories and parameters should act as a reasonable starting point for the possible analyses to be catered for.

- Fruit/Juice
    - Harvest Parameters (pH, TA, Baumé)
    - Yield Estimate
- Enzymatic
    - Ammonia
    - Malic Acid
    - Primary Amino Acid
    - Residual Sugar
    - Smoke Taint
    - Volatile Acidity
- Wine Parameters
    - Alcohol
    - Residual Sugar
    - CO_2
    - Dissolved Oxygen
    - Filterability
    - Laccase

    - Turbidity
    - pH and T.A.
    - SO_2
    - Temperature and Baumé
- Trials
    - Addition/Fining
    - Blending
    - Tasting
- Stability
    - Cold
    - Heat
    - Pinking

9.1.4 Storage Facilities

The required functionality for storage facilities was kept to a minimum because any greater scope beyond that of the winemaking process would complicate the model beyond reasonable boundaries. It is understood that there is a plethora of stock controlling software available that is far better suited to managing this issue. All that is really required is simple functionality to allow the movement of stock to and from various facilities, and some minor functionality to account for sales to maintain a balance. The added prospect of cellar-door sales functionality was quickly excluded as being out of scope for the project, but this is where such functionality would be appropriate in future development.

9.2 Legislation to consider

Although there is no obligation for the proposed system to fulfil any legal functions, it ought to be able to provide information to simplify the generation of appropriate data for the enterprise to aid any reporting obligations.

9.2.1 Levies

For example, there is a wine grapes levy administered by the Federal Government’s Department of Agriculture, Fisheries and Forestry which is payable by the owner of the fruit when it enters the winery for processing, based on a calculation of a base and stepped rate for the net tonnage of grapes. In instances such as these, the system needs to be capable of generating such information, either explicitly as a designed report or through a data mining function.

9.2.2 AWBC Label Integrity Program

The Australian Wine & Brandy Corporation (AWBC) Label Integrity Program is a regime under which all winemakers operate in Australia. It stipulates rules about record keeping, label claims and is the main reason why some winemakers keep any records at all. As mentioned earlier, for this project, the LIP program should be regarded more as an impetus and initiator of the need to keep well structured records rather than a mandatory and bureaucratic system of oppression.

9.2.2.1 Label Claims

The LIP program is all about ensuring that information on any label of Australian wine can be justified through record keeping as fitting within certain criteria. The rules are subject to change but generally apply to claims concerning the vintage of the wine, the Geographical Indication (GI) of the source fruit, and the variety of that fruit. Claims are not mandatory, but any claims made must fit into certain criteria. The system will not interpret these criteria. It will be up to the winemakers to decide label claims and their appearance or otherwise on any labels. The system just needs to be able to display the make-up of the wine with respect to variety, vintage, and region and keep track of how it changes with successive processing within the enterprise.

9.2.2.2 Record Keeping

The following is an excerpt from the WineAustralia website regarding the record keeping requirements of the Act.

> **Records to be kept by wine manufacturers**
> **Records To Be Kept By The Record Keeper (AWBC Act, Section 39F)**
>
> - The identity of the record keeper;
> - The kind of wine goods to which the record relates
> - The identity of the person from whom the wine goods are received;
> - The quantity of the wine goods received;
> - The vintage, variety and geographical indication of the wine goods;
> - Details of steps taken by the record keeper that changed or affected any of the following:
>   - The vintage, variety or geographical indication of the wine goods;
>   - The tank or other storage unit in which the wine goods were stored;
>   - The volume of the wine goods stored in the tank or storage unit;
> - The date the record keeper supplies the wine goods;
> - The identity of the person to whom the wine goods are supplied;
> - The quantity of the wine goods that are supplied;
> - The vintage, variety and geographical indication of the wine goods;
> - Any other details in relation to the wine goods that are prescribed by the regulations.
>
> Records must be made within 3 days and retained for 7 years.

http://www.wineaustralia.com/australia/Regulation/LabelCompliance/LabelIntegrityProgram/Requirements/tabid/259/Default.aspx

These requirements are regarded as elementary to the data model. The main means of displaying this information will be via the proposed “Wine Tracker” function (*9.1.2.8 A Wine Tracker*.)

10. The Graphical User Interface

This is a high level description of the proposed user interface. It should be made clear that the nature of this user specification document is to provide descriptions of the functionality, not how it should be achieved technically. It is provided purely as a seed for the model design, not as a detailed design of the interface. It is expected that the final user interface will develop from the model as its functionality is developed. This is different to normal application development processes because the goal is not to provide an application, but to provide a data model. The user interface will be there just to show how the model can be applied, and exercise its capabilities.

The most important consideration we need to deal with when attempting to design the user interface is the fact that many people interviewed throughout the course of this project made the assumption that the system would be assuming control over the winemaking process. Nothing could be further from the truth. The interface needs to be designed so that it reflects the actual thought processes of the user when they are considering the way they produce grapes or manufacture wine. The interface will hide the complexities of the data model. The exchange of data between the user and the interface is very different to the technicalities of the interface's interaction with the model and the data layer.

Secondly, the interface should allow and encourage the user to explore the system. To do this it should use images and icons that make the user want to investigate. By giving the interface a passive appearance rather than strict structures and menus, we can avoid the effect of the system making the user change their thought patterns to accommodate the system's paradigm.

It was decided that a common theme for the interface should be the inclusion of images and maps familiar to the user. This offers familiarity for the user and can help put the user in context with the system. There is no avoiding push buttons and other common controls like grids and icons. Although they may tend to push the user into computer-like thought patterns, features such as contextual help, tooltips and reassuring messages can often counteract the sense that the user is operating a computer rather than truly thinking about the viticultural, winemaking, or oenological issues at hand.

The basic unit for the interface will be the "facility". Each such object will provide the context for interaction with the system. The main window for the application should be an MDI (Multiple Document Interface) with tree controls on the left (Figure 16). The top tree should display a list of all the appropriate facilities available with contextual nodes under each branch depending on the type of facility. Tree controls are familiar to most users (ubiquitously used throughout many Windows based applications including “Windows Explorer”) and can be tailored to display the desired level of information to the user. They also show a hierarchy of entities within each facility that is easily navigated and intuitive. The second tree control will provide access to system specific functions such as data maintenance and user access. This should only be available to users with appropriate permissions.

On clicking a node within the facility tree, an appropriate MDI child window will appear in the body of the MDI. This window will be maximised and cannot be minimised. Although an MDI window does allow multiple windows to be displayed at any one time, this type of display could lead to confusion for the user. The choice of using an MDI is purely technical.

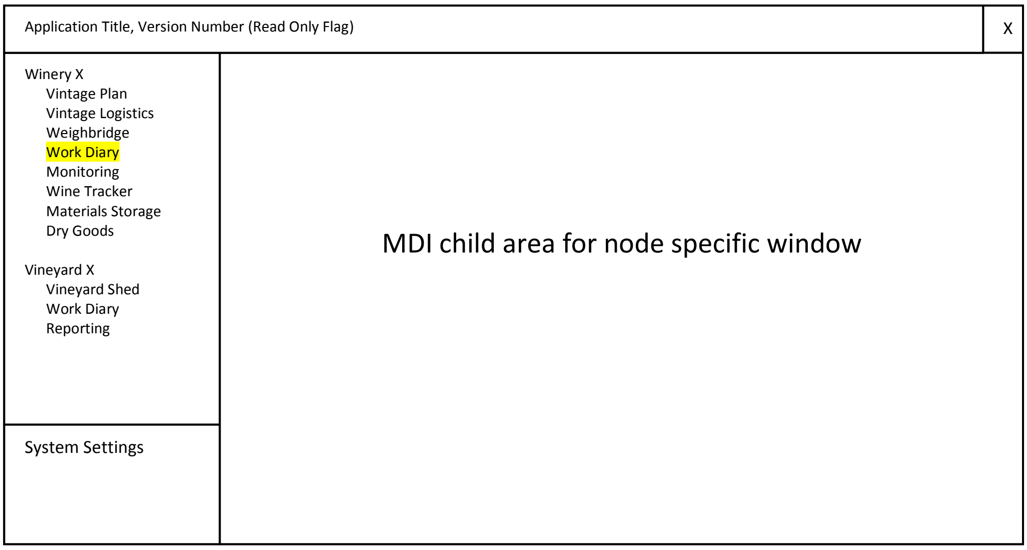


Figure 16. Multiple Document Interface for application

10.1 Vineyards

A Managed Vineyard is defined as a vineyard, vineyard shed with machinery, and staff that is under direct management by the user of the system. The manager is generally responsible for maintaining all aspects of vines, including pruning, spraying, and harvesting, the machinery and materials used to perform these tasks, and the people employed to perform them.

The aim of the interface is to reflect the common tools currently used in the vineyard, to offer a clever, intuitive, and elegant alternative to them, so that it is not only able to record them for easy recall, but enhance their relevance by displaying them in a richer and contextually relevant way. It should enable the user to create and update standing orders and one-off tasks, and provide tools to analyse the status of the vines and fruit to order and justify these tasks.

10.1.1 Main Menu

The main node should display a map of the vineyard, ideally a satellite image, with the blocks, irrigation zones, and any special zones clearly marked. These details will be data driven by tables in the database and will therefore be configurable and changeable without modifying the application software. Special zones are defined as zones within a block that the user may want to demarcate from the rest of the block for special treatment because it is of particular high vigour or the fruit warrants special treatment for disease.

A note on Blocks, Irrigation Zones and Special Zones

It has been found that some irrigation systems use zones that cover more than one block. This was perhaps an economic decision rather than a practical one given that it this removes some flexibility for irrigation regimes. Nevertheless, this has implications for the user interface as well as the data model (to be defined later) because displaying individual data for a block will need to include irrigation data, but this data may or may not be applicable to more than one block. The interface will need to identify which irrigation zone a block belongs to. On the other hand, special zones will need to be confined within a block rather than crossing blocks so that a hierarchy can be maintained

If the vineyard does not use Irrigation Zones then we need to define all blocks within a single zone, and the interface will automatically disregard zones if only one zone exists.

Each zone or block should have a hotspot on the screen to provide a tooltip display of basic data about the item. Ideally the main menu should also allow navigation to the other child nodes.

From each block, a details screen should be generated that displays technical information about the block and allows status information and readings to be entered and updated. This should include...

- Vine Status
- Jobs for the Block
- Yield Estimates
- Harvest Date Estimates
- Parcel Definitions

It should be noted that the alteration of parcel definitions, yield and harvest date estimates will have significant effect on the proposed Winery Vintage Planning and Vintage Logistics screens described below.

10.1.2 The Vineyard Shed

Under the Main node, a vineyard shed node should appear to contain entities that relate to the vineyard as a whole, and are not part of the vineyard; for example, equipment such as harvesters and tractors, weather station equipment, chemicals and materials used in the vineyard. These screens should display appropriate information regarding the state of the entities and allow generation of worknotes and/or input of results from each.

10.1.3 Work Diary

The work diary is a general display of all the jobs that are associated with the vineyard. A date control should allow the jobs to be filtered so that only relevant jobs are displayed. This will include a flag to display only incomplete jobs. It is noted that this screen will be a clone of the same function in the winery facility, and further details are explained there. (*10.2.5 Work Diary*)

10.1.4 Generating worknotes

Jobs in the vineyard will be generated from the main menu as well as the Vineyard Shed node. The list of possible job types will include all those listed in *7.1.2.2 Vineyard Activities*. Business rules will need to make sure that worknotes are attached to appropriate object types depending on the selected object on the screen.

10.1.5 Data Mining – Analyse/Report

It is expected that the Data Mining function here will be a clone of that proposed for the winery facilities, with obvious emphasis on the vineyard blocks rather than wines. In addition, more specific and pre-designated reports that will prove useful should include functions such as

- Fruit used in wines
- Harvest Date evolution
- Yield estimates evolution
- All spraying jobs should be able to be presented in the form of a typical spray diary.

10.1.6 Unmanaged

Unmanaged vineyards will be far simpler to present and manage within the context of the model because in these cases we are only interested in the raw data for harvest dates, and yields, as well as maturity data used to estimate these parameters. It will only be necessary to keep basic block information to help differentiate the derived parcels.

10.2 Wineries

As described earlier, the nodes for the winery facilities are designed to provide a sense of exploration rather than straight functionality. Each node has a deliberate sense of purpose, but does not force any procedural flow on the user. The intention is to allow the user to probe and discover.

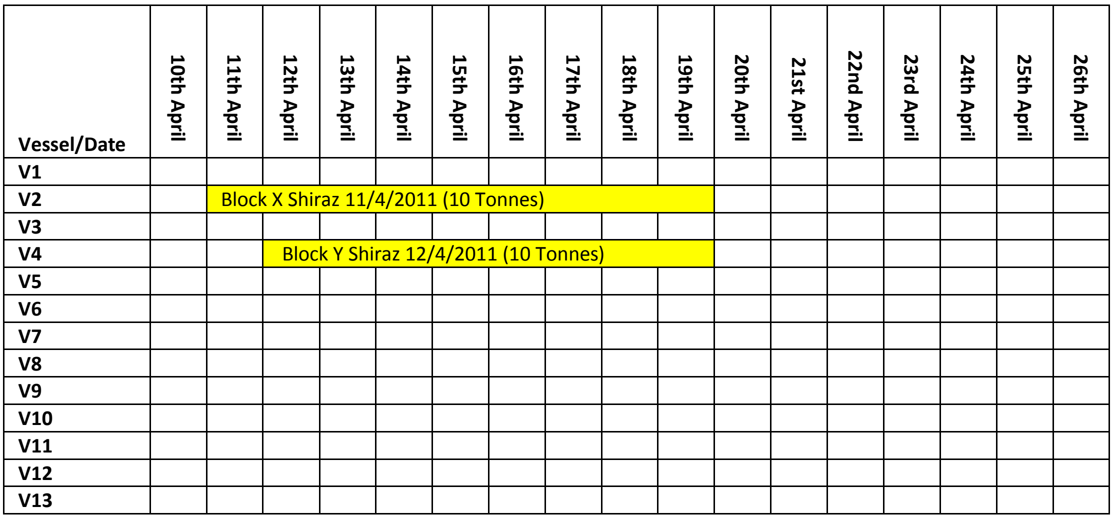
10.2.1 Vintage Plan

The vintage plan screen is a summarised and visual version of the vintage planning documents cited in case study 1 (*7.1.18.1 Vintage Planning*). It employs a drag and drop approach to matching up parcels of fruit to vinification definitions ("vindefs") and to subsequent wine batches. Definitions of these entities are kept deliberately vague and most parameters are not mandatory from a user interface perspective, to allow them to have a diverse range of meaning, depending on the preferences of the user. For example, a fruit parcel may be defined as all the fruit in a block or a small proportion of it. If all the fruit is going to be processed in the same way then there is no need to define more than one parcel, even though it is going to be processed over several days of harvesting. If the individual parcels are going to have very different processing, then it may be in the interests of clarity that this is clear in the vintage plan and that greater detail is required within the vinification definition.

Each parcel can only be linked to a single vinification definition but multiple vindefs can be linked to a wine batch. Information about the fruit parcels is sourced from other areas of the application, namely the block details in the vineyard facilities. If this information is changed in any way, then this screen will need to reflect those changes where possible. If a parcel is deleted, then any links to any vindefs will also need to be severed. Harvest date and yield estimates may also impact on the vessel allocation details and therefore the vintage logistics screen.

10.2.2 Vintage Logistics

This screen will allow the user to assign ferments derived from the vindefs on the vintage plan screen to specific vessels for the expected fermentation period (Figure 17). It will list unallocated ferments and allow them to be dragged onto vessel-time allocations. The vindef will indicate which vessel types are appropriate and any incorrectly allocated ferments will be highlighted. The screen will also highlight overlaps in ferments and allow reallocation to other vessels.



Vessel/Date	10th April	11th April	12th April	13th April	14th April	15th April	16th April	17th April	18th April	19th April	20th April	21st April	22nd April	23rd April	24th April	25th April	26th April
V1																	
V2		Block X Shiraz 11/4/2011 (10 Tonnes)															
V3																	
V4			Block Y Shiraz 12/4/2011 (10 Tonnes)														
V5																	
V6																	
V7																	
V8																	
V9																	
V10																	
V11																	
V12																	
V13																	

Figure 17. Vintage Logistics screen

It should be noted that any changes in either the fruit parcels or parameters within the vindefs may have an effect on the allocations. It is conceivable that some changes may cause these vessel allocations to be invalidated. Indeed, any problems in allocating vessels may need to be resolved by changes in fruit parcel definitions or vindefs. It will be necessary to ensure that the user is notified of any changes, when the cause of the changes is actioned.

10.2.3 Weighbridge

Introducing new fruit into a winery can be a complex process because it requires reference to vineyard, winery and planning entities within the data model. From a user's point of view, the concept of a weighbridge node makes its function obvious and provides a useful and intuitive prompter. The screen should list all the parcels defined for reception in expected harvest date order. When a parcel is selected, details should appear of how much of that parcel has already been delivered and to where. A button should also be available to generate a fruit reception job which should allow the user to specify processing options based on the vindef that the parcel is attached to. The window should restrict the user to specifying vessels defined in that vindef, but not in any particular order. The current state of those vessels should be displayed to allow the user to divide and target the appropriate amount of fruit to each vessel. The fruit processing job should also be seeded with the rest of the information from the vindef including yeast and nutrient selection, analyses and additions, and processing options such as pressing regimes. These options will be changeable by the user. Only the vessel selection is restricted to avoid making too many changes to the vessel allocation plan.

10.2.4 Winery Map

This screen should provide a plan of the winery with two main areas; the tank farm and the barrel hall. The names are there to suggest that the vessels in the tank farm are static whereas those in the barrel hall are mobile, and are not supposed to represent actual areas in the winery, although they may in fact exist. The main purpose of this screen is to allow the user to probe and discover within the winery, and be the first port of call when a visual search is required.

This will be achieved by icons that can represent the types of vessels and the states of the wines in them. Flashing indicators can indicate that incomplete jobs are attached to the vessel and/or wine. Simple flags can indicate the presence of chips, micro-oxygenation sinters, gas blankets, and temperature control settings. The colour and texture of the tank can represent the colour and state of the wine.

For greater details, a tooltip can be used that appears when the mouse pointer dwells on a tank. This should provide text confirmation of the various indicators as well as LIP information of the wine. Double-clicking on a vessel will display a modal window with all the tooltip information plus a list of all the jobs associated with the vessel or wine and provide access to display the current and historic readings for the wine.

Right clicking on the vessel will display a menu which will allow the user to navigate to other screens as well as generate jobs for the selected wine and make any other proposed function available such as the wine groups and flags (*9.1.2.5)*. The vessels will be multi-selectable so that actions like menu options can be applied to more than one wine at a time. Dragging and dropping wines will be interpreted as wine transfers and will seed the transfer generation screen with appropriate setting based on the source and target vessels.

10.2.5 Work Diary

This screen should display a list of all the jobs and associated worknotes for the selected winery. It should allow each job or worknote to be edited, completed, or cancelled, subject to business rules concerning when and if such actions should be allowed. The screen should also allow the creation and maintenance of standing orders in the winery. Any jobs created from standing orders will appear when the work diary is presented. Initially they will require confirmation from the user before they become real jobs. The jobs and worknotes themselves should be displayed hierarchically with each job having its worknotes listed below in order. The list should be colour coded to indicate the status of the worknotes. When a job or worknote is selected, the appropriate buttons for completing, editing, cancelling, should be enabled or disabled.

This screen should also display a diary for the user to use as they see fit. This will allow winery work and other personal or work matters to be diarised in the one place. Diary entries will only be visible to those who generated them.

10.2.6 Monitoring

The importance and significance of concepts like HACCP for the winemaking industry is very much a matter of opinion. This necessitates that any monitoring of the wines from a analytical perspective must be completely configurable by the user. This screen should list all of the current wines in the winery and allow the user to define and associate a monitoring algorithm with each wine. The user has the freedom to create individual or generic algorithms. These algorithms may be as specific and detailed or as simple as required. The user will be allowed to create tests as steps in the algorithm from a set of parameters and define criteria by which these parameters can be judged. The criteria may involve the value or range of the parameter or the date that the reading was taken.

10.2.7 Tracker

This screen is potentially the most powerful part of the application, and will draw the most from the capabilities of the model. In simple terms, this screen ought to be able to display the current state of a wine as well as all the actions, events, and source wines that got it into the state it is in. A simple pyramid structure with the current wine at the top is expected to be adequate. Below this will be the wines from which this wine was derived, linked by a line and the number of the worknote that caused the new wine to be created. One of the key issues that this specification generated was how and when a wine entity is brought into existence and what actions can happen to it before it must necessarily spawn a new wine or wines and retire itself. This is covered in *11.2 Transferring, Racking, Filtering, and Splitting Wines*. Each entity in the pyramid will have its own tooltip to show the state it was in immediately before the subsequent action was applied. It is expected that this screen could become quite large and complicated given the prospect of displaying some wines with very complex ancestry.

10.2.8 Materials

The user interface requirements for these materials are not very demanding. The materials types were divided into categories in *8.2.7 Material tracking* and this will be an adequate means of dividing the screen. Adding new or editing existing instances of particular batches of these materials can be performed through a simple dialog. Deleting a batch should only be possible if the batch has not been used by any completed worknotes.

10.2.9 Dry Goods

The dry goods screen will be a clone of the materials screen, with the categories replaced by the various types of dry goods available.

10.2.10 Unmanaged Wineries

Just like the unmanaged vineyard, an unmanaged winery will simply need to make available a sub set of the available functionality of the fully managed version. In fact the only functions needed are a small subset of the weighbridge function and a very basic version of the tank map. This will enable fruit to be processed into wine and placed into a vessel which can then have basic additions made and readings taken without the generation of jobs or worknotes.

10.3 Laboratories

The laboratory interface should consist of a tree list of available analyses as specified in *9.1.3 Laboratories* with an indication of the number of tests booked. When a node is selected, a list of the incomplete worknotes for that analysis type should appear with fields to enter the analysis results. There will also be an option to enter the result without the calculation parameters in the event that only the result is known, not the calculation variables.

Although these are common parameters used in winemaking to assess wine, some of these parameters have several techniques for their measurement. Often, the calculations are the only difference, but sometimes they differ sufficiently to justify a different interface for input into the system. If possible, some analysis calculations can be data driven, but others will need to have multiple calculation interfaces. New techniques may fit into either category, so occasionally an upgrade to the software may be required.

Unmanaged Laboratories will appear the same as a managed laboratory except that the non-calculation variables option will be enforced.

11. Design Issues and Resolutions

The evolution, recognition, and resolution of some critical issues during this project did not occur in a neat sequence of events somewhere between the specification of the system and the subsequent designing and building of it. Matters like these tend to appear only when the entities involved are built and put through their paces. Such is the case for these issues. Rather than detail the chronology of how they occurred, it is adequate to simply note their existence and describe the problem and its resolution.

11.1 Internal Logistics

In any database system, we need to protect the integrity of the data by ensuring that relationships are maintained. This is particularly important for high level data records that can be referred to by many other records. In the event that such a record becomes redundant, then deleting will cause these referring records to become orphaned from this relative data. For example, if a block of fruit is uprooted and replaced in a winery, deleting that block from the data base will cause all the fruit parcels that came from that block to lose their identity as to which block they came from.

This issue is well known and accounts for the fact that many record types and their object model counterparts have “Valid” or “Active” flags attached to them. This will filter out records that need to be removed from presentation when dealing with current options on the interface

11.2 Transferring, Racking, Filtering, and Splitting Wines

From a data modelling point of view, it is not appropriate to treat a wine in the same way we might model an individual object such as a person or a common discrete object. This is because a wine is never commonly maintained as a discrete object for long. It can be mixed with other wines, or split into separate parts. If half the volume of a wine is pumped from one tank to another then we need to recognise the creation of a new wine, but which tank can be said to be the new wine and which is the original? What should happen if the target tank already has wine in it?

It must be said that a solution was not obvious from the beginning. The idea of a single wine existing from fruit source to bottle is possible but uncommon. At some point most wines are racked of lees, topped with other wines, or otherwise split and combined in some way. A new approach was needed and several unsuccessful scenarios and schemes were put forward to resolve this issue. In the end they all failed the final hurdle which was the ability to successfully maintain a complete historic track of wine entities with traceable LIP information, reading, and location information.

The ultimate solution was to take a very conservative approach; which leads to 3 rules.

1. Whenever possible, create new wine entities rather than attempt to maintain current wines.
2. Only complete transfer of a wine from one vessel to another empty vessel will maintain the original wine.
3. Any lees remaining in the source tank is regarded as wine and therefore does not constitute a complete transfer.

When applied to all the possible scenarios we get the following results...

% Source Wine transferred	Target Tank already contains wine	Old Source	Old Target	New Source	New Target
< 100%	No	Transferred Away	N/A	Yes	Yes
100%	No	Remains	N/A		
< 100%	Yes	Transferred Away	Transferred Away	Yes	Yes
100%	Yes	Transferred Away	Transferred Away		Yes

Table 12. Transfer Scenarios

There are several repercussions that need to be considered when wine is transferred. In particular we need to consider...

1. Existing worknotes need to be re-pointed to the appropriate new wines.
2. Ensure that LIP data is generated
3. Wine readings and settings are inherited appropriately
4. New wines are linked with their source wines.

11.3 Jobs and Worknotes

11.3.1 A Structure

From the beginning, it was clear that the generation of work assignments should not be regarded as a simple task oriented process. Firstly, individual tasks are more often than not part of a larger set of tasks designed to achieve a certain goal. Secondly, that overall goal is an important factor in the motivation and justification of the work in the context of the winemaking process and is therefore essential to the integrity of the cause and effect chain we wish to create and maintain.

The creation of tasks in wineries and vineyards can be complex because, conceptually, there may be more than one step, each step might involve actions on more than one entity within a facility, some actions are dependent upon completion of previous actions, and these sets of actions need to be manageable by the user. It was decided to adopt the convention of naming the overall task a "Job" and each sub-task within each job a "worknote". This was adopted to emphasise the fact that a "job" could be perceived as achieving an overall goal, whereas a "worknote" is commonly used as a unit of work assigned to cellar staff within a winery. The model will contain structures of job types and worknote types as templates for the creation of instances of jobs and associated worknotes.

In simple terms, when a job and associated worknotes are created, there is an initial input from the user. It is not envisaged that there should need to be any input into the system until the job is completed and all the completion parameters are available.

The data model will need to manage all aspects of manipulation of this structure including the configuration and sequencing of all the different types of worknotes under the various types of jobs. It is not expected that the creation of new job types will be a common occurrence after the initial configuration of the system, so this functionality could easily be confined to a system admin role.

It is relatively simple to define all the conceivable types of worknotes possible, as long as the structure is flexible enough to avoid locking-in how the task they describe is to be completed. For example, the addition of SO_2 to a wine might be part of an overall job to rack and return a wine in barrel. The addition of SO_2 might be through the use of a sulphitometer, or by the addition of PMS or SMS. These options should be selectable by the user and part of the worknote structure, but there is no need to create a worknote type for each method of addition.

Creating and editing instances of jobs and worknotes should be available at all appropriate points within the interface. Naturally, not all job types should be available at all facility types so each job type will be qualified by a facility type indicator. However, the underlying worknotes types for each job type will not necessarily be carried out at the same facility. An example might be the collection of samples of fruit for yield estimates or harvest parameters. The first worknote type, the collection of samples, would obviously need to be performed at the vineyard, but the analysis might be carried out at a laboratory where appropriate tools are available. Upon creation of the job and worknotes, the user will need to identify where such worknotes are to be performed.

11.3.2 A State Machine

A state machine has been developed to ensure that all worknotes are performed in a logical sequence. Each worknote is given a status as it proceeds from creation to completion. The various states include...

- Waiting* - Prior to the start date for the job, allowing for pre-booking of worknotes
- Standby – Awaiting completion of a previous worknote.
- Attention – Previous worknotes is complete but this worknote requires additional information (based on results or previous worknote) before it can be completed
- Ready* – Worknotes can be performed.
- Active* – being performed
- Complete* – Done
- Cancelled* – Called off

* The status for the job is based on the status of its constituent worknotes, and will be one of these.

The user has the right to modify various parameters associated with the job or its worknotes, including the cancellation of any of the worknotes or the complete job. Obviously, once a job has begun, the state machine will need to enforce data integrity and thereby restrict some modifications through the interface. By the same token, the system should not preclude completion of dependent tasks before the tasks on which they depend are completed. This is because the state of the system might not reflect reality. Often, particularly during vintage periods, tasks are completed very quickly or performed sequentially at such a rate that it is unrealistic to expect the user of the system to have to race off to their office to update the system before beginning the next step in the sequence.

A good example to illustrate this is a fruit reception job. It involves several steps, some optional, some dependent on others, and multiple facilities.

The job is to process an incoming parcel of fruit. Many of these decisions will be seeded by the Vinification Definition in the Vintage Plan. These are the options available.

- Chill or rest the fruit prior to processing.
- Optionally run initial analyses prior to processing to establish need for SO_2 or H_2T additions.
- Assign appropriate vessels – usually predetermined in Vintage Logistics Plan.
- Crush, Destem, both or neither.
- Press first or ferment on skins (white or red).

- Make optional additions based on results of optional analyses, either at the must bin, at the press, or at the fermentor vessel.
- Optionally inoculate with yeast (and nutrient) either at must bin, at the press, or at the fermentor vessel.

11.3.3 Exceptions

Here we see several complicating factors that need to be catered for within the data structures and the interface.

1) Some worknotes are optional.
2) Analyses worknotes might be performed in a laboratory (a distinction necessary for the data model to ensure data integrity)
3) Some worknotes are optional based on the selection of other worknotes.
4) The sequence of worknotes is critical to make sense of the overall process.
5) The additions, for example, might be based on factors other than the analyses, such as taste.
6) It is conceivable that the whole process is performed before the user has a chance to get to the system to update, so post-facto data entry is a distinct possibility
7) Winemakers change their mind, and so the system will need to allow for changes to occur when completing the worknotes that reflect last minute changes in options, sequences, or other parameters.

In summary, it is not expected that all jobs or worknotes will be able to conform to a standard that will completely allow a data-driven solution. If possible, common solutions might be found to allow for simple situations. It is conceivable that some tasks do not necessarily require results to be entered other than the fact that they have been performed. The aim is to ensure that the structures for the data model are flexible enough to cater for all likely possibilities, and yet ensure that the integrity of the data is maintained so that it remains meaningful and reflects the reality of what actually occurred. The price for such functionality is that the business logic within the classes will need to react to a conceivably large number of flags and driving parameters to ensure appropriate behaviour, and the configuration of new job types and worknotes will contain a degree of redundancy for less complicated situations.

11.3.4 Scope

Another consideration is the fact that although winemakers do like to collect sets of tasks and regard them as a single job, these sets tend to be within a discrete scope, either with respect to the type of individual task, or kept within a distinct wine. If a job becomes too broad in its scope then it loses its sense of purpose or function to the winemaker. All of the actions to be performed on a single wine for the day do not need to be within a single job. If a wine is pressed, then that is a job, although it is conceivable that the winemaker will want to perform more actions upon that wine after the pressing. The point is that the job for the pressing will never cover more than the pressing alone, and subsequent actions will necessitate more jobs to be generated. This is the preference of the winemaker, not a restriction of the model. Each situation will have a distinct set of circumstances, but the need for broad jobs with multiple different tasks for multiple entities is neither required by the user or necessary for the design to cover.

This raises and resolves another issue. Should the winemaker be able to create tasks for wines when those prospective wines will only be created once other jobs are completed? If the winemaker wishes to press two wines into a single wine, and then perform tests or other tasks on the new wine, then how can the model cope with this, given that the new entity has not yet been created? If only one wine is pressed then the model will be able to reconcile any subsequent tasks by redirecting them to the new entity, but two wines becoming one makes this redirection ambiguous. Once again, this is an issue of scope for the job. It is expected that the jobs themselves will become more elaborate as users become familiar with the concept and the capability of this structure.

Jobs may not be available or suitable for all enterprises. So it will be necessary to be able to disable some jobs from the database and have the application read this to disable the appropriate options from the interface. It is deemed not necessary to have this setting on a facility level because this would be over-engineered. If a job is not appropriate for a facility (say, a particular laboratory analysis for a laboratory), then the user can avoid selecting this option when the time comes to select the appropriate facility when the job is created.

11.3.5 Not just for Cellar Notes

The creation of a job or worknote does not necessarily mean that they ever need to be printed out. The structure can and should also be used as a means of recording events that occurred without necessarily having the formality of a printed copy circulating through the facility. For example, laboratory analyses jobs need never be printed out, but the job/worknote data structure is perfectly appropriate to record the event.

11.3.6 Automating follow up jobs

The continuity of a job could appear to be enhanced by the option to allow further worknotes to be added to a job after some of them have been completed. So, for example, if we have a job that consists of an acid tasting trial, then an optional acid addition, then another optional acid tasting trial after that, it might be argued that the user ought to have the option to add an extra acid addition worknote, and so on. The danger in this scenario is that jobs might tend to push the boundaries of the accepted scope as discussed before (*11.3.4*). It is conceivable that the state engine will not be able to decide when it is appropriate to complete a job. In the end it is believed that keeping jobs within discrete boundaries is more manageable for the user and the data model, especially when it may also compromise other functions such as the wine tracker and the state engine.

11.4 Barrel Sets

The concept of a barrel set was intuitive to some winemakers but the implications of managing such an entity had never been contemplated properly. The danger is that if we start introducing rules for their management, then this might alienate users unfamiliar with the concept. Unfortunately we cannot please the users at the expense of logic. As a result, the movement of vessels in and out of barrel sets needs to have some business rules designed to ensure that there is no ambiguity about what is expected when a movement is performed. The primary rule is that moving a vessel is not the same as moving any wine that is in it. In effect, a barrel set is like a club where membership inherits the clubs attributes rather than the vessel taking possession of the clubs assets. One of the properties settable for a barrel set is an “Unused” flag which indicates that the barrel set is purely there to contain unused vessels, and will therefore be treated differently by the business rules. The table below (Table 13) summarises the possible scenarios and outcomes that need to be applied in the business layer. Red movements are not allowed and will generate the indicated error. Blue movements are allowed.

	Target			
	Full	Empty	Empty and Unused	Stand Alone
Source Vessel State				
In Full BS	**NO (1)**	**3**	**3**	**NO (2)**
In Empty BS	**2**	**1**	**1**	**1**
Empty and in unused BS	**2**	**1**	**1**	**1**
Stand alone and full	**NO (1)**	**NO (3)**	**NO (3)**	**Do Nothing**
Stand Alone and empty	**2**	**1**	**1**	**Do Nothing**

Table 13. Barrel set manipulation scenarios

1	Simple reassignment of vessel to new barrel set
2	Assign to new barrel set. Inherits wine by association
3	Simple reassignment of vessel to barrel set. Wine inheritance lost.

NO 1	You cannot put these vessels (which have wine in them) into another barrel set with wine in it
NO 2	You cannot split this wine here. Use the main screen to perform a transfer. 1. Create a new Barrel set 2. Move the barrels you want to split to the new Barrel set 3.Do the transfer
NO 3	Because this is a lone vessel You need to transfer the wine from this vessel before you can move it into a barrel set.

11.5 Readings Inheritance

When some or all of a wine is transferred into another, should the new wine inherit parameters based on the values of the source wines and their proportions? This cannot be assumed because of the complex nature of the chemistry. Sulphur dioxide levels can be significantly affected by the transfer process, even if it is just a straight transfer. Similarly, pH levels can never be assumed to behave in a “mass balance” sense because of buffering and other complex interactions, not to mention that it is not measured on a directly proportional scale, nor is it a state variable. The risk involved in assuming parameter inheritance is too great to consider adopting. In fact, the absence of a wine reading may prompt the winemaker to go ahead and get a fresh reading. The provision of a Wine Tracking function should allow the user of the model to investigate the origins of any particular reading type for a wine based on its source wines. This function will allow the cause and effect of processes and actions to be traced, but false assumptions and assumed readings will complicate this function and its utility to the user. It was decided that only the remaining source wines from transfers and racking, where the remainder is not lees, should duplicate lab readings for new wines created. Complete transfers to an empty tank will automatically retain all readings because the original wine is not replaced, just relocated. However the system will also flag some parameters as “assumed” in the event of any transfer of a wine to another vessel.

The addition of any additive can affect some parameters too. Each additive type can be flagged as affecting any of the major parameters for a wine, and when added, these parameter readings will be flagged as “suspect” for that wine.

11.6 What colour is the wine?

The determination of the colour of the wine for representation on the GUI is problematic. Most red grapes, but not all, have white pulp and can be made into red or white wines. This is controlled by skin contact time after crushing. A good example is the common use of Pinot Noir grapes used for both red wine and white sparkling wine. If pressed immediately after crushing or whole bunch pressed, the juice may have a pink tinge, but this is usually fined out or decolourised during the fermentation process. As a result it is not possible to determine the colour of the wine, or rather the way that the user wishes to view the colour of the wine and the way it will be treated, just from the grape variety. For this reason, if the source parcel of fruit has red skin and is pressed immediately after initial processing, a “Treat as white” flag is available to the user to indicate that the system should treat the wine as a white wine. This value can be regarded as a parameter that will be inherited by any subsequent wines derived from this wine.

11.7 A Configuration Application

The requirement to have the ability to configure the system soon made it obvious that a separate application would be needed to allow the users to update the static code-set data and manipulate the configurative data for the system. This is only mentioned here because it has a minor impact on the nature of the data model because it necessitates “update” and “refresh” type methods for some of the entities that are only used by this application and not the main application, which would regard these sets as static. The list of entities that can be created, edited and updated, and even deleted include...

- Users
    - Permissions
    - Logon status
- Vessels
    - Names
    - Type and category
- Wine Additives
- Dry Goods
- Pressing Regimes
- Grape Varieties
- Regions
- Facilities
    - Managed and Unmanaged
- System Constants

12. THE DATA MODEL DESIGN – FROM THE BOTTOM, UP

This chapter describes the class object structures, functions and attributes, and source database tables used to describe and populate the data model entities. They are listed in logical groups that best describe the sub-systems of the model. A brief description of the business rules that apply to the objects appears in the next chapter which will explain how the application uses the data model. Many objects are instantiated in collections that have standard methods to add to, delete from, and search the collection for items. The collection entities are not listed here unless they have specific and unusual functionality beyond these standard methods.

12.1 GLOBAL PARAMETERS

The Global Parameters entity is the first object to be instantiated. It contains many of the configurable constants used throughout the rest of the application. It also provides a container for some common system-wide code-sets. (Figure 18)

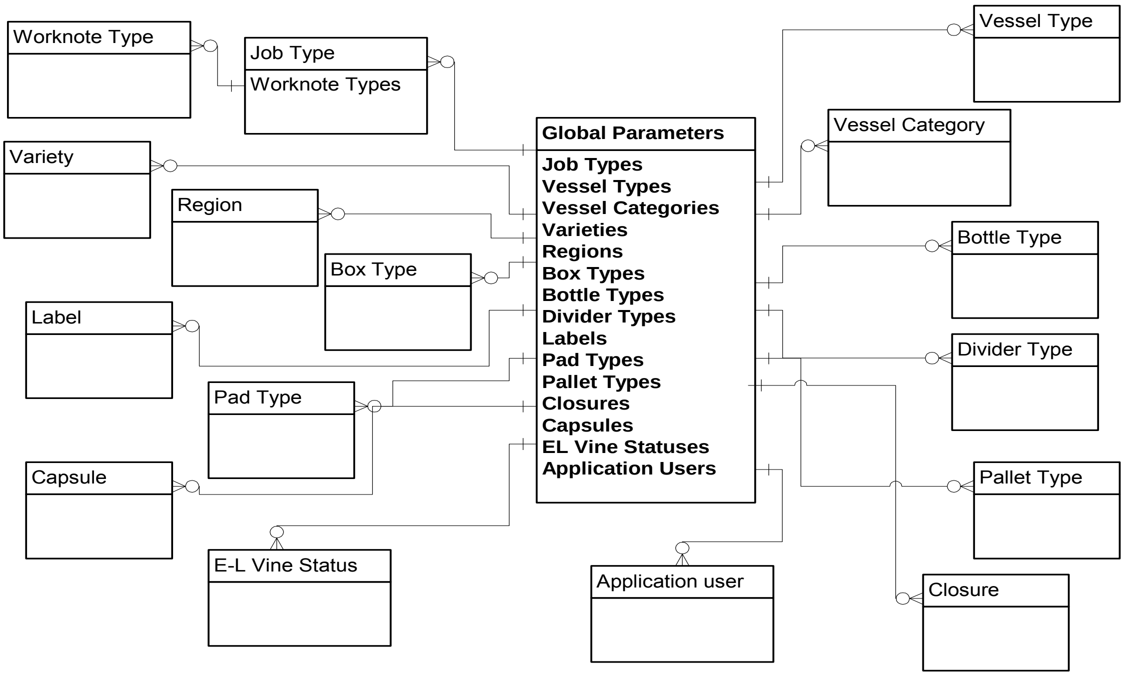


Figure 18. Global Parameters object diagram

12.1.1 Properties

Property Name	Description	Sourced from
SplashImage	Name of image used on the splash screen for the application	t_VWMS_SystemConstants
EnterpriseName	String containing the name of the Enterprise. Used mainly in reporting.	t_VWMS_SystemConstants
IconPath	The full path of the folder used to contain all images used in the application	t_VWMS_SystemConstants
EnterpriseLogo	The name of the icon file for the enterprise. Used mainly in reporting.	t_VWMS_SystemConstants
BarrelDepreciation	The enterprise-wide rate of depreciation (per annum) of barrels while in use.	t_VWMS_SystemConstants
LogToFile	Flag indicating whether logged information should be dumped to a log file	t_VWMS_SystemConstants
LogToDB	Flag indicating whether logged information should be dumped to the database	t_VWMS_SystemConstants
WhiteDragIcon	Icon file used on the interface when a white wine is dragged.	Icon path + hardcoded filename
RedDragIcon	Icon file used on the interface when a red wine is dragged.	Icon path + Hard Coded filename
BottleIcon	Icon file used on the interface for a bottled wine.	Icon path + Hard Coded filename
Current Date	The date to be used for all entry date and time stamping. This is particularly useful for retro-entering data. This will be changeable from within the application.	System Registry
Current Vintage	Used to tag data elements as belonging to the user specified vintage rather than using the current date year. This will be changeable from within the application.	System Registry
Lab Messages	Boolean indicating the presence of messages from the laboratory that need to appear on the interface.	t_VWMS_SystemConstants
Last Compacted	The date and time the database was last	t_VWMS_SystemConstants

	compacted.	
Compact Every Hours	The number of hours since the last database compaction that should instigate another compaction.	t_VWMS_SystemConstants

12.1.2 Collections

The following collections are based in the global parameters entity. Most of them are simply code-sets of reference data that are fairly static. However the job types and vessel categories entities also have collections within them.

12.1.2.1 Job Types

The job types collection contains details for all the job types defined in the system. These apply across the enterprise, because it is possible to have a job that refers to more than one facility within the enterprise, so they need to be consistent across the whole application.

Property Name	**Description**	**Sourced from**
JobTypeName	A short name for the Job Type	t_CS_JobTypes
FacilityType	ID if the facility for which this job type is used.	t_CS_JobTypes
Enabled	Flag indicating that this job type is available to be used	t_CS_JobTypes
Description	Longer description of the job type	t_CS_JobTypes
ID	Unique Integer Identifier for the Job Type	t_CS_JobTypes
TargetSpecific	Flag indicating whether the job type is targeted to a specific entity	t_CS_JobTypes
GenerallyAvailable	Flag indicating whether the job type is available to be generated from a general list rather than only available in a specific hard-coded context. This usually indicates that the creation of the jobs and their worknotes do not require any further details to be specified other than the target object.	t_CS_JobTypes
StandingOrder	Flag indicating whether the job type can be turned into a standing order	t_CS_JobTypes
CompleteInSequence	Flag indicating whether the worknotes for this job type need to be completed in sequence.	t_CS_JobTypes

NoDialog	Flag indicating whether the job requires any dialog interaction from the user before it can be generated,	t_CS_JobTypes
Worknote Types	Collection of worknote types for the job type (see below)	t_CS_WorknoteTypes

12.1.2.2 Worknote Types

This entity describes the functional properties of worknotes associated with the job type that it belongs to. The Worknote Types collection in each Job Type entity is loaded based on the *t_JobWorknotes* table which act as a cross-linker to indicate which worknote types and their sequence, applies for each job type. This means that it is possible for a worknote type to be used in more than one job type, and even more than once within an individual job type.

Property Name	**Description**	**Sourced from**
ID	Unique identifier for the worknote type	t_CS_WorknoteTypes
FacilityType	The facility type to which this worknote applies	t_CS_WorknoteTypes
TargetObjectType	The type of the object that this worknote type applies to.	t_CS_WorknoteTypes
Description	A general description of the worknote type	t_CS_WorknoteTypes
Sequence	The order that this worknote instance will appear in the list for the job type	t_CS_JobWorknotes
Instructions	Text description of SOP or instructions for completing the worknote	t_CS_WorknoteTypes
Editable	Flag indicating whether the worknote can be altered after it is created	t_CS_WorknoteTypes
RequiresResults	Flag indicating whether the worknote needs to have a result specified before it can be completed	t_CS_WorknoteTypes
Addition	Flag indicating that the worknote is a wine addition.	t_CS_WorknoteTypes
Transfer	Flag indicating that the worknote involves a transfer of wine.	t_CS_WorknoteTypes
CanBeCancelled	Flag indicating whether the worknote can be cancelled separately from the other worknotes in the job.	t_CS_JobWorknotes
NeedsRatification	Flag indicating whether the result from this worknote needs to be ratified before it is officially accepted as data	t_CS_JobWorknotes
RequiresParameters	Flag indicating whether the worknote needs to have parameters specified before it can be started	t_CS_WorknoteTypes
RequiresTime	Flag indicating whether the worknote is time critical.	t_CS_WorknoteTypes

12.1.2.3 Vessel Categories & Types

Each vessel has a vessel type. A Vessel type entity describes particular properties of that vessel type to do with volume, what it can be used for, and how it should be presented on the user interface. Vessel types are grouped into categories which describe more general properties. By association, each vessel inherits the category properties from its type.

12.1.2.3.1 Vessel Categories

Property Name	**Description**	**Source**
ID	Unique identifier for the vessel type	t_CS_VesselCategories
Category	Description of the Vessel Category	t_CS_VesselCategories
OnSkinsFermentor	Flag indicating that vessels in this category can be used for on skin fermentation	t_CS_VesselCategories
DefaultCapManagementWNID	The Worknote Type ID of the default cap management. This is used by the standing order generator to create a cap management job and worknote appropriate for the vessel category	Refers to ID in Worknote Types entity

12.1.2.3.2 Vessel Types

Property Name	**Description**	**Source**
ID	Unique integer identifier for the vessel type	t_CS_VesselTypes
Description	Description of the vessel type	t_CS_VesselTypes
Code	Short alphanumeric code used at the start of the name of a vessel to identify the vessel type. Used to generate a default name for a vessel but is not mandatory.	t_CS_VesselTypes
CapacityTonnes	The capacity of the vessel type in tonnes when dealing with on skins fermentation	t_CS_VesselTypes
CapacityLitres	The capacity of the vessel type in litres when holding wine or juice	t_CS_VesselTypes
CategoryID	Refers to the ID of the Vessel Category that this type belongs to	t_CS_VesselTypes
Icon	Name of the icon file used on the user interface for this vessel type	t_CS_VesselTypes

TypeUllageAllowance	The minimum number of litres below the capacity that indicates that the vessel is on ullage.	t_CS_VesselTypes

12.1.2.4 Other Collections

The remaining entities, all in collections in the Global Parameters object, are relatively simple code-sets used to list common items used throughout the data model. They tend to be quite static, and are loaded when the application is started but never need to be refreshed during the lifetime of the instantiation of the application. Their constituent properties reflect the fields in the data base tables from which they are sourced, so there is no need to go into further detail here.

EL Vine Statuses	The Eichhorn-Lorenz vine status table includes image file details for display on the interface
Varieties	41 most common grape varieties used in Australia
Regions	Otherwise known as Geographical Indications (GI) this is the list of recognised regions in Australia
Box Types	Box types including capacity and the number that makes up a pallet.
Bottle Types	Colour, style, Volume, punt and maker of bottles supplied to the enterprise.
Divider Types	Available dividers
Labels	Front image file, back image file, and description of labels available for use during bottling
Pad Types	For laydown boxes, a pad is generally used to divide the layers.
Pallet Types	Available pallet types during bottling, Generally distinguishes ownership.
Closures	Available closures, with a flag indicating that it requires a capsule too.
Capsules	Available capsules

12.2 Application Users

The Application User entity is a single instance entity used by the model to keep information about the user logged into the system. The object is initiated by using the Logon Method which uses a system API to determine the network logon userid. This is used to verify the user and means there is no need to perform a logon to the system.

A collection of this entity is also used in the Global Parameters object to retrieve information about all the users of the system, to determine which facilities they have logged out and to set the “KillMe” flag to logout users remotely.

Property Name	Description	Source
UserName	Full name of the User	t_Users
CurrentFacilityLocked	ID of the facility they have locked out for editing	Set by Business Layer
UserID	User ID string of user	GetUserName API
LoggedOn	Flag indicating that the user is logged on	t_Users
Editor	Flag indicating whether the user has Update permissions	t_Users
Admin	Flag indicating whether the user has Admin permissions	t_Users
SuperAdmin	Flag indicating whether the user has Super Admin permissions	t_Users
LockedFacility	Name of the facility they have locked out	
KillMe	Flag set in database to tell the user to logoff automatically	t_Users
AskBeforeExiting	Flag indicating that the application should present an "Are you sure?" message before shutting down	t_Users

12.3 Data Layer

The data layer is not an entity in the sense of an object used in the model but is rather an object used to encapsulate and contain all of the communications between the data model and the database. Apart from keeping all the communications and SQL in one place, it also acts as a candidate for implementing distributed processing by creating it as a separate service on a database server.

When it is initialised it immediately retrieves the database name and path from the system registry, and opens a connection. This connection is kept open but can only be referenced within the data layer object. The only public property is the Database version string which is displayed on the splash screen and at the top of the application MDI form.

There are three main function categories performed by this object.

CompactDatabase Access databases expand when being used and need to be regularly compacted to ensure they do not get too bloated. The empty database is about 5Mb, and each vintage adds another 7 Mb. However, when the database is used it can often expand out a further

20 Mb just through the internal workings and memory/space management. This function requires that the database is disconnected so it needs to be performed when no one is logged in.

LoadMainData The sequence of events in loading the data model is slightly more complex than might be assumed. This is because some objects need to reference other objects when they are initialised, but these reference objects may not have been loaded yet. For example, when the wineries are loaded they need to reference the vineyards, but the vineyards also need to reference the wineries. It is necessary to load the data in stages and refresh some objects affected by this paradox.

All the other functions in this object fit into the third category and are used to retrieve, edit, delete, or create database records for each of the various entities in the object model. The data layer is therefore the only other function familiar with the inner structures of the model entities, apart from the objects themselves. However these functions do not contain any business rule validation and perform simple translations of object data into database table data and vice-versa.

12.4 Enterprise Objects

The Enterprise object is a container for enterprise-wide and enterprise-specific entities, as opposed to the global parameters object which is used for generic industry information. The main collections concern the facilities defined for the enterprise. The rationale is that it is more convenient to load the facilities into collections of their own types rather than a single collection because the collection can be given facility type specific methods and parameters that encapsulate the functionality. Each facility type will be detailed separately, later. It is also used to collect Jobs and worknotes, Fruit Reception Jobs, diary entries, standing orders, system messages, and tracker display items.

\

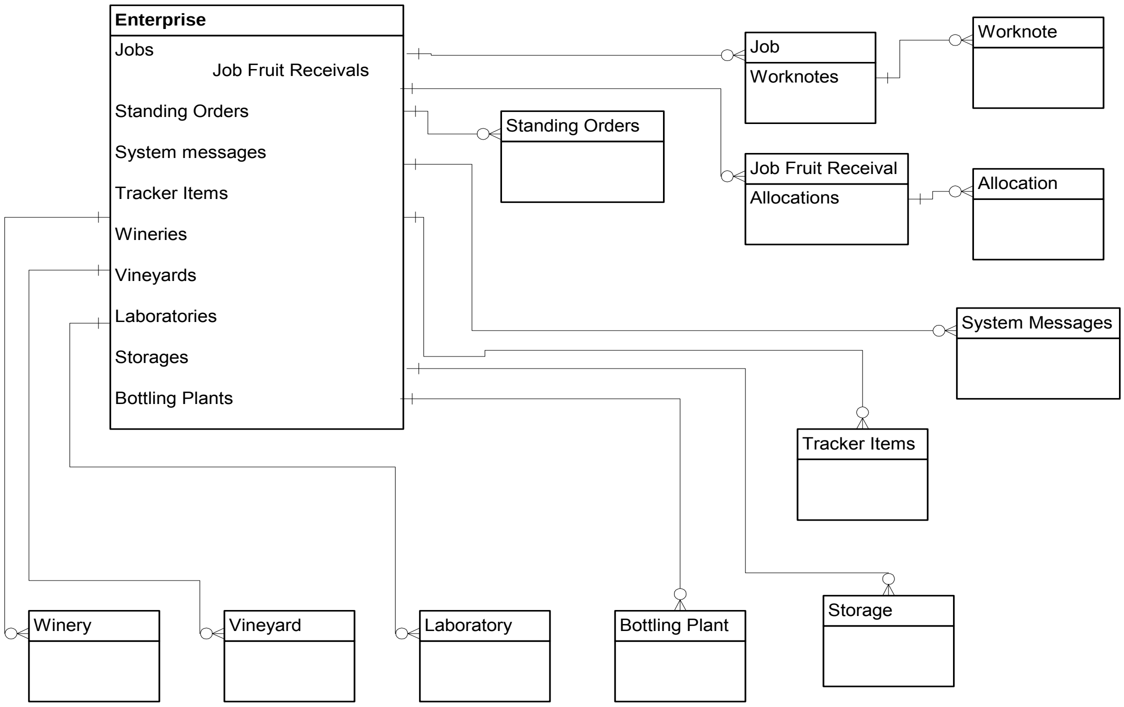


Figure 19. Enterprise Objects

The enterprise object has its own methods...

LockFacility, FacilityLockedBy: Facilities are locked out to the user when no-one else has them locked out. This ensures that only one user is able to update anything in the facility at any one time. These methods are used to handle these functions. They read from the database rather than rely on pre-loaded values, to ensure that the actions of other users are recognised immediately.

GetWineryName, GetVineyardName, GetStorageName, GetLaboratoryName, GetFacilityName are functions used to retrieve the names of the various facility types given their ID. The generic version "GetFacilityName" looks in all the facility collections for the matching facility ID.

12.4.1 JOBS

The Jobs object is a single object declared and instantiated in the Enterprise object. It consists of two collections and several methods use to manipulate items in these collections. The first collection is used for all the individual Job entities, the second collection is for all the Fruit Reception Jobs which are specialised Job structures to contain parameters specific to Fruit Reception Jobs.

12.4.1.1 METHODS

CreateJob	Creates a blank job in the database and instantiates it in the jobs collection.
FindWorknote	Used to find a worknote with the matching ID by searching all the jobs in the collection
CurrentScumRun	A scum run is a special type of job that can only have one active instance at a time. This function is used to identify that instance
ConfirmJob	A job can be in a state of "Waiting to be confirmed" if it is generated from a standing order. This converts that job to a Ready state
FindExistingSOJob	This function is used to identify any jobs that have been generated from a standing order for a particular day to ensure that standing order jobs are not duplicated.
RollIncompleteJobs	Method used to change the target for worknotes in current, incomplete jobs when the current target has been replaced by another with a different ID

Jobs, as described earlier, are a set of tasks designed to achieve a certain outcome. Each job is a specific instance of a set of tasks. The structure is such that it can represent all of the different types of jobs conceivable by the user. They are driven by the Job Types object which specifies some generic properties for the job. More specific parameters are maintained as properties or attributes. Worknotes are stored in a collection within the job object.

12.4.1.2 JOB PROPERTIES

All of the properties for a job, apart from the ID which is generated when the job is created, are set contextually by the business rules within the application, depending on the users input. All values are stored and retrieved from the *t_Jobs* table, apart from Status which is computed based on the business rules in the SetStatus function in the class.

ID	Unique Identifier of the job
JOBTYPE	ID of the job type. References ID in the JobType collection in Global Parameters
FacType	The ID if the Facility type that this job belongs to.
Heading	Text for the heading for the job
StandingOrder	The ID of the Standing Order that caused this job to be generated. Blank if user generated.
Vintage	The current vintage as specified in Global Parameters

DateTimeCreated	The date and time the job was created
Status	Calculated by the net effect of the constituent worknotes
SpecialComments	Text field for user input regarding any exta information for the Job
Steps	The number of logical steps in the job. This is not the same as the number of worknotes.
TargetSpecific	Flag from Job Types indicating whether the job is a general job or targeted at a specific entity within the enterprise.
CompletionDate	The date the job was completed. This will be the same as the date the last worknote was completed.

12.4.1.3 Job Methods

The following methods available in the Job object are primarily concerned with maintaining the object or worknotes within its collection. In general, business rules in the application have already determined whether these functions should be called.

DeleteAllButFirstWorknote	This function is used to rebuild a job and worknotes when the job is edited and fundamentally changed. As the names suggests, it deletes all but the first worknote (a job must contain at least one worknote) so that the generation process can be processed again without the need to formally modify all the existing worknotes. It is a pragmatic shortcut and used to simplify the process of modifying an otherwise complex structure.
DeleteWorknoteIndex	Deletes a worknote given its index.
DeleteWorknote	Deleted a worknote given its ID
Save	Call the Data layer function to save the Job
SetStatus	This is the status machine for the job. It parses all the worknotes and determines the appropriate status for the Job as an overall status.
CreateWorknote	Function to create a blank worknote in the database and instantiate it in the collection
CompleteWorknote	Function to set and complete a worknote. The worknote type determines how this is carried out and may require user input via a sequence of dialogs.
PrevWorknote	Returns the ID of the previous worknote in the sequence given a specified ID
NextWorknote	Returns the ID of the next worknote in the sequence for a specified ID

CancelJob	Cancels all the worknotes in the job and the job.
CancelWorknote	Cancels a specific worknote in the job.

12.4.2 Worknotes

A worknotes collection exists within each individual Job object. Apart from the usual collection manipulating functions, this object also has two methods, FindWorknoteIDForTargetIDOfType and FindWorknoteIDOfType which are used to identify worknote objects according to specific criteria. The worknote object is simply a reflection of its source database table *t_Worknotes*.

jobID	The ID of the Job to which the worknote belongs
FacilityID	The ID of the facility that this worknote applies to
TargetID	The ID of the object that the worknote applies to.
Heading	A text heading for the worknote
Vintage	The current vintage as specified in Global Parameters
DateTimeCreated	The date and time the worknote was created
RequiredStartDate	The date that the worknote becomes active
CompletedDate	The date and time the worknote was completed
Sequence	The step or sequence of the worknote for the Job
WorknoteType	Refers to the ID of the worknote type object that this worknote belongs to
RequiresResults	Flag indicating whether the worknote requires the input of any results before it can be completed
ID	The unique identifier for the worknote
Status	The status of the worknote as computed by the status engine
P1-15	Parameters 1 to 15 used to specify details of the worknote. Their meaning is specific to the worknote type.
R1-12	Results fields 1 to 12 to allow results to be stored. Like the parameters, their meaning is specific to the worknote type

12.4.3 Fruit Reception Jobs and Allocations

The reception and processing of fruit is the most complicated form of a job in the model. Each parcel of fruit can potentially be allocated to many vessels and each allocation may not require all the processing stipulated by the job. This was alluded to in *section 10.2.3 Weighbridge* which justified the use of a separate screen to manipulate such jobs. This means that each fruit reception job gets a fruit reception job object as well as a regular Job object. This separate object contains extra information about the fruit processing, including allocation entities with information about how the fruit was allocated to individual vessels. There are two tables that store this information in the database: *t_Job_FruitReception and t_Job_FruitReception_Allocations*.

12.4.3.1 Fruit Reception Jobs Collection

The collection for fruit reception jobs has three methods.

GetAllocatedTonnageForVessel	This function works out how much fruit was allocated to a specific vessel. It loops through all the vessel allocations of all the fruit receival jobs looking for matches according to various parameters passed into it. This function is quite versatile and is used for identifying current and past allocation of fruit.
FindJobFruitRec	This method identifies fruit receival jobs associated with a Job ID
FindAllocationForWine	This method finds the allocation of fruit that created the wine. This works because each time more fruit is added to a ferment, it becomes a new wine.

12.4.3.2 Fruit Reception Job

This object is used for an individual fruit reception job. It contains all the parameters that apply across the board to all the allocations

ID	Unique identifier of the fruit receival job
ParcelID	The ID of the plan fruit parcel for this recieval job
WineryID	The ID of the winery where the fruit is being received
jobID	The ID of the Job record that this fruit reception job applies to.
ExpectedTonnes	The number of tonnes of fruit expected. This is to two decimal places.
ExpectedYield	The expected yield of liquid from the fruit measured as a percentage of litres per kg
Reception	The ID of the fruit reception equipment to be used of this job.
HarvestMethod	There is a choice of two settings; 0= Machine, 1 = Hand
PMSPerBin	Grams of PMS to be added to each bin prior to or immediately after harvested.
OtherBinAdditive	ID of additive to be added to each bin prior to or immediately after harvested
OtherBinAdditiveRate	Rate of addition per bin of additive from above
Bins	The number of bins that made-up the parcel of fruit. This is used to

	calculate the total additions of PMS and or other additive.
Chill	Flag indicating that the fruit ought to be chilled before it is processed.
Rest	Flag indicating that the fruit ought to be rested before it is processed.
Process	Choice of 2 = Destem Only, 3 = Crush and Destem, 4 = Whole Bunch Press
AnalyseSO$_2$ AnalysepHTA AnalyseTempBaume AnalyseMalic AnalysePAA	Flags to create various analysis worknotes within the Job
AddSO$_2$ AddH$_2$T	Flags to create SO$_2$ and Tartaric Acid addition worknotes within the job.
Yeast	ID of yeast to be added to the fruit
Nutrient to fruit	ID of Nutrient to be added
YeastRate	Rate of addition of yeast
NutrientRate	Rate of addition of nutrient
AddH$_2$TWhen AddSO$_2$When YeastWhen	Indicators for when the additives should be added to the fruit. 7 = At Must Pump, 8 = Add at Press, 11 = Add at Fermentor
Press	Flag indicating whether the fruit is to be pressed during processing
PressRegime	ID of Pressing regime to be used during processing
PressSplit	Flag indicating whether the juice from processing is going to be split into Free-run and pressing components to be stored in separate vessels
AsWhite	Flag indicating that the juice should be regarded and displayed as white juice/wine

12.4.3.3 Allocations

The fruit reception job vessel allocations collection is used to store the individual vessel allocation objects from the parcel being received. It has only one significant method, FindAllocationForWine, which finds the allocation of fruit that created the wine specified. This method looks only in the allocations of the fruit reception job it belongs to, not all the fruit reception jobs.

The allocation object contains only a few properties

ID	A unique identifier for the object
JFRID	The ID of the fruit reception job
WineID	The ID of the wine created by the processing of the parcel fruit for the vessel
VesselID	The ID of the vessel
Tonnes	The number of tonnes to two decimal places of fruit.
Drain	A flag indicating that the vessel is to receive drained juice rather than must.
Analyse	Flag indicating that this allocation will have the selected analyses specified in the parent fruit reception job performed

Additions	Flag indicating that this allocation will receive the selected additions specified in the parent fruit reception job
Yeast	Flag indicating that this allocation will receive the selected yeast specified in the parent fruit reception job

12.4.4 Standing Orders

Standing Orders are templates for jobs. They contain enough information to allow a job and associated worknotes to be generated when required, based on the parameters stored with this object. This data is stored in the *t_StandingOrders* table

ID	Unique identifier for this object
Active	Flag indicating whether this standing order is active. This over-rides the star and expiry dates
JOBTYPE	The ID of the type of job that should be generated for this order.
SpecialComments	Text comments that will be put in the job when it is generated
LabID WineryID VineyardID StorageID	The ID's of the facilitities to which this standing order applies. Generally only one of these will be set.
Target	The ID of the target object (if required) for the job. The type of object is specified in the Job type object.
Basis	Repeated Daily =0, Weekly = 1 or Monthly = 2.
Hours	Bitmap of hours that the job is scheduled. (Starting at 3am = 2(bit1), 6am = 4(bit2), 9am = 8 (bit 3) etc)
Days	Bitmap of days that the job is scheduled. (Starting at Sunday = 2, Monday = 4, Tuesday = 8 etc)
Months	Bitmap of months that the job is scheduled. (Starting at January = 2, February = 4, March = 8 etc)
MonthDay	Day of the month that the job is scheduled
StartDate	The start date for the standing order to be checked
ExpiryDate	The last date for the standing order to be checked

12.4.5 System Messages

The System Messages entity is a very simple approach to providing messages to users of the system when certain events have occurred. For example, a message might be generated when a laboratory analysis is completed. The message will appear on the user interface until it is cleared.

Property	Description	Sourced From	Modified by
Cleared	Property that can be set to true by the application to clear the message	t_SystemMessages	App via Data layer
ID	A unique identifier for the message	t_SystemMessages	App via Data layer
Msg	The Message	t_SystemMessages	App via Data layer
Generated	The data and time the message was generated	t_SystemMessages	App via Data layer

12.4.6 Tracker Items

A tracker item is an entity that describes the properties of an element of the wine tracker display. The data for these items is generated by the business layer of the application and dumped into the *t_TrackerDump* table in the database. Each entry is then used to instantiate a tracker item to allow the tracker screen to be built. The same table is used by all users, so the table also has a UserID field to distinguish between data items for users who may be using the tracker function at the same time.

Property	Description
ID	Unique identifier for the item
ObjectID	The ID of the object that the item represents
ObjectType	The type of object that the item represents. Either Wine, Fruit, or CreatedWineFruit.
TargetObject	The ID of the item that this item is linked to.
WNID	The ID of the worknote that created the link between the two items
Layer	The row that the item should be displayed on the screen
WNTarget	The id of the target object in the worknote cited above
WNCompletedDate	The date and time the worknote was completed
wnheading	The Heading for the worknote

12.5 Facilities

Each facility type is loaded from the *t_Facilities* table into the appropriate facility collection structure of the enterprise object. This means that each facility has a unique ID rather than the ID being unique only amongst similar facility types, thus avoiding the need to have to keep a facility type parameter whenever the data model refers to a specific facility.

12.5.1 Diary Entries

Diary entries are facility and user specific and not shared. Entries are stored in the *t_Diary* table. The object details are displayed here to avoid repetition, but an instance of this collection appears in each of the main facility types

ID	A unique identifier for the object
User	Identity of the creator if the diary entry
ForDate	The date the diary entry applies to
DiaryNote	The text of the diary entry
StartTime	The start time of the diary entry
EndTime	The end time if the diary entry
FacilityID	The ID of the facility that the diary entry appears in
DateCreated	The date and time the diary entry was created.

12.5.2 Wineries

The Winery object (Figure 20) is the most complicated facility object because it has the most object types hanging off it. However the hierarchy of these objects provides a convenient means of grouping them into logical concepts in order to explain their structure and function.

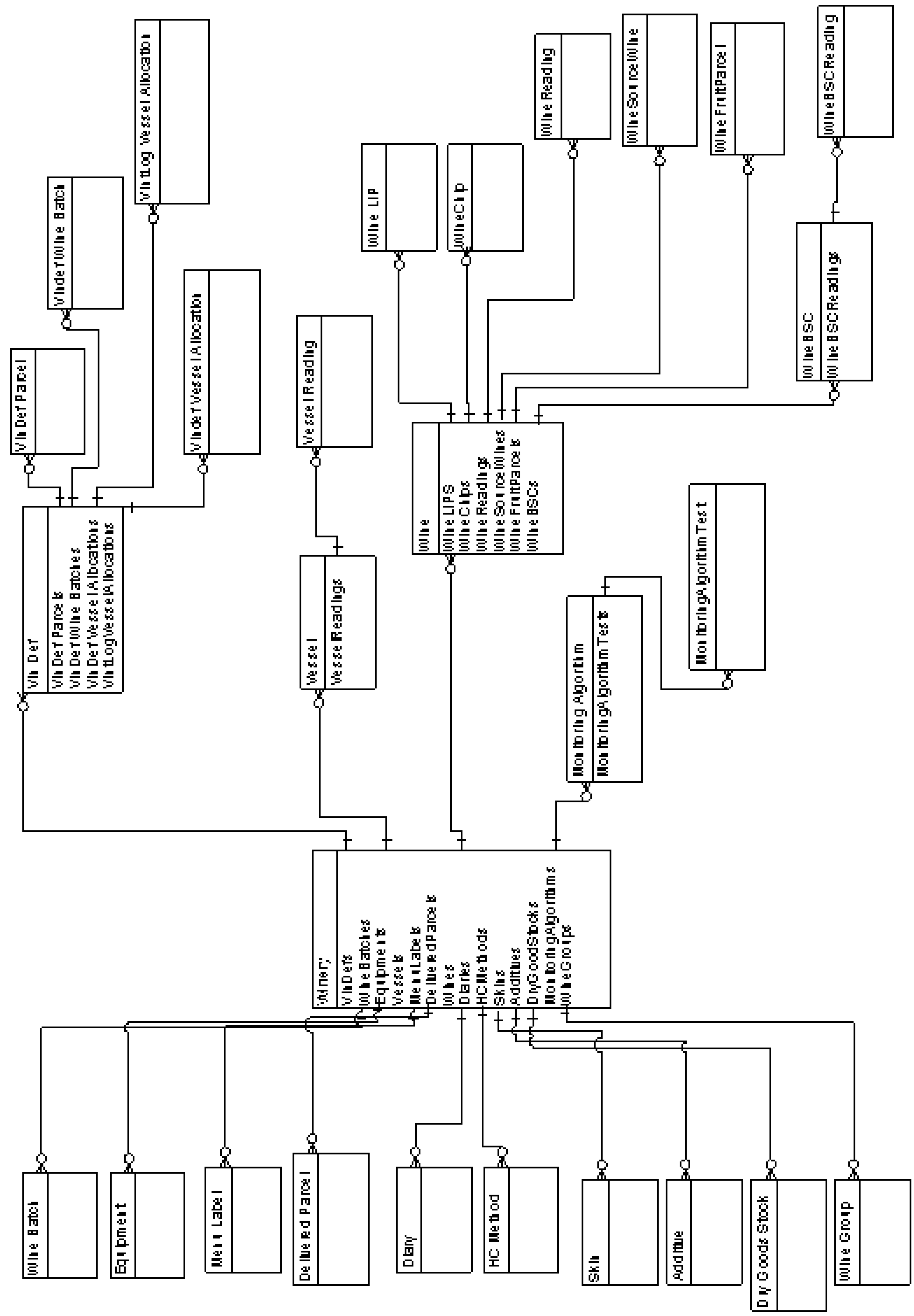


Figure 20. Winery Object Diagram

The Wineries collection, as part of the Enterprise object, contains several methods used to help search the winery object for specific entities as well as manipulate wine names.

FindVinDef	Method to find a specific Vinification Definition object
FindWine	Method to find a specific wine object
FindWineFromBSC	Method to find a specific wine object from the ID of a Barrel Set Component
FindVesselIDFromBSC	Method to find a specific vessel object from the ID of a Barrel Set Component
FindVessel	Method used to find a vessel object from its ID
FindAdditive	Method used to find a additive object from its ID
FindWineReading	Method used to find a wine reading object from its ID
FindVesselReading	Method used to find a vessel reading object from its ID
WineryIDForWine	Method used to determine the ID of the winery object for a given wine ID
GenerateCombinedWineName	This method derives the wine name when two wines are combined, based on the constituent fruit varieties, regions and vintages.
GenerateProcessingWineName	This method generates a default name for a wine at the fruit reception processing stage based on the vineyard block detail
GenerateNextWineName	This method uses a simple algorithm to attempt to logically increment a wine name.

The winery object consists of only a few properties and methods. The table *t_Facilities* is used to store this information

Sequence	The ordinal position that this object should be displayed.
Managed	Flag indicating whether this winery is managed or not. This has a major effect on how the winery will be displayed
MainView	Flag indicating if the winery should be displayed on the main menu of the application
Image	Filename of the image used on the main menu when the winery is displayed
SubtleImage	Filename of the image of the winery used in the background.

ID	Unique identifier of the winery
WineryName	The name of the winery
VintagePlanImage VintageLogisticsImage WeighbridgeImage WorkDiaryImage MonitoringImage	Filename of the background image to be displayed on the various predefined screens.
Code	A short code or abbreviation for the winery. Generally used in wine names or other objects that belong to the winery.
GetParcelsIDsByHarvestDate	Method used to retrieve all the parcels defined for the winery by harvest date. This calls a special function in the data layer rather than building the list using the vineyard objects, mainly as a means of speeding up the process. See *15.4.6 The Data Model Language*
FindVinDefVesselAllocation	Method used to find a vinification definition vessel allocation object given its ID
FindVintLogVesselAllocation	Method used to find a Vintage Logistics vessel allocation object given its ID

The collections in each winery object can be grouped under general headings.

12.5.2.1 Vintage Planning & Logistics

The structure and behaviour of the Vintage Planning and Vintage Logistics objects are carefully designed to help the user interface achieve its design goals, just as much as they are concerned with enabling the population of the database with relevant data. The nature of vintage planning is heavily focussed on manifesting conceptual and abstract objects in the minds-eye of the user, and presenting them on the screen. These abstract objects are nevertheless well understood in the mind of the winemaker, although they will differ in granularity and detail from one winemaker to another, and will not necessarily even be known by the terms used here. This is the only design area of the data model that recognises the user interface as a major influence, even though the interface is designed around these abstract objects mentioned earlier. The source tables for these structures all have the "*t_Plan*" prefix.

12.5.2.1.1 VinDefs

The VinDef (Vinification Definition) object is the cornerstone of the Vintage Planning screen. It links fruit parcels to fermentations and defines parameters for getting from the former to the latter. It also provides seeding information for the fruit reception jobs created when the corresponding fruit parcel comes into the winery.

The vindefs collection contains only one significant method, apart from the usual collection methods. “PlanVesselTimeFree” is used to compute whether a certain fermentation window is available for a vessel.

The VinDef object is sourced from the *t_Plan_VinificationDefs* table and contains several properties, two methods and four important collections

VinDefName	The name of the VinDef
Vintage	The nominated vintage that the vindef belongs to.
WineryID	The ID of the winery that the vindef belongs to.
FermentOnSkins	Flag indicating that the fruit should be fermented on skins rather than being pressed of during processing
InitialProcessing	This is a string of colon separated IDs that relate to predefined actions in sequence to describe the initial processing of the fruit.
Description	A brief text field for a description of the vinification definition
Yield	The expected yield of juice/wine as a percentage of the fruit in litres per kilogram
PressRegime	The ID of the pressing regime containing parameters to be used during the fruit processing
Split	Flag indicating that the processing job pressing phase should be split between free-run and pressings
Yeast	The id of the yeast that will be used for all of the ferments
FermentationLength	The number of days the fermentation is expected to take
FruitSummary	A summary of the fruit attached to the vindef
TotalTonnes	The total tonnes of fruit attached to the vindef
IsRed	A method used to work out whether the fruit has red skins

The VinDefParcels collection is simply an array of IDs of the parcels that belong to the vindef. This array is built from the *t_Plan_ParcelToVinDef* table. The parcels referred to are the Plan parcels attached to the vineyard objects (*see 12.5.3.2.1 Parcels*)

12.5.2.1.2 VinDef Wine Batches

The VinDef winebatches collection contains information about the wine batches that the vindef is supplying . This is sourced from the *t_Plan_VinDefToWineBatch* table.

WBID	The ID of the Wine Batch object
Proportion	The proportion of the vindef that is being apportioned to the wine batch
ID	The ID of the vindef wine batch object

The collection also has a method, “DistributeProportions” which is used to reset proportions to wine batches when the number of wine batches linked has changed.

12.5.2.1.3 Vindef Vessel Allocations

The vessel allocations objects are used to define the vessels to be used for the ferments defined in the vindef. The data is sourced from the *t_Plan_VinDefVesselTypeAllocations* table.

VinDefID	A back reference to the VinDef to which the allocation belongs
VesselType	The ID of the vessel type nominated for the vessel allocation
Amount	The number of tonnes allocated
ID	A unique identifier for the vessel allocation object
Multiple	The number of vessels of the vessel type required by the vindef
Priority	The order of presentation of the allocations

12.5.2.1.4 Vintage Logistics Vessel Allocations

The Vintage Logistics Vessel Allocations are more specific bookings of individual vessels for vindefs. These are related to the Vintage Logistics screen. The data is sourced from the *t_Plan_VintageLogisticsVesselAllocations* table.

The collection object has a method “Rebuild” which is used each time the vintage logistics screen is loaded to make sure that any vindef changes are reflected in these allocations.

The vintage logistics vessel allocation object is linked to the vindef vessel allocation, thereby identifying itself as one of the vessel allocations to fulfil the vindef.

ID	Unique identifier for the object
vdvaID	The ID of the VinDef Vessel Allocation object
VinDefID	Back reference to parent VinDef
Vessel	ID of the vessel for this allocation
StartDate	The expected start of the fermentation for this allocation
FermLength	The number of days that the fermentation is expected to take

Multiple	Position in display order of this allocation in the collection
Amount	Number of Tonnes allocated for this vessel
IsCommitted	Flag indicating whether the vessel allocation has been done.

12.5.2.2 Wine Batches

Wine batch objects are conceptual groupings of wines based on the users design. Wine batches themselves have no bearing on any jobs generated in the winery, unlike the vindefs which seed the fruit reception jobs. They do contribute to the estimates for oak requirements

Description	Text description of the wine batch
WineryID	Back reference to the ID of the parent winery
Vintage	The current vintage as specified in Global Parameters
BatchName	Text name of the wine batch
Vessels	Colon separated list of IDs of vessel types selected for the maturation phase for this wine batch
MaturationLength	Number of months expected for maturation process
OakPercentage	Percentage of wine volume to go to oak
OakNewPercentage	Percentage of wine going to oak that goes to new oak
TotalVolume	Total volume in litres of the wine batch
WineSummary	This property generates a summary of the wine in terms of volume and percentage of the various varieties of fruit and regions.

Each wine batch has a collection of Wine Batch Vindef Ids which is simply a collection of IDs of the Vindefs that are linked to this Wine batch. They are sourced from the *t_Plan_WineBatches* table.

12.5.2.3 Vessels

Each vessel object describes an individual vessel defined in the winery. Barrel Sets are special kinds of vessels in that they are conceptual groups of vessels, but need to appear in the same context as actual vessels because they are de-facto vessels themselves. The great variation in the types and properties of vessels means that this entity is quite broad in nature. The prospect of new vessel types being developed means that the entity does not contain any business rules for excluding properties based on the settings of others. For example, the automated heating and cooling properties are not turned off for non tank farm vessels because it is conceivable that such a vessel might be available in tfhe future. Properties such as the status and wineid are derived from wine readings rather than directly from database records for the vessel, in the *t_TankMap* table. Most of the other properties are set based on the vessel type or category.

ID	Unique Identifier for the vessel
Icon	Base of the filename of the icon used to represent the vessel on the screen
StatusID	ID of the status of the vessel. This may be set to the state of the vessel if there is no wine in it, or be based on the wine.
State	The ID of the state of the vessel. Only valid if the vessel is empty.
WineID	The ID of the wine in the vessel.
AutomatedHeating	Flag indicating whether the vessel is heated automatically by sensors and computer controlled elements.
AutomatedCooling	Flag indicating whether the vessel is cooled automatically by sensors and computer controlled cooling jackets
ManualHeating	Flag indicating that the vessel can be heated by manually controlled heating elements
ManualCooling	Flag indicating that the vessel can be cooled by manually controlled cooling jackets
SortOrder	Order that the vessel should appear in on screen display list
Maker	ID of the manufacturer of the vessel. Mainly specified for Oak vessels.
YearMade	The year of manufacture of the vessel. Mainly specified for Oak vessels.
Toast	Text description of the toast level of the vessel. This is not a codeset value because it makes no sense to have a codeset for a value that is not standardised. A Sequin Moreau barrel with medium toast level cannot be compared to that of another manufacturer even though they may use the same standard phrases for toast level. In addition, toast levels descriptions are continually changing as manufacturers become more descriptive, so maintaining a codeset becomes problematic.
Material	Text description of the vessel material. Mainly specified for Oak vessels.
Loading	Flag used by the object to stop recursive parameter setting during the instantiation phase of the entity
BSMessage	Text generated if the vessel is a barrel set and there are issues or discrepancies concerning volume parameters.
DaysUsed	Calculated from vessel readings, this is the number of days that the

	vessel has had wine in it. Mainly used for oak vessels
BarrelUsedForRed	Flag computed from readings to indicate whether the vessel has ever been used for a red wine
BarrelSetString	Text list of all the individual vessels, if the object is a barrel set
BarrelMakeUpString	Text list of the percentages of makers that make up the barrel set
CapacityLitres	Returns the volume capacity of the vessel which may be sourced from the vessel type if the object is an individual vessel , or calculated from the list of vessels if the object is a barrel set.
CapacityTonnes	The tonnage capacity of the vessel, calculated from the vessel types of the constituent vessels if a barrel set. Any vessels not flagged as available for on skins wine or juice do not contribute to the total.
ProportionOak	The proportion of the vessel or barrel set that is made of oak.
UllageAllowance	The volume of wine or juice in litres under the nominated capacity of the vessel that indicates that the vessel is regarded as being "on ullage"
AnyWorknotes	Flag indicating that there is an active worknote that somehow involves this vessel object. This function generally knows if the vessel is involved because the work note type indicates whether the target of the worknote is a vessel
BarrelSetNeedsAttention	Flag indicating that there is a BSMessage currently applicable to the barrel set.
CurrentTonnage	Calculates the current amount of fruit currently in the vessel. This is only for wines resulting from a fruit processing job.
Skin	This property is computed by looping through the skins collection in the winery object and identifying any entities associated with this vessel.

Whenever the following properties are set, they are also recorded and time-stamped as readings. This allows the model to keep an historic record of these vessel specific properties.

Tankfarm	Flag indicating whether the vessel should be displayed and treated as a Tank Farm vessel (immovable) or Barrel Hall vessel
WineryID	Back reference to the ID of the winery that this vessel belongs to
TypeID	Reference to the Global Parameters Vessel Type collection of the ID if the vesseltype
CategoryID	Reference to the Global Parameters Vessel Category collection of the ID if the

	vessel category
VesselName	The name of the vessel
HeatingOn	Text of heating setting. This can be a numeric or text value.
CoolingOn	Text of cooling setting. This can be a numeric or text value.
MOX	Flag indicating the presence of a micro-oxygenation scinter in the vessel.
GasBlanket	Flag indicating the presence of a nitrogen diffuser in the vessel.
Chips	Flag indicating the presence of chips in the vessel
BarrelSet	ID of the Barrel set to which this vessel belongs
XPos	X-axis coordinates for on-screen display
YPos	Y-axis coordinates for on-screen display
Unused	Flag used to indicate whether the barrel set is regarded as an "Un-used" set which is special category used to keep un-used vessels together but exclude the barrel set from being selected for use in operations within the winery.
BSNotes	Text notes for the barrel set
UllageInBarrelSet	Flag indicating that this vessel is the nominated vessel in the set that is on ullage.

There are three methods associated with the vessel entity.

AddVesselReading	This adds a vessel reading to the vessel's reading collection
SetStatus	This function calculates and sets the statusID of the vessel based on the state of the vessel and whether or not it contains wine.
SetState	This function sets the state of the vessel, which is only used when the vessel does not contain wine.

12.5.2.4 Vessel Readings

The vessel reading object allows an historic record of various parameters to be maintained and referenced when appropriate. The data is stored in the *t_Vessel_Readings* table.

Vintage	The vintage setting when the reading was generated
WorknoteID	The id of the worknote (if any) that caused the generation of this vessel reading
ReadingDateTime	The date-time stamp of the reading
ReadingValue	Value of the reading. This is stored as text to allow any form of data

ReadingType	System defined constant for the type of parameter
ID	Unique identifier of the reading
VesselID	Back reference to the vessel to which this reading applies

12.5.2.5 Wines

The wine object is central to the data model. As soon as a parcel of fruit is processed it becomes a wine. All wines are constantly accounted for throughout the instance of the model, even when the wine has been bottled. The Wines collection has one significant method called “CombineWines” which manages the common occurrence of two wines being combined to make one. Apart from this it is like any other collection. The wine object contains several collections as well as properties and methods. Most wine parameters are stored in the Readings objects, but more elaborate structures are required for other wine features, and these are described as well. The *t_Wines* table only keeps the first 9 parameters (ID to Vintage Claim)

Properties

ID	Unique identifier for the wine
WineryID	Back reference to the ID of the winery to which the wine belongs
OnSkins	Flag indicating whether the wine is still in skins, either from the original source fruit or added
Lees	Flag indicating that the wine is lees. This is still regarded as wine by the AWBC and still contributes to the L.I.P of any wine to which it is added
MonitoringAlgorithm	The id of the monitoring algorithm that is applied to this wine. Refers to *12.5.2.10 Monitoring Algorithms*
WineGroup	The id of the wine group that this wine belongs to. Refers to *12.5.2.11.8 Wine Groups*
RegionalClaim	The id of the regional claim group that this wine belongs to. Refers to *12.5.2.11.8 Wine Groups*
VarietalClaim	The id of the varietal claim group that this wine belongs to. Refers to *12.5.2.11.8 Wine Groups*
VintageClaim	The id of the vintage claim group that this wine belongs to. Refers to *12.5.2.11.8 Wine Groups*
MonitorScoreDescription	When the Monitor Score method is called, a text description of the score is generated at the same time and the result is stored here for

	reference.
Skins	This property is computed by looping through the skins collection in the winery object and identifying any entities associated with this vessel.

Methods

CopyGroupsForOtherWine	This copies the wine group settings from another wine to this wine.
LastActionDateTime	This method computes the date time stamp for the last action performed on the wine. This is important for ensuring that worknotes are completed in sequence.
FindWineBSCReading	This function retrieves a barrel set component reading from the barrel set collection given its id.
MonitorScore	This function applies the nominated monitoring algorithm to the wine and establishes the current score for the wine.
GetWineHarvestDate	This function loops through the delivered fruit parcels for this wine and determines the earliest harvest date
GetMass	This function computes the mass of fruit used to create this wine, using the delivered parcels of the wine and all its source wines.
TempWineNameRoot	Generates a wine name for the wine based on the source fruit.
SetWineStatus	Adds a wine reading to the wine that sets the status of the wine and attributes it to a work note.
SetWineSubStatus	Sub status values are a bitmap of flags for various hard-coded sub statuses. This function adds a wine reading to the wine with this value
WineComment	Text field for user comments for the wine.
AnyWorknotes	Computes whether there are any active worknotes for the wine
AnyRatifications	Computes whether there are any unratified readings associated with this wine.
GotChips	Computes whether there are currently any chips remaining in the wine.
SourceWineString	Generates a text string of the source wines names that constitute this wine.
IsWineASource	Computes whether a specific wine is a source wine of this wine.

The wine object also contains collections for LIP, Chips, readings, Source Wines, Fruit parcels and Barrel Set components, described next.

12.5.2.5.1 LIP

The structures for storing LIP information include the volumes of wine derived from specific fruit sources. As a result, the LIP list is used to determine the actual volume of wine as well as the LIP information regarding proportions of the wine from vintage, region, and variety perspectives. The data is stored in the *t_Wines_Readings* table.

The LIP collection contains several procedures

LatestLIPDate	This computes the date of the last worknote applied to the wine, and hence the date of the latest LIP records
Fortify	This is a method that allows the addition of material to a wine that adds volume to the wine. The choice of the word “fortify” is because it was first thought that only the addition of ethanol to fortify a wine would fit into this situation. This adds an LIP record to the wine with blockid = 0 which implies ethanol
GetVolume	Calculates the volume of the wine based on the LIP records
SetNewVolume	Resets the volume of the wine by proportionally tweaking the volume properties of the LIP records for the wine
BuildLIPString	This generates text in several optional formats that describes the LIP information for the wine.
LIPArrays	This generates arrays of proportions of the wine for Variety, Vintage, and Region
DuplicateForOtherWine	This method duplicates the LIP information of this wine into another wine.

Each LIP record details the volume of an individual part of the wine with specific vintage and block information, and hence regional and variety information, for a given worknote. This means we keep historic and current information concerning the LIP information for each wine. Each set of records are identified by the worknote ID. The data is stored in the *t_Wines_LIP* table.

ID	A unique identifier for the LIP record
WineID	Back reference to the wine

Vintage	The vintage for the LIP record
BlockID	The id of the block, from which we derive the region and variety of the source fruit
Volume	In litres
WNID	The worknote id that created this record

12.5.2.5.2 Readings

Wine readings are used to store a range of parameters associated with the wine. These include laboratory readings as well as other significant properties and settings not covered by the other collections associated with the wine entity. The readings collection includes...

LatestReadingDate	This retrieves the date of the latest reading for the wine
BuildSubStatusString	The substatus of a wine is a bitmap of various hard-coded flags. This function translates values into text
DuplicateCurrentLabReadingsForOtherWine	This function copies the current readings for this wine to new records in another
SetCurrentLabReadingsAsAssumed	This method duplicates all the current readings for the wine and sets them to suspect; usually because the wine has had some process applied to it.
AcidAdded	This sets any current readings affected by the addition of acid to be flagged as suspect
SO2Added	This sets any current readings affected by the addition of SO_2 to be flagged as suspect
ConcAdded	This sets any current readings affected by the addition of grape concentrate to be flagged as suspect

Each reading entity is stored in *t_Wines_Readings* the table and consists of the following properties

ID	A unique identifier for the reading
Vintage	The vintage year during which the reading was generated
WorknoteID	The ID of the worknote that caused the reading
ReadingDateTime	The date and time stamp for the reading
ReadingValue	The value of the reading. This can be a number or text

ReadingType	The type of the reading. This value comes from a list of constants defined in the system
WineID	Back reference to the wine that this reading belongs to.
Assumed	Flag indicating that the reading is assumed but has not been taken directly from the wine as it existed at the time
Suspect	Flag indicating that the reading is not reliable because some event occurred to the wine after the reading was taken
AwaitingRatification	Some readings are not regarded as official until they have been ratified by an appropriately authorised user

It should be noted that the term “Readings” might be misleading in that these objects and this structure is used to record settings applied by the user to the wine as well as observed readings read from the wine. For example, the name of a wine is stored as a reading, but the name is really a setting. Perhaps the term “Reading“should be regarded as an abbreviation of “Reading-Setting”.

12.5.2.6 Source Wines

The source wines entities are simply structures that contain references to other wines that have been sources for this wine. They enable the tracking and ancestry of a wine to be computed. The information is sourced from the *t_Wines_SourceWines* table.

ID	A unique identifier for the record
WineID	Back reference to the wine object that this record belongs
SourceWineID	The ID of the wine object that is a source of material for this wine
WNID	The ID of the worknote that caused the source wine to be part of this wine
Volume	The volume of source wine that is regarded part of this wine

12.5.2.7 Fruit Parcels

The fruit parcels collection is a simple list of plan fruit parcels in the winery object that are part of this wine and the amount in tonnes of that parcel in the wine. This is sourced from the *t_Job_FruitReception* and *t_Job_FruitReception_Allocations* tables.

ParcelID	The ID of the fruit parcel as defined in the vintage plan
Tonnes	The mass of the parcel that went into this wine.

12.5.2.8 Chips

The chips collection object has only one major method; "ChipString" which generates a text description of the type and amount of chips currently in the wine. The chip entity is a description of an individual addition of chips to the wine. It also includes the removal from the wine so that an historical record is kept of when any chips treatments were applied and for how long. The nature of addition of chips is that it works much faster than storage in oak but not so fast that the effect of the addition is not time independent. This is another example of data that could be derived from the jobs and worknotes but is far more efficiently stored in its own right in the *t_Wine_Chip* table.

ID	Unique identifier for this instance of a chip addition
WineID	Back reference to the wine entity
ChipsID	ID of the chips; references the additives collection for details.
Amount	Mass in grams of the chips added
InWNID	The ID of the addition worknote
OutWNID	The ID of the chips removal worknote

12.5.2.9 Barrel Set components

The creation of Barrel set component entities is necessary when individual parts of a barrel set need their own identity for the sake of laboratory analysis. Even if a wine in a barrel set is regarded as a single object from the model's perspective, on some occasions it is necessary to analyse individual barrels because they are physically separate and can have different behaviours and parameters. Whenever a wine needs to be broken down for separate analysis, the "GetOrCreateBSCForVesselID" method is used to generate the BSC list; each element consists of the following properties...

ID	Unique identifier for the Barrel Set Component
VesselID	The ID of the individual vessel
WineID	Back reference to the wine

Each component has a collection of readings which have the same structure as the wine readings collection and objects. The barrel sets are stored in the *t_Wines_BSC* table, and the readings in the *t_Wines_BSC_Readings* table.

12.5.2.10 Monitoring Algorithms and Tests

It was decided that Monitoring Algorithms should be winery based rather than enterprise based, because each winemaker will have different criteria for judging a wine's condition. Comparison of scores of wines at different wineries serves less purpose than allowing each winery to maintain its own set of algorithms. The collection and the algorithm entity are unremarkable because the only major parameters they contain are the name of the algorithm and a collection of Monitoring Algorithm Tests. These objects have the following structure

ID	Unique identifier for the test
AlgorithmID	Back reference to the Monitoring Algorithm
Parameter	The wine reading type that this test is applied to
TestType	One of these options... • days since last analysis after • ppm over • ppm under
TestPoints	The number of points added
TestThreshold	The number of units required before points are added
TestIncrement	The number of units of the parameter that adds points to the score.

Each test follows a simple formula of adding a certain number of points (TestPoints) to the score for the wine each time the selected parameter (Parameter) crosses by a specified amount (TestIncrement), certain criteria (TestType) beyond a certain level (TestThreshold) . This is explained in *13.3.7 Monitoring and Algorithms*. The data is stored in *t_MonitoringAlgorithms* and *t_MonitoringAlgorithmTests* tables.

12.5.2.11 Other Objects

The remaining collections are also for objects that are winery specific and need to be defined as such because they might involve individual preferences of the decision makers at the winery, or geographic location might preclude the use of all the objects at all the defined wineries.

12.5.2.11.1 Skins

The skins collection is part of the winery object, but is also used to manifest properties within wine, vessel, and equipment objects when they are associated with that object type. In other words, a skin object may represent a stand-alone body of skins left in a vessel, in a press, or still in a wine. Skins are only defined as an entity when they appear in their own rights rather than being inherently part of an un-pressed ferment. The data is stored in the *t_Skins* table.

ID	Unique identifier for the Skin Object
Description	The description of a skin is the name of the wine(s) from which it came when pressed. However, when it is added to another wine with skins, it does not assume the name of that wine, but rather the two skins are combined into another object and the description becomes the names of the two source skins combined.
PressWN	The ID of the pressing worknote that created this skin
PressID	The ID of the press containing this skin
VesselID	The ID of the Vessel containing this skin
WineID	The ID of the wine containing this skin
Status	The status of the skin (In wine, press, vessel, or dumped)
WineryID	Back reference to the winery

12.5.2.11.2 Delivered Parcels

Delivered Parcel are differentiated from Plan parcels in that they represent actual parcels of fruit delivered to the winery as opposed to the parcels in the vintage plan which are divided into allocations by the fruit receival job allocations. This table and associated entities in the model are therefore a summary of the parcel because the same information could be derived from the allocations.

The collection has a procedure “GetParcelDeliveredTonnes” which calculates the number of tonnes of a plan parcel that have actually been delivered.

The delivered parcel data is sourced from the *t_WineryDeliveredParcels* table and the object has the following structure...

ID	Unique Identifier for the object
WineryID	Back reference to the winery
Vintage	The vintage that the parcel is part of.
PlanParcel	ID of the plan parcel that generated the Reception Job, that generated this delivered parcel.
HarvestDate	The date the parcel was harvested
HarvestType	The method used to harvest the fruit. Hand or Machine.

BlockID	The ID of the source vineyard block
Owner	ID of the owner of the fruit
Tonnes	Total tonnage of the parcel
Docket	Weighbridge Docket. Optional field in the event that a weighbridge docket is available
ProcessingJobId	ID of the Fruit Reception Processing Job

12.5.2.11.3 Equipment

The equipment objects are used to store information about common information about equipment in the winery. There are some Press specific features of these entities such as the "GetPressIndexForRegimeID" procedure in the collection that scans the list for press objects that have a matching regime. All equipment objects are populated with data from the *t_Equipment* table. The equipment object has this structure...

ID	Unique identifier for the object
TypeID	Equipment type from system constant list
Description	Text description of the equipment
Icon	Filename for the icon that represents the equipment on the screen
WineryID	Back reference to the Winery

If the object is a press, the PressRegimes collection is populated with PressRegime objects which describe predefined pressing processes, which may or may not be automated. This data is osurced form the *t_CS_PressingRegimes* table

ID	Unique identifier of the press regime object
RegimeName	Text name of the regime
Description	Simple text description and/or instruction for the regime

12.5.2.11.4 Additives and Stock

Additive is a general term for any material that can be added to a wine. They are divided into 8 different types according to their function.

- Yeast
- Chips

- Acids
- Sulphur Dioxide
- Nutrients
- Fining Agents
- Malolactic Fermentation Bacteria
- Other

The additives collection is used to define the types of additives available in the winery.

ID	Unique identifier for the additive
AddType	The additive type that it belongs to
Description	Text description of the additive
Restriction	Text of additive restrictions as per manufacturer advice.
Rate	Default rate of addition
RateFactor	Units of rate
Trialable	Flag indicating whether it is possible to run Addition Trials for this additive
ByVolume	Flag indicating whether units of addition are by volume or alternatively by mass
AffectspH	Does the additive affect the pH of the wine?
AffectsTA	Does the additive affect the Titratable Acidity of the wine?
AffectsSO2	Does the additive affect the Free or Total SO_2 content of the wine?
AffectsSugars	Does the additive affect the reducible sugar level of the wine?
AffectsVolume	Will the addition affect the overall volume of the wine. Currently only valid for ethanol addition
Sulphites	Does the additive contain sulphites?
Egg	Does the additive contain egg white?
Milk	Does the additive contain milk?
Nuts	Does the additive contain nuts?
Casein	Does the additive contain Casein?
Default	Is this the default additive for its type?

The additives stock objects define the actual stock and batches of the winery's current additives.

ID	Unique identifier for the batch of stock
AdditiveType	The type of additive

AdditiveID	The ID of additive referenced from the additives collection
Batch	Text of batch number or description
Cost	Unit cost of the additive
Complete	Flag indicating that the batch has been exhausted

The data for these objects are sourced from the *t_CS_Additives* and *t_AdditivesStored* tables.

12.5.2.11.5 Menu Labels

Each managed facility defined in the system has a main menu on the interface as well as a set of nodes in the tree menu on the left of the main application window. As each facility is loaded during the initial startup, key menu elements are added and recorded in these objects to link these menus directly to the nodes in the tree. This means that when a item in the facility menu is clicked, it is as if the node in the tree that it is linked to has been clicked.

Strictly speaking, these objects are not really part of the model but rather an aide to the user interface to help it relate the menu features to the tree menu. They are loaded from the *t_GUI_MenuLabels* table.

ID	Unique identifier for the menu item
FacilityID	The ID of the facility which this element belongs to.
NodeType	Type of node for the facility
Label	Text of the menu item
LinksToNode	The node in the tree menu that this item links to
LabelLeft	X-axis positioning of the menu item on the menu window
LabelTop	Y-axis positioning of the menu item on the menu window

12.5.2.11.6 Heating and Cooling Methods

Heating and cooling methods are winery specific, subject to the preferences of the winemaker and the availability of the appropriate equipment. The diversity of means of heating and cooling makes it necessary to have these loaded dynamically from the database (table *t_CS_HeatCoolMethods*) rather than hard-coded. It should be noted that this list excludes methods that utilise vessel features such as heating or cooling plates.

ID	Unique identifier for the method
Description	Text description of the method

Heat	Flag indicating that it can be used to heat a wine
Cool	Flag indicating that it can be used to cool a wine

12.5.2.11.7 Dry Goods Stock

Stock of dry goods is attached to the winery rather than the bottling facility because the stock tends to be controlled and stored in the winery rather than by the bottling facility. Like the Additives Stock objects, these objects refer to the dry goods objects which define the goods that are available for the enterprise. Dry goods types are stored in separate database tables instead of being defined as a type within the same table. This is because dry goods have different features and properties. The goods stock data is stored in the *t_DryGoodsStored* table.

ID	Unique identifier for the dry good batch
WineryID	ID of the winery where these goods are available
DryGoodType	The type of good, which determines which collection of dry goods objects it refers to
DryGoodID	The ID of the item in the appropriate dry good collection
Batch	Text description of the batch number or description
Cost	The unit cost of the dry good
Complete	Flag indicating that the batch is exhausted.

12.5.2.11.8 Wine Groups

These objects are used as reference objects for wine groups, regional claims, varietal claims, and vintage claims. The data is stored in the *t_WineGroups* table.

ID	Unique identifier for the group
WineryID	Back reference to the Winery
GroupName	Text name of group
Category	System Constant of type of group (General, region, variety, or vintage)

Each entity also has two functions...

InUse	Computes whether the group is referenced by a wine
UsedBy	Computes which wines are referencing this group

12.5.3 Vineyards

The Vineyards entities (Figure 21) are simpler to model because there tends to be less variation in the year to year activities taking place, and the entities and objects tend to be more consistent, real-world, and static in nature.

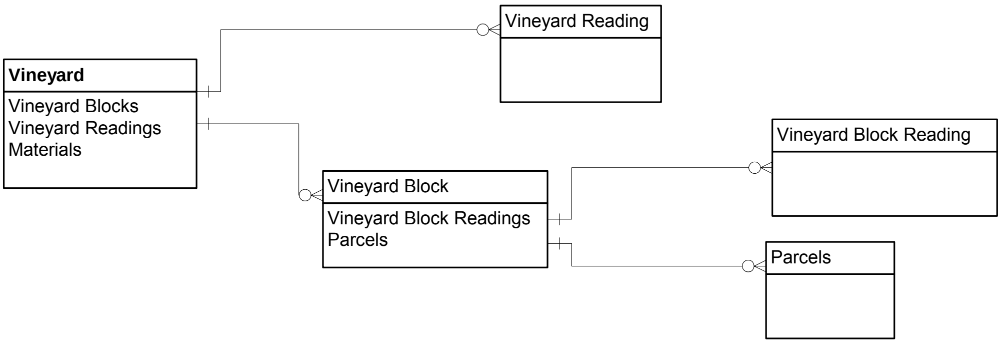


Figure 21. Vineyard Object Diagram

The vineyards collection is a collection of vineyard objects as well as having the following functions which operate on the collections in each of the vineyard objects

FindParcel	Searches Parcels collection in each vineyard object to get the parcel object required.
GetVineyardBlockName	Searches vineyard blocks collection in each vineyard object to get the block name required.
GetVineyardBlock	Searches vineyard blocks collection in each vineyard object to get the block object required.
GetVineyardForBlock	Searches vineyard blocks collection in each vineyard object to find the block name and return the parent vineyard object.

The vineyard object has the following structure sourced from the *t_Facilities* table.

ID	Unique identifier of the vineyard
VineyardName	The name of the vineyard
Code	A short code or abbreviation for the vineyard. Generally used in parcel names or other objects that belong to the vineyard.
Region	Text of the region that the vineyard is in.
RegionID	ID of the region object from Global Parameters

State	Text of the state that the vineyard is in
Country	Text of the country that the vineyard is in
Managed	Flag indicating whether the vineyard is managed
MainView	Flag indicating if the vineyard should be displayed on the main menu of the application
Image	Filename of the image used on the main menu when the vineyard is displayed
SubtleImage	Filename of the image of the vineyard used in the background of some dialogs.
Sequence	The ordinal position that this object should be displayed.
WorkDiaryImage	Filename of the background image to be displayed on the various predefined screens.

12.5.3.1 Vineyard Readings

Each vineyard has a collection of reading entities consisting of the following properties...

ID	A unique identifier for the reading
VineyardID	Back reference to the vineyard that this reading belongs to.
ReadingType	The type of the reading. This value comes from a list of constants defined in the system
ReadingValue	The value of the reading. This can be a number or text
ReadingDateTime	The date and time stamp for the reading
WorknoteID	The ID of the worknote that caused the reading

This data is sourced from the *t_VineyardReadings* table.

12.5.3.2 Vineyard Blocks

The vineyard blocks collection contains a list of vineyard block objects. The data is stored in the *t_VineyardBlocks* table.

ID	Unique identifier for the vineyard block
BlockName	Text of the name of the block
MenuX	X Coordinates for the main vineyard menu for the label for this vineyard block
MenuY	Y Coordinates for the main vineyard menu for the label for this vineyard block
Variety	Text of the grape variety of the block
VarietyID	ID of the grape variety for the block
VarietyCode	Letter code for the variety of the block

Area	The area in hectares of the block
Clone	Text of the clone of the variety for the block
Planted	The year the block was planted
Rows	The number of rows in the block
Vines	The number of vines in the block
Selected	Flag used to indicate that the block has been selected on the user interface
Colour	“R”ed or “W”hite
InitialParcelCheck	Creates a default first parcel if there are none defined and a yield and harvest date have been defined.
PreviousWineString	Computes a text list of the wines (bottled or current) that have used this vineyard block fruit

12.5.3.2.1 Parcels

The parcels collection in each vineyard block contains the proposed parcels defined for the vintage. The collection contains several properties

Unaccounted	Calculated from expected yields and parcels for the vineyard block, this is the amount of fruit of the estimated yield that has not been assigned to a parcel. This may be negative indicating that the planned parcels are in excess of the estimated yield.
RemCount	The number of parcels defined as “Remainder”. Can only allow one parcel to be defined as remainder so this acts as a trigger for an error message.
ExpectedBlockYield	The expected yields for the block for the current vintage
GetCountForVintage	Computes the number of parcels defined for the vintage

The parcel object describes details for the proposed batches of fruit defined for delivery to a winery. It is sourced from the *t_Plan_Parcels* table because the data is vintage specific and more related to the vintage plan than the block.

ID	Unique identifier for the parcel
BlockID	Back reference to the Block that the parcel is defined from
VineyardID	Back reference to the Vineyard for the block for the parcel
VinDefID	If the parcel has been allocated to a Vinification Definition, this is the ID of that vindef

Delivered	Flag indicating that the parcel has been delivered, computed from the delivered parcels collection for the nominated winery
ExpectedBlockYield	The total tonnes expected for the block that this parcel belongs to.
Description	Text describing the parcel
AmountType	Parcels can be defined as (1) a proportion of the total for the block, (2) absolute tonnage, or (3) remainder of the block.
EquatesTo	Computed from the Amount type and expected yield of the block, this is the tonnage that the parcel equates to.
Amount	The magnitude of the amount, depending on the selected amount type
ExpHarvestDate	The expected harvest date for the parcel
HarvestType	The method of harvesting to be used for this parcel
Vintage	The vintage that this parcel is defined for
WineryID	The target winery that the parcel is destined for.

12.5.3.2.2 Vineyard Block Readings

Vineyard block readings are similar to vineyard readings and wine readings. The vineyard block reading data is stored in the *t_VineyardBlockReadings* table and the object structure is...

ID	A unique identifier for the reading
BlockID	Back reference to the vineyard block that the reading applies to.
ReadingType	The type of the reading. This value comes from a list of constants defined in the system
ReadingValue	The value of the reading. This can be a number or text
ReadingDateTime	The date and time stamp for the reading
WorknoteID	The ID of the worknote that caused the reading
Vintage	The vintage that the reading belongs to

12.5.3.2.3 Materials and Stock

The materials attached to a vineyard are associated with spraying and fertilisation. The structure is identical to the Additives structures in the Winery, and the same database tables are used to store the information. Additive types will be either “Fertiliser” or “Spray”.

12.5.4 Storage Facilities

Storage facilities are basically used as a means of storing bottled wines. The storage collection has a couple of methods to search for bottled wines.

FindBW	Searches all storage facilities to find a bottled wine
FindBWForWine	Searches all storage facilities to find a bottled wine from a wine

Each storage facility object is populated from the *t_Facilities* table and consists of the following properties as well as a Bottled Wines collection.

ID	Unique identifier of the vineyard
StorageName	The name of the facility
Code	A short code or abbreviation for the facility.
Managed	Flag indicating whether the facility is managed
MainView	Flag indicating if the facility should be displayed on the main menu of the application
Image	Filename of the image used on the main menu when the facility is displayed
SubtleImage	Filename of the image of the facility used in the background of some dialogs.
Sequence	The ordinal position that this object should be displayed.

12.5.4.1 Bottled Wine

Bottled Wines are entities that represented batches of bottled wine resulting from bottling worknotes. The data is stored in the *t_BottledWines* table.

ID	Unique identifier of the bottled wine
Description	Text description of the bottled wine
WineID	ID of the wine that was bottled to create this bottled wine entity
LabelID	ID reference to Labels object in global parameters
BottleType	ID reference to Bottle Type object in global parameters
BoxType	ID reference to Box Type object in global parameters
DividerType	ID reference to Divider Type object in global parameters
PadType	ID reference to Pad Type object in global parameters
CapsuleID	ID reference to Capsule Type object in global parameters
ClosureID	ID reference to ClosureType object in global parameters
WNID	ID of the bottling worknote that generated this bottled wine

BottleCount	Count of bottles in this bottled wine object
StorageID	Back reference to the storage facility

12.5.5 Laboratories

The Laboratory object is consistent with the other facility objects

ID	Unique identifier of the Laboratory
LaboratoryName	The name of the Lab
Code	A short code or abbreviation for the Lab.
Managed	Flag indicating whether the Laboratory is managed
MainView	Flag indicating if the Laboratory should be displayed on the main menu of the application
Image	Filename of the image used on the main menu when the Lab is displayed
SubtleImage	Filename of the image of the Lab used in the background of some dialogs.
Sequence	The ordinal position that this object should be displayed.

12.5.5.1 Enzyme Kits

Each Laboratory object has a collection of Enzyme Kits which describe various parameters used to do the calculations for the tests. Each type of kit is assumed to use the same method and calculation, differing only in the constants in the equation. The data is maintained in the *t_CS_EnzymeKits* table.

ID	Unique identifier for the kit
TestType	System constant for the type of kit
LabID	Back reference to the Laboratory object
Brand	Text of the brand or kit model
Factor1	Constant used in the calculation
Factor2	Constant used in the calculation

12.5.6 Bottling Facilities

Bottling plants are not loaded in the menu like other facilities because there is no functionality required on the interface. Of course if another interface is designed that can use the object then it is available. However bottling facilities only appear in the bottling worknote dialog.

ID	Unique identifier of the Plant

PlantName	The name of the Plant
Code	A short code or abbreviation for the Plant.
Managed	Flag indicating whether the Plant is managed
MainView	Flag indicating if the Plant should be displayed on the main menu of the application
Image	Filename of the image used on the main menu when the Plant is displayed
Sequence	The ordinal position that this object should be displayed.

12.6 The Database and System Constants

The database consists of several tables. There are no macros, queries or other relational database features used in the implementation of this project in order to avoid any commercial product specific features. The tables are all named using a simple "t_" notation so that they are easily found in the program in the form of structured query language (SQL) statements.

Codeset tables are given a "t_CS_" prefix to differentiate them from other data tables. The role of the codeset table is to contain static reference data, in the sense that the information will change little over the usual operation of the system. This definition is blurred somewhat by the fact that some reference data can change more often than the term "static" implies, depending on the nature of the data and the disposition of the user. The most obvious example of this is the t_CS_Additives table which seemed to need constant updating when new users arrive on the scene, or during the vintage preparation period when a lot of these products are purchased. System tables were given a "t_VWMS" prefix, and other tables were given a prefix befitting the context of their use; including "t_Wines" and "t_Plan".

Nearly all tables have an ID field as an auto-number primary key, which helps to ensure that data referentiality is maintained. The tables are detailed in *Appendix B : Database Tables* and *Appendix C : System Constants*

13. THE APPLICATION

Given the data model structure, the database, and the defined system constants, the basis for demonstrating how the application uses these structures to apply business rules and display a user interface is now available. It has to be reiterated that the process of building the model and then the application was not sequential. Many of the features of the model and methods within were prompted by the requirements of the application. Indeed, it is perfectly reasonable to expect that the tools to manipulate the model data need to be tailored to the purpose they fulfil. It was reassuring to note that very little of the structure of the data model changed during development.

Rather than just dump the code into this document, it is more useful to provide pseudo-code for the application so that it is more readily understood by those unfamiliar with the programming language syntax. For the same reason, the structure of this chapter will be to describe the application in logical and functional user-interface features rather than any code based format. Screen shots will also be provided to illustrate the application and functionality. These will come from the pilot version of the application running at the Whitebox Winery in Heathcote, Victoria.

It is assumed that the reader is familiar with the concept of modal dialogs which are windows that appear and need to be closed before the user can interact with the other parts of the system. Most windows in the system are modal dialogs apart from the MDI child windows which can only be viewed one at a time anyway. Many of these modal dialogs can be invoked in one of three modes; New, Edit, Complete. These modes equate to the process being applied to the jobs or worknotes that the dialog are for. For example, if we say that the “Transfer Wine” dialog is invoked in “New” mode then we know that the dialog will create a new job and worknotes rather than expect input of details of an existing job. Obviously, the mode may also affect the appearance of the dialog to such an extent that it might appear to be a completely separate dialog.

13.1 LOADING THE DATA MODEL – INITIALISATION

Obviously, the start-up sequence for the application is purely procedural, requiring no user input. A Splash Screen (Figure 22) is displayed to give the user an indication of progress.

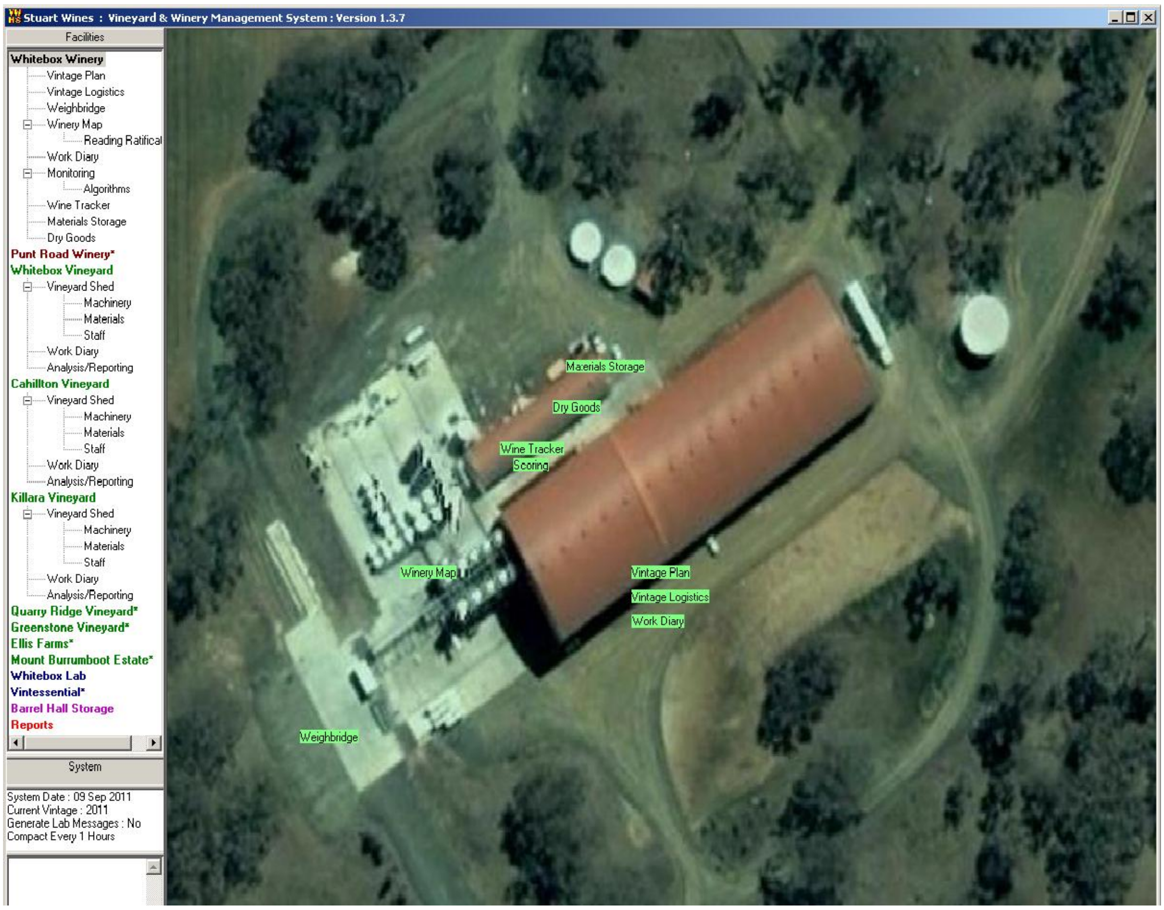
After performing a few checks, the data model is loaded. As each facility is loaded, some information is also added to an array (called gMenu) that is used to work out the order for displaying the facilities in the tree control.

Figure 22 Application Splash Screen

```
Check that no other instance of the application is running
Create the instance of the Data Layer object (gDL).
     Retrieve the database parameters from the system registry
     Create a connection to the database
Create instance of Global Parameters object (gGP)
      Load constants into gGP from system registry and database
Check that the database has been recently compacted
      If not, attempt to compact the database
Create an instance of the Application User Object (gAU)
Create as instance of the Enterprise object (gEnt)
Attempt to logon
     Retrieve the current user using the "GetUserName" API
     Populate gAU. Permissions etc
            If there is no data then exit the application.
     Set the user to "Logged On" in the database.
Load the Splash Screen
Check the Database version matches the application version
      If there is a mismatch then exit
Load the data model
      Load the gGP code sets
      Load the Job Types into gGP
            For each job type load the worknote types
      Load the Vessel Types into gGP
      Loop through and load the Wineries into gEnt
            Load the facility information
            If the winery is managed
                  Add gMenu information
                  Load Menu Images
                  Load the Monitoring Algorithms
                  Load the Wine Groups
                  Load the Additives
                  Load the Dry Goods Stock
                  Load the Load Winery Menu Labels
                  Load the Equipment
                  Load the Wines
                  Load the Vessels
                  Load the Heating and Cooling Methods
                  Load the Skins
                  Load the Wine Batches
                  Load the Delivered Parcels
                  (Note: the vindefs are not loaded yet because the
                  vineyards have not been loaded)
            If the winery is not managed
```

```
                  Add gMenu information
                  Load the Wines
                  Load the Vessels
                  Load the Delivered Parcels
            Load the Diary entries
      Loop through and load the Vineyards into gEnt
            Load the facility information
            Load the vineyard readings
            If the vineyard is managed
                  Add gMenu information
                  Load Menu Images
                  Load the Diary entries
            Load the vineyard blocks
                  Vineyard parameters
                  Load Block Readings
                  Set the block expected yield from the readings
                  Load Block Parcels
                  Create default parcel if date and yield specified
      Load Vindefs (now that vineyards are loaded)
      Load Wine Batches (now that vineyards are loaded)
      Loop through and load the Laboratories into gEnt
            Load the facility information
            If the lab is managed
                  Add gMenu information
                  Load Menu Images
            Load the enzyme kits
      Loop through and load the Storage Facilities into gEnt
            Load the facility information
            If the facility is managed
                  Add gMenu information
                  Load Menu Images
            Load the bottled wines for this facility
      Loop through and load the Bottling Plants into gEnt
            Load the facility information
      Load the Standing orders for the enterprise into gEnt
      Load the jobs into gEnt
            Load job information
            Load worknotes for this job
            Recompute the status and save the job if it is not complete
      Load the Fruit Reception jobs into the Jobs object
            Load the allocations for each Fruit reception job
                  Find the wine for this parcel and add it to the wines
                  WineFruitParcels collection
      Compute the vessel status for each vessel in each winery
            Set all vessels status to the state of the vessel
            Loop through all the current wines
                  Get the vessel from the wines readings
                  Convert the wine status to a vessel status
                  Apply that status to the vessel
Load the MDI
      Loop through the gMenu array to add the facilities and sub nodes
Retrieve the last selected node from the system registry
Select this node from the menu
      This loads the matching screen into the MDI
Set the system Parameters label under the tree menu
Unload the splash screen
```

13.2 The MDI interface



System Date : 09 Sep 2011
Current Vintage : 2011
Generate Lab Messages : No
Compact Every 1 Hours

Figure 23. MDI Interface

The MDI (Multiple Document Interface) window (Figure 23) is a Microsoft Windows standard user interface structure that allows the presentation of dialogs within the same boundary window. This allows some features to always be on the screen. In this case, the panels on the left are permanently placed to display the navigation through the application and other important system information. The top panel is a so-called treeview control, chosen because it is a familiar concept for Windows application users. It displays the facilities configured for the system and various applicable child nodes underneath each facility node. Wineries appear as green, Vineyards as burgundy, Laboratories as blue, storage facilities as purple and a Reports Node appears at the bottom in red. Each of these node types will be described separately, as selecting each node using the mouse or keyboard causes a different screen to appear in the main body of the MDI. Below the tree is a System panel which displays important system information. Below this is a field to display system messages. The MDI form also allows multiple so-called "child" windows to be loaded and displayed at once. In this instance, all child windows are maximised automatically so that only one is visible at a time, but several can be loaded if circumstances are appropriate.

13.2.1 About Permissions and Facility Locking

Before going into details of the main features of the MDI, it is first important to describe the concept of facility locking which is a feature of the system. One of the main tenets of the user interface was to provide a context to feed and manipulate the data model. This allowed for a freedom of access to all the features of the model where it seemed contextually appropriate so that the flow of logic for the user would be less restricted by the application’s structure. In other words, the user should be able to jump from feature to another, contextually, without losing the winemaking flow of logic.

This freedom means that it is potentially dangerous to allow more that one user the ability to change data in the model while other people might be accessing it. We do not want to restrict access to the data, but we need to ensure data integrity by locking update permissions to logical areas of the model. Given the freedom we also want to allow, it is necessary to make the lock out boundary for the model quite large. In fact it was found that the most suitable boundary was on a facility basis. This meant that, permissions allowing, the first authorised user to select a node will effectively lock out any other user from updating data for that facility. Any other user subsequently selecting a node from the same facility will not be able to perform editing or updating of any data in that context. The words “(READ-ONLY)” will appear on the application title bar. (Figure 24)

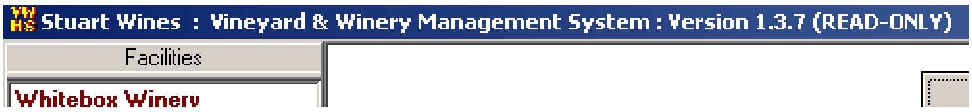

Stuart Wines : Vineyard & Winery Management System : Version 1.3.7 (READ-ONLY)

Figure 24. Title Bar on MDI

This was not regarded as a serious restriction because, given the scale of the enterprise that the system is designed for, it is unlikely that more than one person will need to be updating the data at a time. A consequence of this design is that the presumption that this user is the only updater, affords some performance improvement; an issue which becomes important as the model fills, and is discussed later. The permissions scheme used for the application is a simple 4-tier system.

	User Object Flags		
Permission Level	**Editor**	**Administrator**	**SuperAdministrator**
Read-Only			
Normal User	✓		
Administrator	✓	✓	
Super Administrator	✓	✓	✓

Table 14. Permissions Scheme

These levels are set by flags in the user object which are referenced by all dialogs and menu items in the user interface to ensure that the various functions are available only to users with appropriate settings. In general, Administrator level allows ratification of laboratory results, access to system messages, and the ability to reset wine volumes. These functions are deemed the responsibility of senior winemakers and so the Administrator level permission might also be seen as a senior user. The Super Admin permission level also allows access to the system parameters window and the ability to create a wine as a means of introducing new wines without any provenance.

13.2.2 The Menu

Each node in the tree is assigned a node type as it is loaded. It also has information regarding the facility to which it belongs. So when a node is selected several things occur...

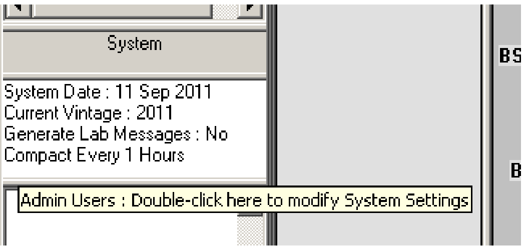
```
Extract the Node Type from the selected node
If this node is different to the previously selected node
      Extract the facility ID from the node
      If the user does not have this facility locked out
            Unlock any facilities they do have locked
            Attempt to lock the selected facility
            If successful, update the user object
      Update the title bar on the MDI accordingly
      If the facility has changed or isn’t locked by the user
            Unload all the MDI child forms
      Check to see if the required form is already loaded
      If not then load the form
      Otherwise just set focus to the form
```

(The details of what happens when a specific form is loaded are detailed for each form individually in the following sections of this chapter.)

Notice that any other forms associated with the same facility are not unloaded. This means that it is possible to have several child windows open at one time, thereby saving on loading time. By right clicking on the tree control, a popup menu appears that allows the user to surrender control of the selected facility, if they have it locked out. This causes the facility to become unlocked. If the user then clicks on another node, this will lock the underlying facility again, so it is just a temporary state.

13.2.3 The System Panel

The system panel is on the left of the MDI (Figure 25) , underneath the treeview facility menu. Its function is to display various system parameters. If the user has Super Admin permissions then double clicking on this panel (as the tooltip states) will display the System Settings dialog. (Figure 26)

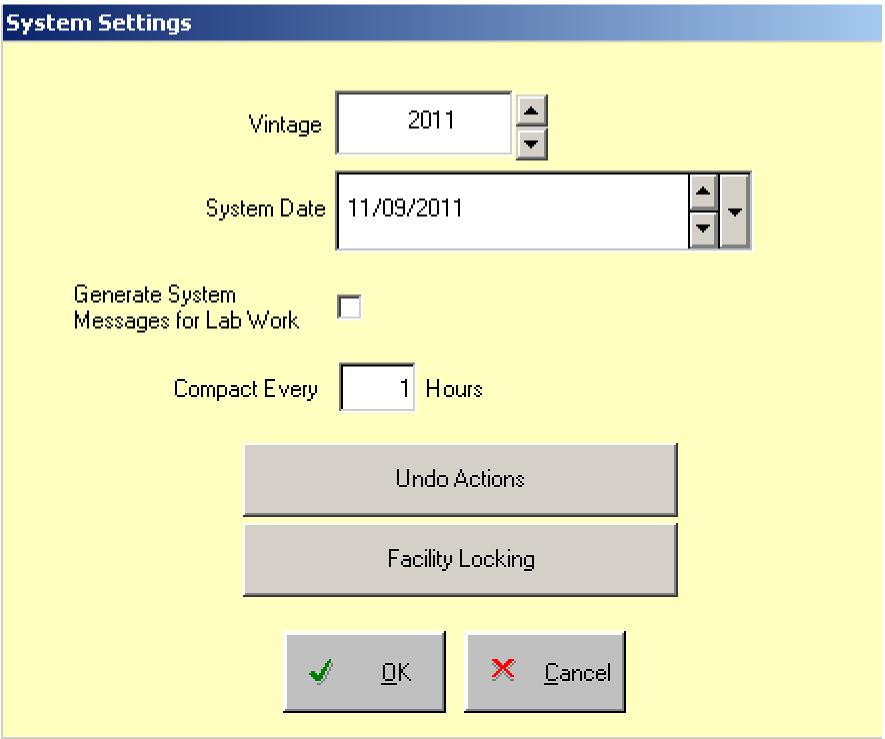


System

System Date : 11 Sep 2011
Current Vintage : 2011
Generate Lab Messages : No
Compact Every 1 Hours

Admin Users : Double-click here to modify System Settings

Figure 25. The SystemPanel



System Settings

Vintage: 2011

System Date: 11/09/2011

Generate System Messages for Lab Work: □

Compact Every: 1 Hours

Figure 26. System Settings

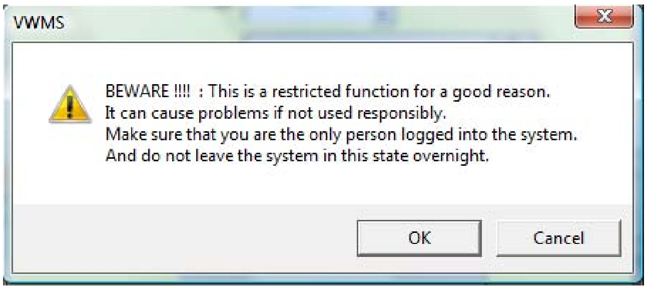
System settings are global parameters and functions that require senior authority to change or execute.

13.2.3.1 Vintage

The Vintage is the year tag that is used to indicate which processing year will automatically be associated with any activities on the system. The definition of a vintage is left to the user mainly because it was difficult to get consensus on a universal definition. For some it was convenient to use financial year dates as defined by the Tax Department; for others, the calendar year was appropriate. Some viticulturists regard a new vintage as starting the second the last fruit comes off their vines.

13.2.3.2 SYSTEM DATE

The System Date defines the date the system thinks is current. This is a useful but dangerous parameter to manage other than when it is set to the current date. This is because it can cause events and associated parameters to be inserted into the sequence even though they were not present or taken into consideration with events or settings that will now appear subsequent to them.



VWMS

BEWARE !!!! : This is a restricted function for a good reason.
It can cause problems if not used responsibly.
Make sure that you are the only person logged into the system.
And do not leave the system in this state overnight.

Figure 27. Warning Message for modifying the System Date

Hence the message shown above is displayed when it is changed to anything other than the real current date.

13.2.3.3 LAB WORK SYSTEM MESSAGES

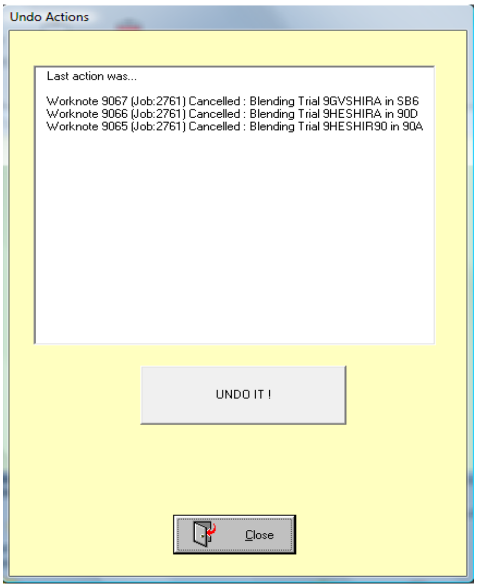
Whenever lab work is completed, winemakers need to know that results are available and may need to be ratified and the implications of the readings need to be considered. This flag tells the system to generate a system message each time a lab result is generated.

13.2.3.4 COMPACT EVERY X HOURS

When the system starts up, it checks when the database was last compacted and uses this value to determine if it needs to be compacted again. The database cannot be compacted unless there are no users connected, so it is common for the database to only get compacted once or twice a day even if this value is set to one or 2 hours. It also depends on the scale of the enterprise and the number of users accessing it. The database really only needs compacting every few days anyway, and the process only takes a few seconds, so it is not a critical issue.

13.2.3.5 UNDO ACTIONS

Despite all the check and balances, and despite an expressed determination to not allow this, albeit tacitly, it became necessary to provide a rollback function to allow the user to undo actions performed on the system. (Figure 28) This requires an extensive scanning of the data model to search for the last action, and then a thorough purge of all the data for any references to this action. The key to the search and delete function is the ubiquitous date and time stamp which is attached to all readings and events in the data model. The searches are performed on the database rather than the data model to ensure the latest data is available and to allow more precise SQL to be used rather than memory intensive data model searches.



Last action was...

Worknote 9067 (Job:2761) Cancelled : Blending Trial 9GVSHIRA in SB6
Worknote 9066 (Job:2761) Cancelled : Blending Trial 9HESHIRA in 90D
Worknote 9065 (Job:2761) Cancelled : Blending Trial 9HESHIR90 in 90A

Figure 28. Undo Actions

The Search function to determine the last action is...

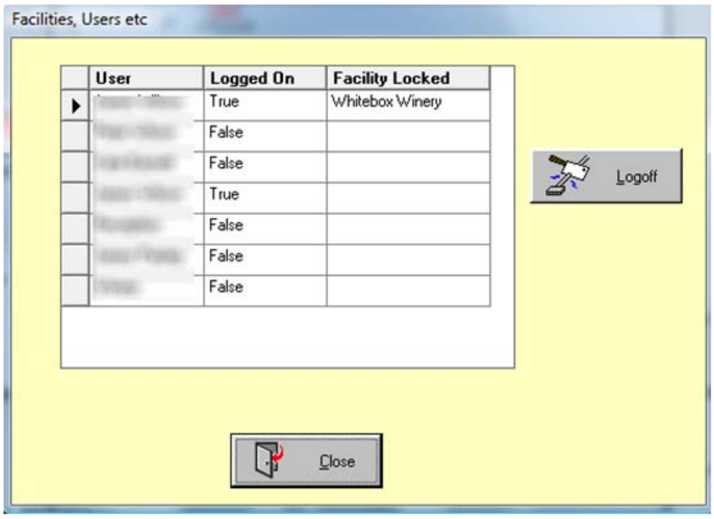
- List all the worknotes in order of their creation date (latest first)
- Record the details of the first worknote in the list and any other with the same creation date-time
- List all the worknotes in order of their completion date (latest first)
- If the completion date-time of the first worknote in the list is later than the current latest date from earlier then Record the details of the first worknote in the list and any other with the same completion date-time. Note that the worknote may have been cancelled rather than completed.
- List all of the wine readings with no worknote id in order of their date-time stamp (latest first)
- If the completion date-time of the first reading in the list is later than the current latest date from earlier then Record the details of the reading in the list and any other with the same date-time.
- List all of the vessel readings with no worknote id in order of their date-time stamp (latest first)
- If the completion date-time of the first reading in the list is later than the current latest date from earlier then Record the details of the reading in the list and any other with the same date-time.

Note that the vessel and wine readings that are retrieved are those without worknote ids because these have been generated by actions that are not associated with worknotes. For example, when a vessel is moved from one barrel set to another. Readings with worknotes will already be covered by the earlier searches for worknotes themselves.

Once the list of last actions has been generated and presented in the list, pressing the undo button will cause the following process to occur...

```
Loop through the list of recorded details for the actions
    If it is a wine reading
        Delete reading from the database and refresh the wine object
    If it is a vessel reading
        Delete reading from the database and refresh the vessel object
    If it is a worknote creation
        Delete the worknote
    Delete any wine readings, vessel readings, LIP, Chips records for this worknote
        If the worknote was a fruit processing wn
            Delete any temporary wines created
            Delete any Processing jobs and allocations created
    If it is a worknote cancellation
        Reset status of the worknote based on the required start date
        Clear the completion date
    If it is a worknote completion
        Reset status of the worknote based on the required start date
        Clear the completion date
        Delete wine & vessel readings, LIP, Chips records for worknote
        If the worknote was a fruit processing worknote
            Delete any temporary wines created
            Delete any Processing jobs and allocations created
If a worknote was deleted or recreated
Check associated job to recreate, delete or status reset
Refresh all jobs in the data model.
```

13.2.3.6 Facility Locking



Facilities, Users etc

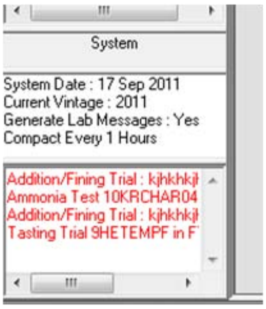
User	Logged On	Facility Locked
[redacted]	True	Whitebox Winery
[redacted]	False	
[redacted]	False	
[redacted]	True	
[redacted]	False	
[redacted]	False	
[redacted]	False	

Figure 29. Facility Locking dialog

Sometimes it is necessary to know who is logged in to the application, or who has a particular facility locked out, and log them out of the system. By selecting a user then pressing the Logoff button, a flag is set in the database. Each instance of the application has a timer that checks every 5 seconds to see of this flag has been set for the currently logged on user, and shuts down if set.

13.2.4 System Messages

The area below the system parameters panel (Figure 30) is reserved for system messages to be displayed. This provides a useful notification system for notifying users of activities of other users. The most common use is for laboratory messages which let other users know the results of laboratory analyses. A brief message is generated and inserted in the *t_VWMS_SystemMessages* table which is periodically checked by a timer to see whether any uncleared messages are there. By double-clicking on the panel (if the user has Admin permission), the system messages screen appears (Figure 31) to display the messages and allow the messages to be cleared.

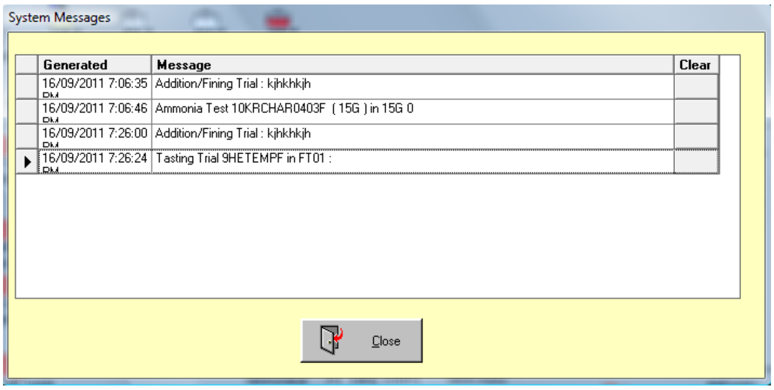


System

System Date : 17 Sep 2011
Current Vintage : 2011
Generate Lab Messages : Yes
Compact Every 1 Hours

Addition/Fining Trial : kjhkhkjł
Ammonia Test 10KRCHAR04
Addition/Fining Trial : kjhkhkjł
Tasting Trial 9HETEMPF in F

Figure 30. System Panel



System Messages

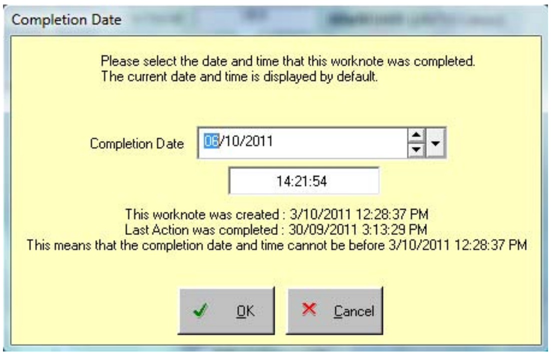
	Generated	Message	Clear
	16/09/2011 7:06:35 PM	Addition/Fining Trial : kjhkhkjh	
	16/09/2011 7:06:46 PM	Ammonia Test 10KRCHAR0403F (15G) in 15G 0	
	16/09/2011 7:26:00 PM	Addition/Fining Trial : kjhkhkjh	
▶	16/09/2011 7:26:24 PM	Tasting Trial 9HETEMPF in FT01 :	

Figure 31. The System Messages dialog

Clearing a message by pressing the “Clear” command button at the end of line, clears the message for all users in the system.

13.2.5 Other System Wide Dialogs

Jobs and worknotes are system wide structures, although they do tend to be specific to one or two facility types. When any worknote is completed however, it is important to know when the task was completed or, in the case of a laboratory worknote, when the sample was taken. Obviously, there are also rules about what dates and time can be specified. The user cannot specify a future date because this could cause data integrity problems. Nor can the date or time go back too far, beyond the date and time the worknote was created, or before the object it is applied to, came into existence. There are two dialogs (both with the universal pale yellow background indicating that they are system wide dialogs), (Figure 32 and Figure 33) that are used in this context, whenever a worknote is being completed.



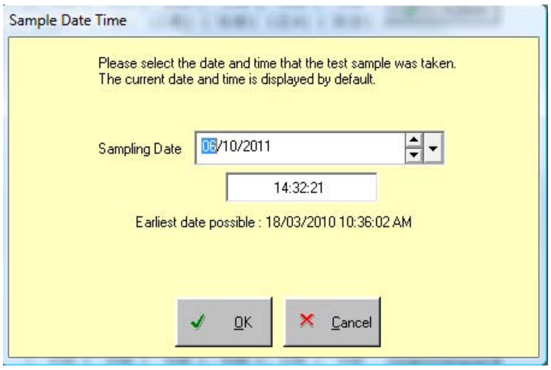
Completion Date

Please select the date and time that this worknote was completed.
The current date and time is displayed by default.

Completion Date 06/10/2011
14:21:54

This worknote was created : 3/10/2011 12:28:37 PM
Last Action was completed : 30/09/2011 3:13:29 PM
This means that the completion date and time cannot be before 3/10/2011 12:28:37 PM

Figure 32. Completion Date Time Dialog



Sample Date Time

Please select the date and time that the test sample was taken.
The current date and time is displayed by default.

Sampling Date 06/10/2011
14:32:21

Earliest date possible : 18/03/2010 10:36:02 AM

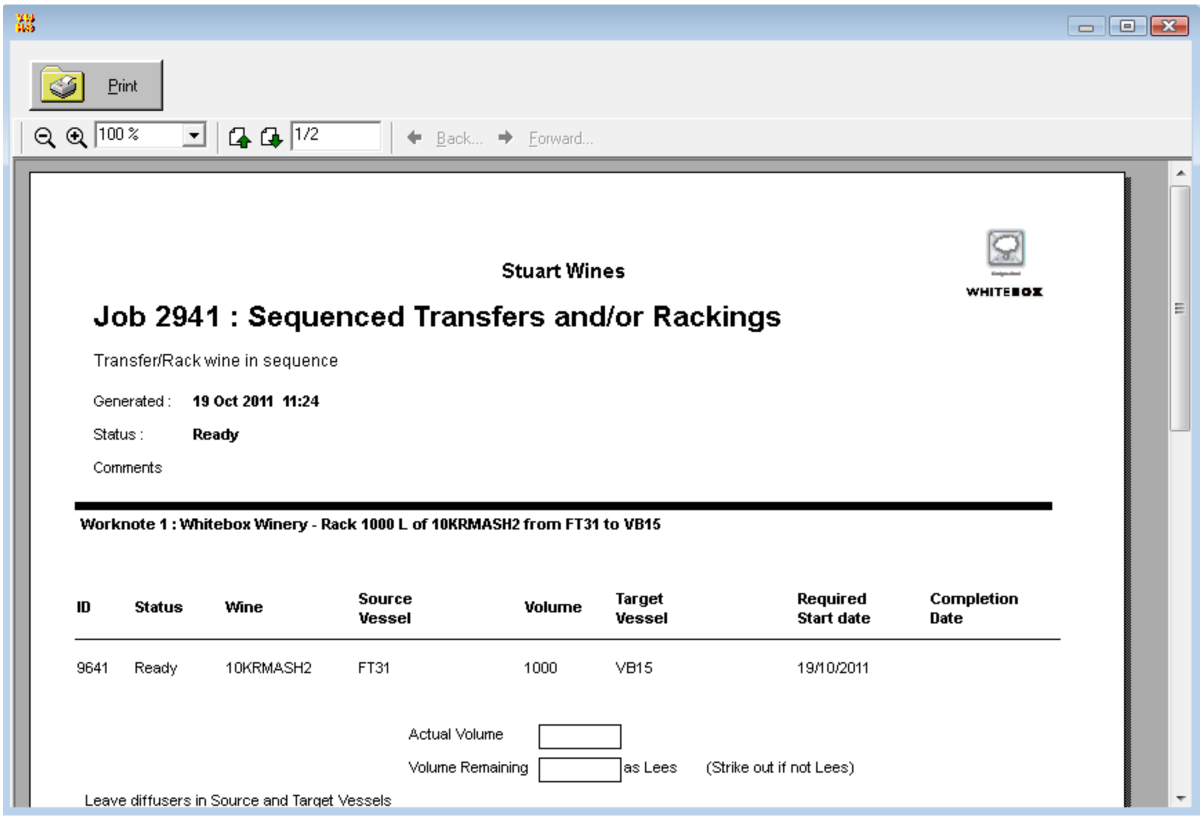
Figure 33. Sample Date Time dialog

Each dialog is initialised with information about when the last action was applied to the entity in question. This is used in the descriptive message that appear in the dialog, and also help set the minimum date and time controls. The date and time fields default to the current date and time, but when the date is set to the minimum allowable date, the time is also set to the minimum time if it is currently set to a time that is too early.

The significance of the date and time is contextual. For example, for a process applied to a wine, the date and time needs to be when the task was completed. For a laboratory test on a wine, the date and time the sample was taken is the critical factor because this identifies the state of the wine. Of course, if a process is being performed on a wine while a sample is taken, then this cross-over state of the wine might be regarded as misleading. The option to also have a start date and time for a process would overcome this, but this is far too over-engineered and regimented to be necessary.

The ambiguity over the state of a wine for a particular analysis is less likely for small scale operations anyway.

The Job Viewer dialog (Figure 34) is also common across the application as a means of displaying a job and worknotes in a form that can be printed for cellar or vineyard staff. The dialog features a print button, as well as zoom and page features.



Stuart Wines

Job 2941 : Sequenced Transfers and/or Rackings

Transfer/Rack wine in sequence

Generated : 19 Oct 2011 11:24

Status : Ready

Comments

Worknote 1 : Whitebox Winery - Rack 1000 L of 10KRMASH2 from FT31 to VB15

ID	Status	Wine	Source Vessel	Volume	Target Vessel	Required Start date	Completion Date
9641	Ready	10KRMASH2	FT31	1000	VB15	19/10/2011	

Actual Volume

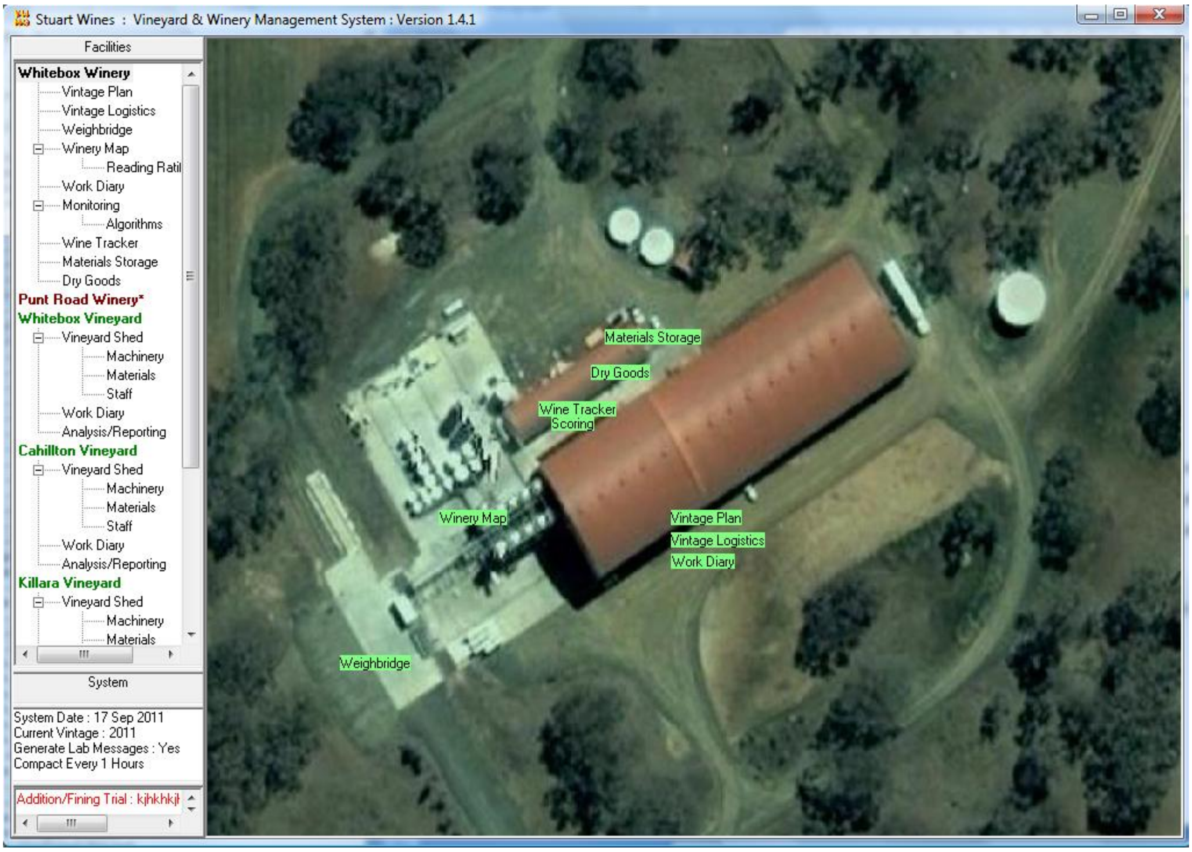
Volume Remaining as Lees (Strike out if not Lees)

Leave diffusers in Source and Target Vessels

Figure 34. Report Viewer Dialog

13.3 The Managed Winery

The main node and child nodes on the tree menu have separate screen displayed as child forms within the MDI interface. The main node for a managed winery is a simple menu of the available child nodes displayed over an image configured for the facility. In the example shown (Figure 35), a satellite image of the winery has been configured so that the menu items can be positioned in a contextually appropriate position on the screen.



System Date : 17 Sep 2011
Current Vintage : 2011
Generate Lab Messages : Yes
Compact Every 1 Hours

Addition/Fining Trial : kjhkhkjt

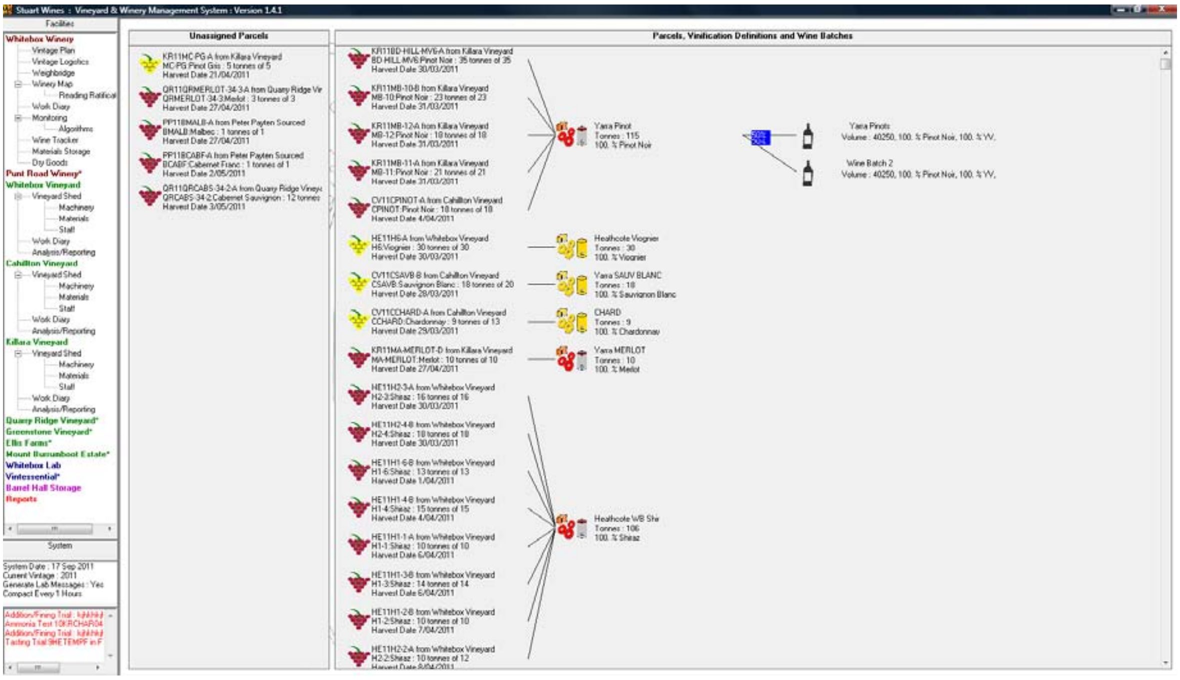
Figure 35. Managed Winery Main Menu

13.3.1 Vintage Plan

The Vintage Plan screen (Figure 36) allows the user to create vinification definitions, and wine batches and link them to planned parcels for a given vintage. In principle, the screen needs to provide the following functions.

1. Display all links between parcels, vindefs, and wine batches
2. Display a list of unallocated parcels
3. Allow vindefs and wine batches to be created and deleted
4. Allow details for vindefs and wine batches to be edited
5. Allow parcels to be linked to a single vindef
6. Allow vindefs to be linked to one or more wine batches.
7. Allow the proportion of a vindef to be varied between wine batches

These functions are performed by using common windows interactions such as drag and dropping, and double-clicking on icons as well as underlying methods and procedures attached to the object entities in the data model which encapsulate and maintain appropriate data links.



Stuart Wines : Vineyard & Winery Management System : Version 1.4.1

Facilities

Whitebox Winery
- Vintage Plan
- Vintage Logistics
- Weighbridge
- Winery Map
  - Reading Ratifical
- Work Diary
- Monitoring
  - Algorithms
- Wine Tracker
- Materials Storage
- Dry Goods

Punt Road Winery*

Whitebox Vineyard
- Vineyard Shed
  - Machinery
  - Materials
  - Staff
- Work Diary
- Analysis/Reporting

Cahillton Vineyard
- Vineyard Shed
  - Machinery
  - Materials
  - Staff
- Work Diary
- Analysis/Reporting

Killara Vineyard
- Vineyard Shed
  - Machinery
  - Materials
  - Staff
- Work Diary
- Analysis/Reporting

Quarry Ridge Vineyard*
Greenstone Vineyard*
Ellis Farms*
Mount Burrumboot Estate*
Whitebox Lab
Vintessential*
Barrel Hall Storage
Reports

System

System Date : 17 Sep 2011
Current Vintage : 2011
Generate Lab Messages : Yes
Compact Every 1 Hours

Addition/Fining Trial : kjhkhkj
Ammonia Test 10KRCHAR04
Addition/Fining Trial : kjhkhkj
Tasting Trial 9HETEMPF in F

Unassigned Parcels

KR11MC-PG-A from Killara Vineyard
MC-PG Pinot Gris : 5 tonnes of 5
Harvest Date 21/04/2011

QR11QRMERLOT-34-3-A from Quarry Ridge Vir
QRMERLOT-34-3:Merlot : 3 tonnes of 3
Harvest Date 27/04/2011

PP11BMALB-A from Peter Payten Sourced
BMALB:Malbec : 1 tonnes of 1
Harvest Date 27/04/2011

PP11BCABF-A from Peter Payten Sourced
BCABF:Cabernet Franc : 1 tonnes of 1
Harvest Date 2/05/2011

QR11QRCABS-34-2-A from Quarry Ridge Vineya
QRCABS-34-2:Cabernet Sauvignon : 12 tonnes
Harvest Date 3/05/2011

Parcels, Vinification Definitions and Wine Batches

KR11BD-HILL-MV6-A from Killara Vineyard
BD-HILL-MV6:Pinot Noir : 35 tonnes of 35
Harvest Date 30/03/2011

KR11MB-10-B from Killara Vineyard
MB-10:Pinot Noir : 23 tonnes of 23
Harvest Date 31/03/2011

KR11MB-12-A from Killara Vineyard
MB-12:Pinot Noir : 18 tonnes of 18
Harvest Date 31/03/2011

KR11MB-11-A from Killara Vineyard
MB-11:Pinot Noir : 21 tonnes of 21
Harvest Date 31/03/2011

CV11CPINOT-A from Cahillton Vineyard
CPINOT:Pinot Noir : 18 tonnes of 18
Harvest Date 4/04/2011

Yarra Pinot
Tonnes : 115
100. % Pinot Noir

Yarra Pinots
Volume : 40250, 100. % Pinot Noir, 100. % YV,

Wine Batch 2
Volume : 40250, 100. % Pinot Noir, 100. % YV,

HE11H6-A from Whitebox Vineyard
H6:Viognier : 30 tonnes of 30
Harvest Date 30/03/2011

Heathcote Viognier
Tonnes : 30
100. % Viognier

CV11CSAVB-B from Cahillton Vineyard
CSAVB:Sauvignon Blanc : 18 tonnes of 20
Harvest Date 28/03/2011

Yarra SAUV BLANC
Tonnes : 18
100. % Sauvignon Blanc

CV11CCHARD-A from Cahillton Vineyard
CCHARD:Chardonnay : 9 tonnes of 13
Harvest Date 29/03/2011

CHARD
Tonnes : 9
100. % Chardonnay

KR11MA-MERLOT-D from Killara Vineyard
MA-MERLOT:Merlot : 10 tonnes of 10
Harvest Date 27/04/2011

Yarra MERLOT
Tonnes : 10
100. % Merlot

HE11H2-3-A from Whitebox Vineyard
H2-3:Shiraz : 16 tonnes of 16
Harvest Date 30/03/2011

HE11H2-4-B from Whitebox Vineyard
H2-4:Shiraz : 18 tonnes of 18
Harvest Date 30/03/2011

HE11H1-6-B from Whitebox Vineyard
H1-6:Shiraz : 13 tonnes of 13
Harvest Date 1/04/2011

HE11H1-4-B from Whitebox Vineyard
H1-4:Shiraz : 15 tonnes of 15
Harvest Date 4/04/2011

HE11H1-1-A from Whitebox Vineyard
H1-1:Shiraz : 10 tonnes of 10
Harvest Date 6/04/2011

HE11H1-3-B from Whitebox Vineyard
H1-3:Shiraz : 14 tonnes of 14
Harvest Date 6/04/2011

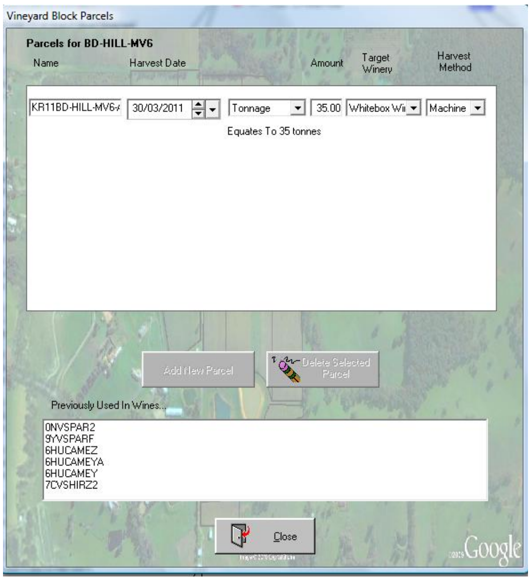
HE11H1-2-B from Whitebox Vineyard
H1-2:Shiraz : 10 tonnes of 10
Harvest Date 7/04/2011

HE11H2-2-A from Whitebox Vineyard
H2-2:Shiraz : 10 tonnes of 12
Harvest Date 8/04/2011

Heathcote WB Shi
Tonnes : 106
100. % Shiraz

Figure 36. The Vintage Plan Screen

When the screen loads, it first loads a vindef, positioning it based on the number of parcels attached to it, then load those parcels, then the wine batches attached to it, then repeat these steps for the next vindef. This means that the objects are evenly displayed on the screen as well as can be achieved. The vindefs are apportioned to each wine batch according to a recorded percentage rather than an absolute amount because this amount can be affected by the size of the parcels, the



Vineyard Block Parcels

Parcels for BD-HILL-MV6

Name	Harvest Date		Amount	Target Winery	Harvest Method
KR11BD-HILL-MV6-/	30/03/2011	Tonnage	35.00	Whitebox Wi	Machine

Equates To 35 tonnes

Previously Used In Wines...

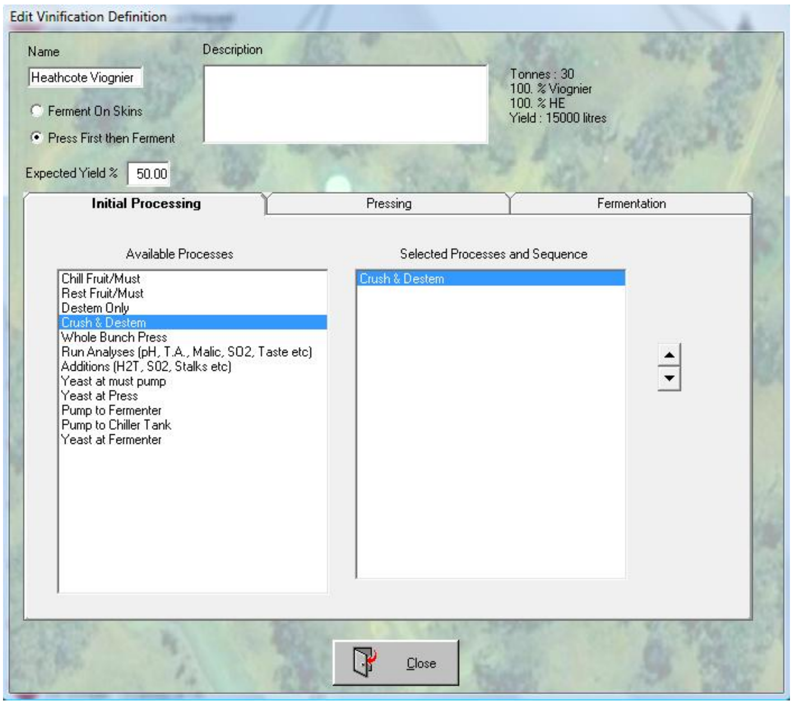
0NVSPAR2
9YVSPARF
6HUCAMEZ
6HUCAMEYA
6HUCAMEY
7CVSHIRZ2

Figure 37. Vineyard Block Parcels Dialog

expected yield from the fruit, and other details from the vindef. If any of these change, then the size of the wine batch changes rather than having to tweak the proportions.

Any unallocated parcels are listed separately in the list to the left of the main display area. These can be attached to a vindef by simply dragging and dropping the object onto a vindef. This then causes the net volume of the vindef and any wine batches attached to this vindef to be recalculated.

Right-clicking on a parcel displays a popup menu to “Deallocate Parcel”, or “Manage Parcels”. The first option causes the parcel to be unallocated from the vindef and cause the screen to be reloaded. The second option displays a window borrowed from the managed vineyard forms, (Figure 37). This dialog allows the various parameters for the parcel to be changed. It does not allow new parcels to be created, or the selected parcel to be deleted, as it does when invoked from the vineyard screen.



Edit Vinification Definition

Name: Heathcote Viognier

Description:

Tonnes : 30
100. % Viognier
100. % HE
Yield : 15000 litres

○ Ferment On Skins
● Press First then Ferment

Expected Yield % 50.00

Initial Processing | Pressing | Fermentation

Available Processes:
Chill Fruit/Must
Rest Fruit/Must
Destem Only
Crush & Destem
Whole Bunch Press
Run Analyses (pH, T.A., Malic, SO2, Taste etc)
Additions (H2T, SO2, Stalks etc)
Yeast at must pump
Yeast at Press
Pump to Fermenter
Pump to Chiller Tank
Yeast at Fermenter

Selected Processes and Sequence:
Crush & Destem

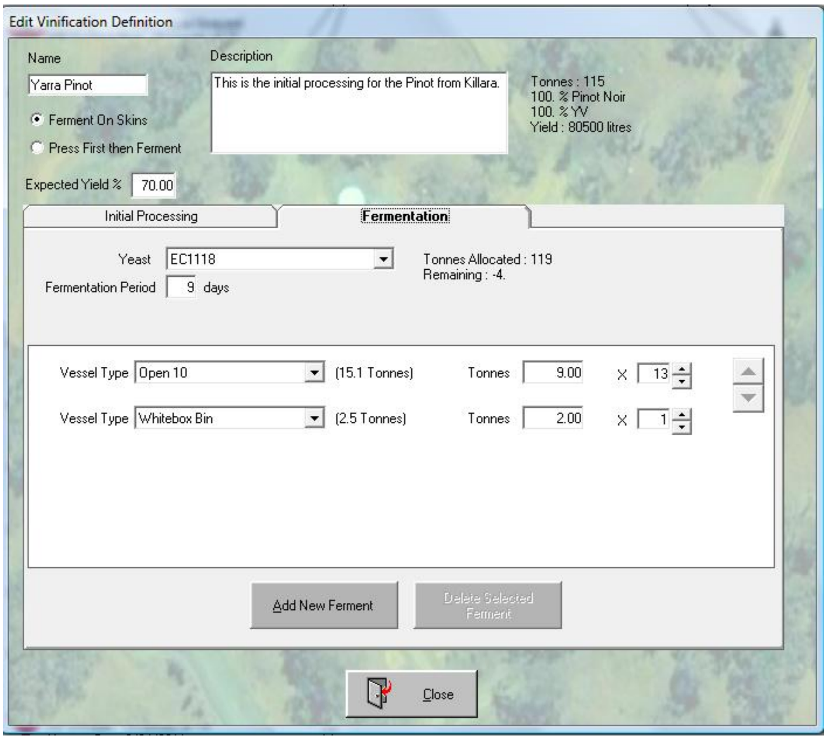
Figure 38 Edit Vinification dialog

Double –clicking on a vindef displays the “Edit VinDef” dialog (Figure 38) that allows the user to edit parameters for the vindef. It also calculates and displays the expected LIP information for the vindef based on all the source fruit allocated to this vindef. The fields on this dialog relate directly to properties of the vindef object. The pressing tab appears only when the “Press First then Ferment” option is selected. Initial processing is stipulated by selecting from the available steps and dragging them onto the selected processes list. This list can be sorted to sequence the processes in the desired order. Each time the list changes, the application rebuilds the Initial processing string in the vindef object by determining the initial processing constant associated with each action (see *20.9*

Vinification Definition Processing) and separating this constant by a colon. This technique is used to avoid spelling errors if the text is changed later.

If the Pressing tab is displayed, then it simply allows the specification of the press and pressing regime to be used, and a flag to indicate whether the output will be split into free run and pressings or treated in toto.

The Fermentation tab of this dialog (Figure 39) is the most important from a modelling point of view because it sets up the vindef allocations which feed the vintage logistics allocations. The dialog allows the user to specify the type and quantity of fermentation vessels required to ferment the juice or must. This is compared to the expected input and short falls or excesses are reported. The user also gets to specify how much of the fruit or juice is going into each vessel; giving complete control to the user rather than assuming the distribution of fruit for each vessel. Allocations can be added or removed, and may often need rechecking as vintage approaches and expected yields vary, parcels are redefined or fruit is reallocated to other vindefs.



Edit Vinification Definition

Name: Yarra Pinot

Description: This is the initial processing for the Pinot from Killara.

● Ferment On Skins

○ Press First then Ferment

Tonnes : 115
100. % Pinot Noir
100. % YV
Yield : 80500 litres

Expected Yield % 70.00

Yeast EC1118

Fermentation Period 9 days

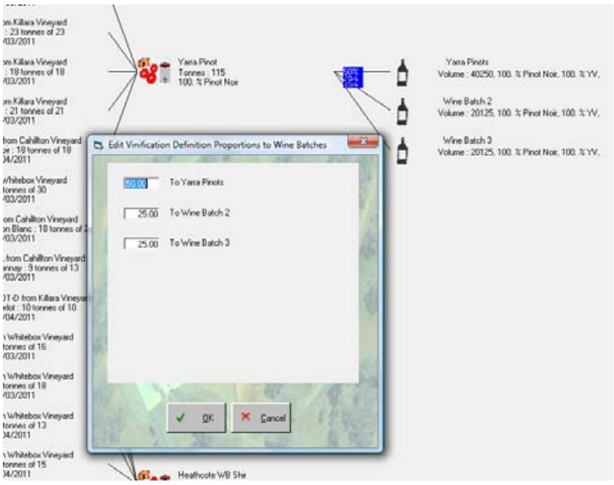
Tonnes Allocated : 119
Remaining : -4.

Vessel Type Open 10 (15.1 Tonnes) Tonnes 9.00 X 13

Vessel Type Whitebox Bin (2.5 Tonnes) Tonnes 2.00 X 1

Figure 39. Edit Vinification Definition Dialog

By right-clicking on a vindef, a popup menu is displayed with two options; to clone the vindef or to delete the vindef. The clone option is simple enough in that it creates a new vindef with many of the parameters copied from the source vindef. The delete vindef deletes the vindef from the database, and also de-allocates any parcels that were attached, any vindef allocations that were created, and any proportions of wine that were attached to any wine batch.



Edit Vinification Definition Proportions to Wine Batches

50.00 To Yarra Pinots

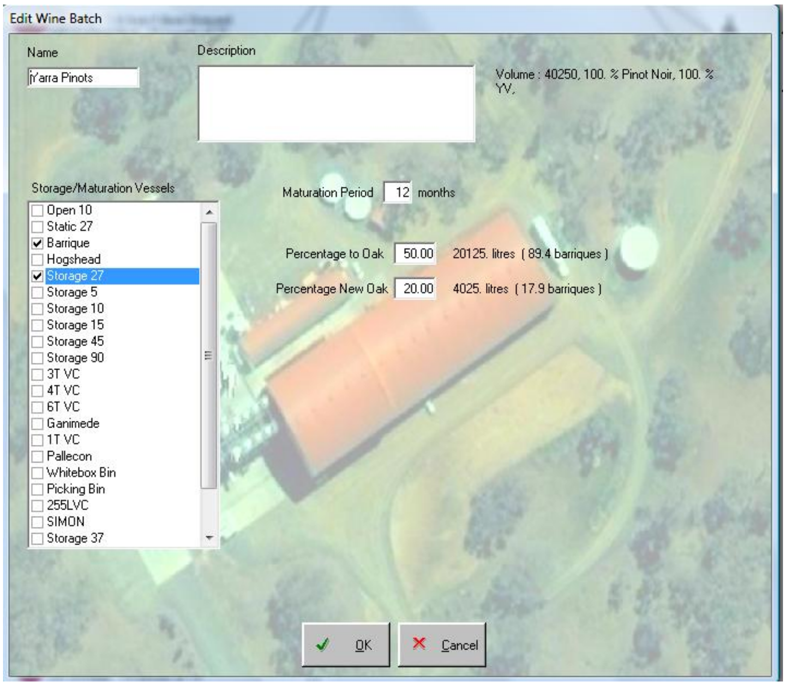
25.00 To Wine Batch 2

25.00 To Wine Batch 3

Figure 40. Edit Vindef Proportions Dialog

The proportional allocation of vindef output to the various wine batches is displayed by the blue labels indicating the percentage. By double clicking on any of these, a dialog is displayed that allows the proportions of the vindef to be altered to each of the wine batches. The OK button cannot be pressed until the percentages add up to 100. Right-clicking on a label displays a popup menu item to delete the allocation without deleting the wine batch.

By double-clicking on a wine batch icon, a dialog is displayed (Figure 41) that allows the user to specify how that wine batch is expected to be treated with respect to the maturation process. The dialog is fairly simple because it represents data at the end of the line of planning structures within the data model. The volume of wine depends on the proportions of allocations from the source vindefs. This, in turn, affects the number of vessels required to accommodate the wine. If oak is selected then the user can specify how many vessels are required and how much of that will be new oak, but there are no implications for these selections because they are only stored for reference sake, and to provide some information on expectations of new oak rather than reporting any short-falls in vessel availability. This is because planning tank space post vintage is enormously problematic before vintage. There are too many variables to consider and too many possibilities and contingencies to cover.



Edit Wine Batch

Name: Yarra Pinots

Description

Volume : 40250, 100. % Pinot Noir, 100. % YV,

Storage/Maturation Vessels

- □ Open 10
- □ Static 27
- ☑ Barrique
- □ Hogshead
- ☑ Storage 27
- □ Storage 5
- □ Storage 10
- □ Storage 15
- □ Storage 45
- □ Storage 90
- □ 3T VC
- □ 4T VC
- □ 6T VC
- □ Ganimede
- □ 1T VC
- □ Pallecon
- □ Whitebox Bin
- □ Picking Bin
- □ 255LVC
- □ SIMON
- □ Storage 37

Maturation Period 12 months

Percentage to Oak 50.00 20125. litres (89.4 barriques)

Percentage New Oak 20.00 4025. litres (17.9 barriques)

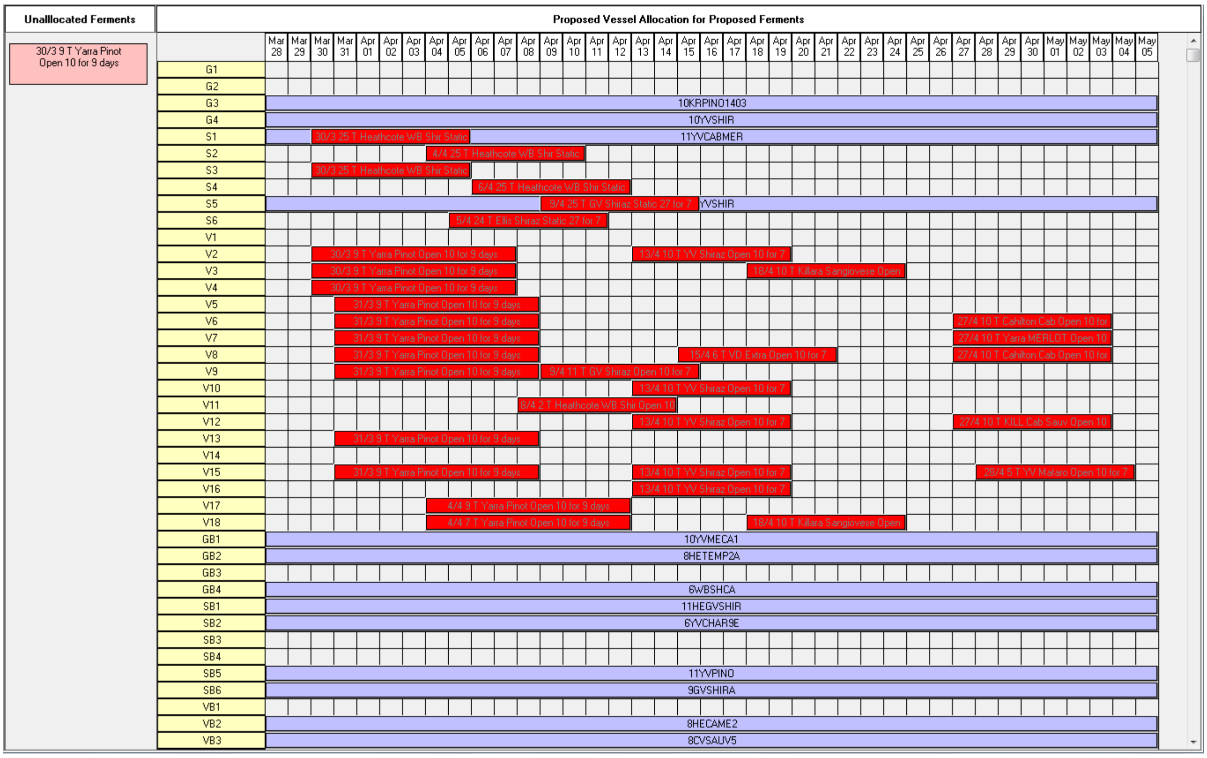
Figure 41. Wine Batch Dialog

Right-clicking on a wine batch displays a popup menu item to delete the wine batch. This also causes the proportions of any other wine batches affected to be recalculated

The only thing left from the list is to create a wine batch or a vindef, and these functions can be done by right-clicking on any vacant part of the display area. This generates a popup menu from which these options are selectable. The icons then appear but with defaulted values that can be edited by double-clicking on them.

13.3.2 Vintage Logistics

In the previous section we saw how vessel types were nominated within each vindef to accommodate the volume of fruit or wine. The Vintage Logistics screen (Figure 42) is used to book specific vessels for fermenting wines based on this vindef data. The display is based upon a spreadsheet structure that is quite familiar to many winemakers and intuitive enough for those who are not familiar with the concept. It allows the user to juggle the ferments around to find the most appropriate vessels, as well as identify and isolate bottle necks and other conflicts.



Unalllocated Ferments

30/3 9 T Yarra Pinot Open 10 for 9 days

Proposed Vessel Allocation for Proposed Ferments

Mar 28 – May 05

G1	
G2	
G3	10KRPINO1403
G4	10YVSHIR
S1	30/3 25 T Heathcote WB Shir Static 11YVCABMER
S2	4/4 25 T Heathcote WB Shir Static
S3	30/3 25 T Heathcote WB Shir Static
S4	6/4 25 T Heathcote WB Shir Static
S5	9/4 25 T GV Shiraz Static 27 for 7 YVSHIR
S6	5/4 24 T Ellis Shiraz Static 27 for 7
V1	
V2	30/3 9 T Yarra Pinot Open 10 for 9 days 13/4 10 T YV Shiraz Open 10 for 7
V3	30/3 9 T Yarra Pinot Open 10 for 9 days 18/4 10 T Killara Sangiovese Open
V4	30/3 9 T Yarra Pinot Open 10 for 9 days
V5	31/3 9 T Yarra Pinot Open 10 for 9 days
V6	31/3 9 T Yarra Pinot Open 10 for 9 days 27/4 10 T Cahilton Cab Open 10 for
V7	31/3 9 T Yarra Pinot Open 10 for 9 days 27/4 10 T Yarra MERLOT Open 10
V8	31/3 9 T Yarra Pinot Open 10 for 9 days 15/4 6 T VD Extra Open 10 for 7 27/4 10 T Cahilton Cab Open 10 for
V9	31/3 9 T Yarra Pinot Open 10 for 9 days 9/4 11 T GV Shiraz Open 10 for 7
V10	13/4 10 T YV Shiraz Open 10 for 7
V11	8/4 2 T Heathcote WB Shir Open 10
V12	13/4 10 T YV Shiraz Open 10 for 7 27/4 10 T KILL Cab Sauv Open 10
V13	31/3 9 T Yarra Pinot Open 10 for 9 days
V14	
V15	31/3 9 T Yarra Pinot Open 10 for 9 days 13/4 10 T YV Shiraz Open 10 for 7 28/4 5 T YV Mataro Open 10 for 7
V16	13/4 10 T YV Shiraz Open 10 for 7
V17	4/4 9 T Yarra Pinot Open 10 for 9 days
V18	4/4 7 T Yarra Pinot Open 10 for 9 days 18/4 10 T Killara Sangiovese Open
GB1	10YVMECA1
GB2	8HETEMP2A
GB3	
GB4	6WBSHCA
SB1	11HEGVSHIR
SB2	6YVCHAR9E
SB3	
SB4	
SB5	11YVPINO
SB6	9GVSHIRA
VB1	
VB2	8HECAME2
VB3	8CVSAUV5

Figure 42. Vintage Logistics Screen

The range of dates appearing across the top of the grid are based on the earliest expected harvest date of all the parcels being delivered to the winery and the expected last day of fermentation of all of the proposed ferments. The list on the left contains any unallocated ferments that are derived from the vindef's vintage logistics vessel allocations collection, which are created or modified whenever changes occur in the vindef's vessel allocation collection.

Wines occupying tank space are colour coded to indicate their status.

- Existing wines – Grey
- Existing Red Ferments – Red
- Existing White Ferments – Yellow
- Proposed red ferments - Pink
- Proposed white ferments - Pale Yellow
- Proposed ferment in conflict – Brown

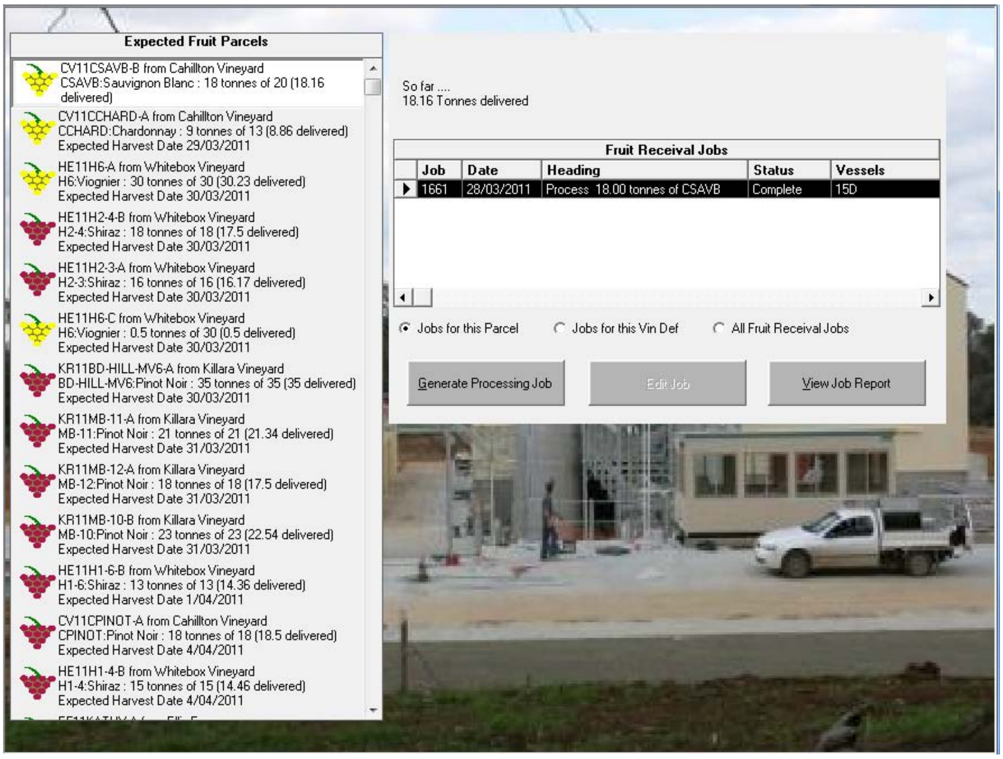
Proposed ferments can be dragged and dropped from any vessel to any other, but the date and length of the ferment is fixed according to the vindef parameters.

There are several key functions featured on this window. The first function is the auto-allocation function that is invoked from a popup menu item that appears when an unallocated ferment is right-clicked. This causes the ferment to be allocated to the first available vessel that is free for the required period. The second is the de-allocate popup menu that de-allocates a ferment back to the unallocated list, and the third function is the “Manage Vinifcation Definition” menu option that generates the same “Edit Vinification Definition” dialog from the Vinification Definition screen. (*Figure 38* Edit Vinification dialog)

13.3.3 Weighbridge

The Weighbridge screen (Figure 43) is used to receive fruit into the winery. This process receives special treatment from an application perspective because it involves several key structures in the model and requires a significant amount of specialised processing. By dedicating a screen to this, we isolate the concept and mindset and can tailor the interaction accordingly.

The list on the left contains all the parcels defined for the current vintage. They are listed in harvest date order. Selecting a parcel causes details for that parcel to be displayed in the area to the right.



Expected Fruit Parcels

CV11CSAVB-B from Cahillton Vineyard
CSAVB:Sauvignon Blanc : 18 tonnes of 20 (18.16 delivered)

CV11CCHARD-A from Cahillton Vineyard
CCHARD:Chardonnay : 9 tonnes of 13 (8.86 delivered)
Expected Harvest Date 29/03/2011

HE11H6-A from Whitebox Vineyard
H6:Viognier : 30 tonnes of 30 (30.23 delivered)
Expected Harvest Date 30/03/2011

HE11H2-4-B from Whitebox Vineyard
H2-4:Shiraz : 18 tonnes of 18 (17.5 delivered)
Expected Harvest Date 30/03/2011

HE11H2-3-A from Whitebox Vineyard
H2-3:Shiraz : 16 tonnes of 16 (16.17 delivered)
Expected Harvest Date 30/03/2011

HE11H6-C from Whitebox Vineyard
H6:Viognier : 0.5 tonnes of 30 (0.5 delivered)
Expected Harvest Date 30/03/2011

KR11BD-HILL-MV6-A from Killara Vineyard
BD-HILL-MV6:Pinot Noir : 35 tonnes of 35 (35 delivered)
Expected Harvest Date 30/03/2011

KR11MB-11-A from Killara Vineyard
MB-11:Pinot Noir : 21 tonnes of 21 (21.34 delivered)
Expected Harvest Date 31/03/2011

KR11MB-12-A from Killara Vineyard
MB-12:Pinot Noir : 18 tonnes of 18 (17.5 delivered)
Expected Harvest Date 31/03/2011

KR11MB-10-B from Killara Vineyard
MB-10:Pinot Noir : 23 tonnes of 23 (22.54 delivered)
Expected Harvest Date 31/03/2011

HE11H1-6-B from Whitebox Vineyard
H1-6:Shiraz : 13 tonnes of 13 (14.36 delivered)
Expected Harvest Date 1/04/2011

CV11CPINOT-A from Cahillton Vineyard
CPINOT:Pinot Noir : 18 tonnes of 18 (18.5 delivered)
Expected Harvest Date 4/04/2011

HE11H1-4-B from Whitebox Vineyard
H1-4:Shiraz : 15 tonnes of 15 (14.46 delivered)
Expected Harvest Date 4/04/2011

So far ....
18.16 Tonnes delivered

Fruit Receival Jobs

	Job	Date	Heading	Status	Vessels
▶	1661	28/03/2011	Process 18.00 tonnes of CSAVB	Complete	15D

● Jobs for this Parcel ○ Jobs for this Vin Def ○ All Fruit Receival Jobs

Figure 43. Weighbridge Screen

It is worthwhile at this point to reiterate the distinction between a planned parcel and a delivered parcel of fruit. These two entities are related to each other but must be considered distinct. Each time a parcel of fruit is delivered to the winery, it must be selected from one of the parcels in the expected list. The information from the parcel and associated vindef are used to seed the fruit reception job that will be generated when the “Generate Processing Job” push button is pressed.

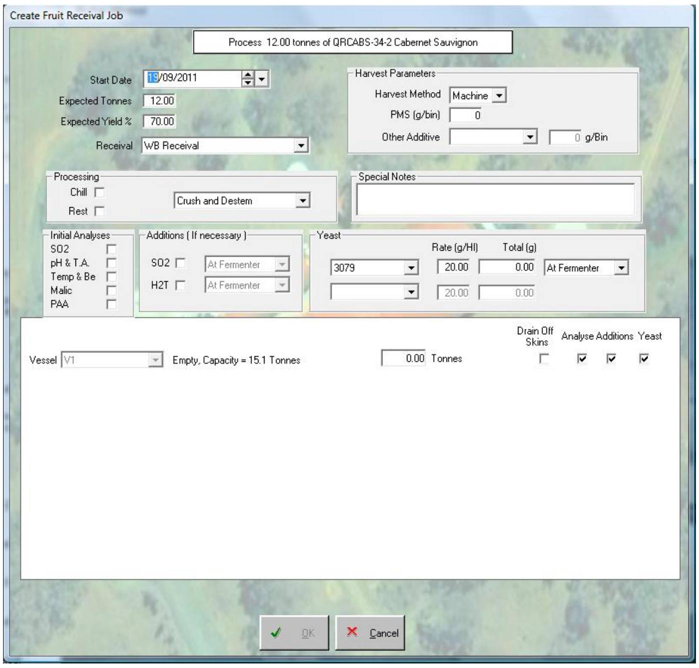
From there, the two entities become separated as any information about the reception of the fruit is stored in the fruit reception job and subsequently created “delivered parcel” object. This means that the plan is kept distinct from reality from the data model’s point of view. The user is free to change the amount of fruit delivered and the vessels that the fruit is processed into; the plan objects such as the vindef act only as seeds for data fields when the dialog appears.

Obviously, if the fruit has not been allocated to a vindef, or the vindef is incomplete and does not have vessel allocation stipulated then the fruit cannot be processed. Explanation messages such as these appear when such a situation occurs...

- No Vinification Definition for this parcel. Click here to go to the Vintage Plan to create one.
- The Vinification Definition for this parcel has no vessel allocations specified. Click here to go to the Vintage Plan to allocate vessel types.
- This parcel has not been allocated to vessels. Click here to go to Vintage Logistics to allocate this parcel to vessels.

13.3.3.1 Fruit Reception Job

When a parcel is selected that has not yet been delivered then the “Generate Processing Job” button is enabled, and when pressed, generates a dialog such as the one below...



Create Fruit Receival Job

Process 12.00 tonnes of QRCABS-34-2 Cabernet Sauvignon

Start Date 19/09/2011
Expected Tonnes 12.00
Expected Yield % 70.00
Receival WB Receival

Harvest Parameters
Harvest Method Machine
PMS (g/bin) 0
Other Additive 0 g/Bin

Processing
Chill □
Rest □
Crush and Destem

Special Notes

Initial Analyses
SO2 □
pH & T.A. □
Temp & Be □
Malic □
PAA □

Additions (If necessary)
SO2 □ At Fermenter
H2T □ At Fermenter

Yeast
3079 Rate (g/Hl) 20.00 Total (g) 0.00 At Fermenter
Rate (g/Hl) 20.00 Total (g) 0.00

Vessel V1 Empty, Capacity = 15.1 Tonnes 0.00 Tonnes Drain Off Skins □ Analyse ☑ Additions ☑ Yeast ☑

Figure 44. Fruit Reception Job Creation Dialog

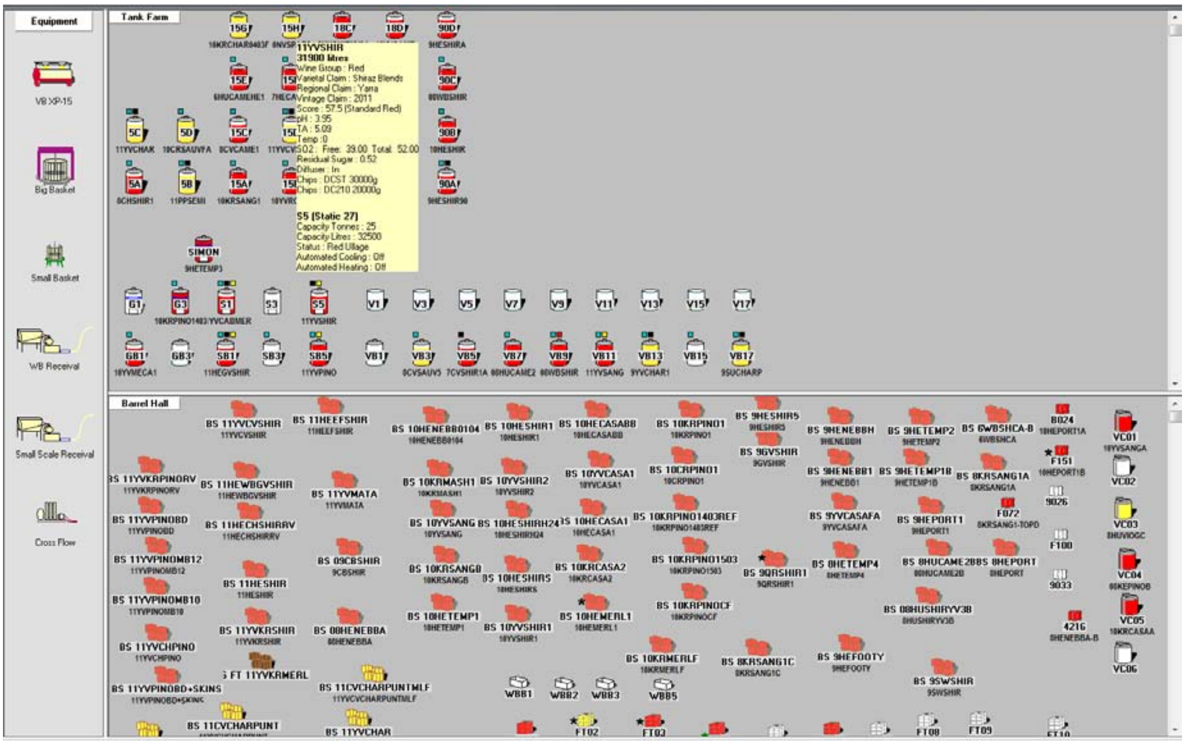
Several fields in this dialog are defaulted to values sourced from the vindef and associated structures for the parcel. In particular, the list of nominated vessels from the vintage logistics vessel allocation that the fruit can be apportioned to is displayed. It should be noted that all vessels are displayed even when they may have reached capacity according to the allocations specified in the original vindef. This is because there may be excess fruit and more may need to be added to some otherwise complete allocations. The various flags that appear to the right of each allocation allow the user to specify which allocations should receive any extra actions such as lab analyses, additions, or yeast and nutrient. This is necessary because some ferments may have to be made up of parts from different parcels and testing or inoculating the ferment before it has been assembled may be premature.

When completed and “OK’d” this dialog will generate the appropriate jobs, worknotes, fruit reception jobs, and fruit reception vessel allocation records. It also generates wine objects with “Proposed” status so that the wine when the tank map is displayed, the appropriate vessels will appear as containing the proposed wine.

13.3.4 Winery Map

The Winery Map (Figure 45) is the standard, “go-to” screen in the application for a winemaker. It has no specific purpose or agenda and serves as a general display for the winery and the wines therein. Each recognisable entity in the winery is represented as an icon on the screen. The various properties of these entities are displayed in tooltips as the mouse pointer moves over them. Interactions such as dragging and dropping will also initiate contextual functions but the majority of functions are initiated from a menu system by right-clicking on the icons. It has to be said that the design and function of this screen remains close to the original concept for the application. Since its inception and first instantiation as a working prototype, many new concepts have been surmised for this screen that extend or enhance this initial structure to enable a more intuitive interaction. Fortunately, none of these new ideas required any changes to be made to the structure of the data model, and so have not been implemented in the application because the primary aim of the interface is to utilise the data model, not provide a definitive user interface. All the functions and features of the data model available to the user are on display so nothing is missing apart from making the interface a little friendlier. If anything, the main problem has been too many new ideas and concepts appearing on a daily basis that could delay the completion of this project indefinitely.

The screen is divided into separate areas which group the entities within the winery into functional categories. The Equipment area lists the various pieces of processing equipment and displays them as icons. The Tank Farm and Barrel Hall areas display vessels and wines; the former being static, fixed vessels, and the latter containing movable vessels that can be grouped and moved independently. These areas are, of course, virtual and do not necessarily represent physical areas within the winery. However, the layout on the screen can be manipulated to represent these entities in a familiar or useful arrangement. The split between the Tank Farm and the Barrel Hall is movable to allow either area to be expanded. Each area scrolls independently.



11YVSHIR
31900 litres
Wine Group : Red
Varietal Claim : Shiraz Blends
Regional Claim : Yarra
Vintage Claim : 2011
Score : 57.5 (Standard Red)
pH : 3.95
TA : 5.09
Temp :0
SO2 : Free: 39.00 Total: 52.00
Residual Sugar : 0.52
Diffuser : In
Chips : DCST 30000g
Chips : DC210 20000g

S5 (Static 27)
Capacity Tonnes : 25
Capacity Litres : 32500
Status : Red Ullage
Automated Cooling : Off
Automated Heating : Off

Figure 45. Winery Map Screen

This screen can invoke many other functions via the menu system as well as drag and dropping interactions. These functions will often require that the screens need to be refreshed to reflect these changes. Often only one or two items on the screen needs to be refreshed, so in order to save processing time, the winery map window supplies a method that can be called by any other function that allows that function to specify which items will require refreshing, rather than have the whole screen refreshed each time. The "TurnOnRefreshFlag" method is commonly called by many functions within the realm of dialogs associated with the managed winery object.

13.3.4.1 Object Icons

Each icon displays the status of the object it represents by colour coding as well as pictorial representations. For example, a static tank is represented by an icon like this... However its appearance on the screen will change depending on its state and nature of wine in it.

Cleaned	Dirty	Rinsed	Pending	Disabled

Red Ferment	Red Must	Red Storage	Red Ullage	Skins
White Ferment	White Juice	White Storage	White Ullage	

Other vessel types include...

Barrique	Ganymede	Hogshead	Open
Pallecon	Storage	Variable Capacity	Picking Bin
Barrel Set			

So, each wine object tells the interface which icon is appropriate to display. A wine on ullage is calculated based on the capacity of the vessel versus the volume of the wine, with an ullage allowance dependent on the type of vessel. This is the *TypeUllageAllowance* property in the *VesselTypes* object. (*12.1.2.3.2 Vessel Types*). Each vessel icon displayed has flashing asterisks which notify the user that some extra important information is indicated for this vessel or wine. The black asterisk notifies that there are currently incomplete worknotes that apply to the vessel or wine. A green asterisk indicates that the wine has some readings that need to be ratified. A red asterisk will appear next to a barrel set when there is some discrepancy between the available capacity of the set and the volume of the wine assigned to that set. Each of these asterisks use properties from the vessel or wine object.

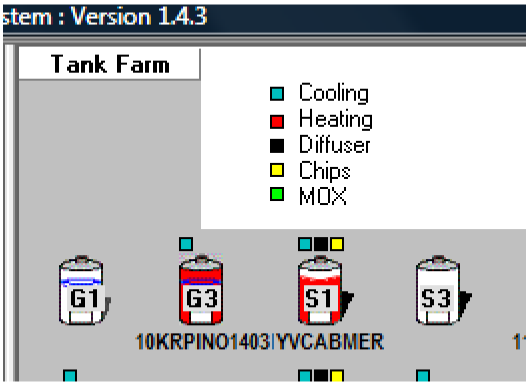
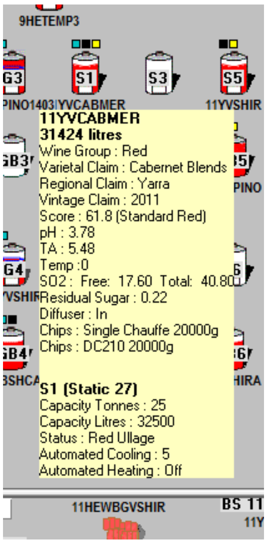


Figure 46. Tank Farm Legend Panel

In addition, vessels in the tank farm have coloured flags which indicate the status of various settings for the vessel. The legend, as shown in the figure above, appears when the mouse pointer passes over the “Tank Farm” label. In the example shown we see a wine in vessel “S1” which has cooling on, a diffuser in place (note the vessel is under ullage), and chips bags in the wine. The type of heating or cooling (manual or automated) depends on the vessel.



11YVCABMER
31424 litres
Wine Group : Red
Varietal Claim : Cabernet Blends
Regional Claim : Yarra
Vintage Claim : 2011
Score : 61.8 (Standard Red)
pH : 3.78
TA : 5.48
Temp :0
SO2 : Free: 17.60 Total: 40.80
Residual Sugar : 0.22
Diffuser : In
Chips : Single Chauffe 20000g
Chips : DC210 20000g

S1 (Static 27)
Capacity Tonnes : 25
Capacity Litres : 32500
Status : Red Ullage
Automated Cooling : 5
Automated Heating : Off

Figure 47. Tooltip example

Tooltips are the pale yellow boxes that appear when the mouse pointer passes over an object on the screen. In the example shown (Figure 47) we see a tooltip as it appears for the wine in the vessel "S1". The top half of the tooltip is dedicated to the wine details, and the bottom part, to the vessel. In this case we see details such as the name and volume of the wine, LIP claims for the wine, the monitoring score, a few basic parameters and details about the parameter flags such as chips in the wine. The vessel details include the capacity of the vessel, its status and any heating or cooling settings which are regarded a property of the vessel rather than the wine.

When an icon is clicked, it is flagged as selected, and the icon gets a purple background. (Figure 48)

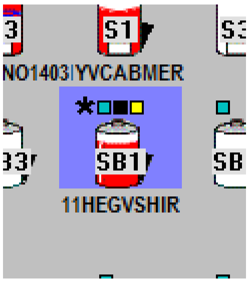


Figure 48. A Selected Vessel

The Windows standard for multiple selections from a list applies here too, so that by holding the Control button down as each successive item is selected, a series of vessels can be selected. This allows menu options that appear when an item is right-clicked to apply to all of the vessels selected rather than just one at a time.

13.3.4.2 Menus

The main way to interact with the system to create jobs and worknotes is through the use of popup menus that appear when objects on the screen are right-clicked. In general there are two main tests to determine whether an item in the menu is enabled.

1. The user needs to have the facility locked out.
2. The worknote type (if any) needs to be enabled.

In addition, any other tests will be cited with the item in the descriptions below.

There are four main menus that will appear depending on what the user right-clicks.

13.3.4.2.1 Equipment

If an item from the Equipment list is selected with a right-click, the menu that appears has the following structure...

Menu Item	**Description**	**Enabled If...**
Details...	Displays the Details dialog for the piece of equipment. Equivalent to double clicking on the item.	Always
Transfer Skins To...	Displays a dialog to allow the user to specify where the skins in the equipment item selected should be transferred.	Only if skins remain in the equipment.
Rinse	Generates a job to rinse the equipment.	If no other worknote applies.
Clean	Generates a job to clean the equipment.	If no other worknote applies.
Sanitise	Generates a job to sanitise the equipment which includes rinsing, cleaning, then treating with SO_2 or hot water or any other sanitising process according to the users discretion.	If no other worknote applies.

Table 15. Equipment Menu

13.3.4.2.2 Tank Farm

If the background of the tank farm is selected with a right-click, the menu that appears has the following structure...

Menu Item	Description	Enabled If...
Find a Vessel...	Displays the “Find a Vessel” dialog	Always
Find a Wine...	Displays the “Find a Wine” dialog	Always
Scum Run	Enables/Disables sub menu items.	Always
Scum Run – Create	Creates a scum run job	If no current scum run
Scum Run - Edit	Displays the Scum Run dialog to enter scum run information	If there is a current scum run
Scum Run - View Report	Displays the Scum Run job as a report	If there is a current scum run
Scum Run - Complete	Displays the Scum Run dialog to enter scum run information and complete the job.	If there is a current scum run
General Worknote	Displays the “General Worknote” dialog in “New” mode	Always
Maximise Tank Farm	Maximises the Tank Farm area of the screen	Always
Maximise Barrel Hall	Maximises the Barrel Hall area of the screen	Always

Table 16. Tank Farm Menu

13.3.4.2.3 Barrel Hall

If the background of the Barrel Hall is selected with a right-click, the menu that appears has the following structure...

Menu Item	Description	Enabled If...
Arrange Barrel Sets...	Displays the “Arrange Barrel Sets dialog to manipulate vessels and barrel set membership.	Always
Find a Vessel...	Displays the “Find a Vessel” dialog	Always
Find a Wine...	Displays the “Find a Wine” dialog	Always
Lockdown	Toggle to enable and disable the movement of vessels within the Barrel Hall area.	Always

Maximise Tank Farm	Maximises the Tank Farm area of the screen	Always
Maximise Barrel Hall	Maximises the Barrel Hall area of the screen	Always

Table 17. Barrel Hall Menu

13.3.4.2.4 VESSELS

Right-clicking on a vessel on the Winery Map generates a popup menu that is quite large and complex. For simplicity's sake, this menu has been broken down and presented below in layers.

Top Layer

Menu Item	**Description**	**Enabled If...**
Comments ...	Displays the "Comments" dialog.	User has winery locked. Only one vessel selected. Vessel contains wine.
Details...	Displays the "Details" dialog.	Single vessel selected
Tracker...	Navigates the user to the Tracker Screen, and selects the wine from the list on the Tracker.	User has winery locked. Only one vessel selected. Vessel contains wine
Create Tank Worknotes	Displays the Tank Worknotes menu	User has winery locked. Only empty vessel(s) selected.
Create Wine Worknotes...	Displays the Wine Worknotes menu	User has winery locked. Vessel(s) contains wine
Blend Calculator...	Displays the Blending Calculator tool	Always
Transfer Skins	Displays the "Transfer" Skins Dialog in "New" mode	User has winery locked. Single Vessel contains skins but no wine
Chips	Displays the Chips menu	User has winery locked. Vessel(s) contains wine and chips
Temperature Control...	Displays the Temperature Control dialog.	User has winery locked. Vessel(s) have manual or automated temperature control features.
Diffuser	Displays the Diffuser menu	User has winery locked.

MicroOxygenation	Displays the MOX menu	User has winery locked
Rename Wine...	Displays the “Rename Wine” dialog.	User has winery locked. Only one vessel selected. Vessel contains wine
Rename Barrel Set...	Displays the “Rename Barrel Set” dialog.	User has winery locked. Only one barrel set selected.
Reset Wine Volume	Displays the “Reset Wine Volume” dialog.	User has winery locked. Super Administrator. Single Vessel contains wine.
Wine Settings...	Displays the “Wine Settings” dialog.	
Disable	Disables the vessel from use	User has winery locked. Super Administrator. Vessel(s) are empty and enabled
Enable	Enables the vessel(s) for use	User has winery locked. Super Administrator. Vessel(s) are empty and disabled.
Create a Wine...	Displays the “Create a Wine” dialog.	User has winery locked. Super Administrator. Vessel(s) are empty and disabled.

Table 18. Vessels Menu

13.3.4.2.5 Create Tank Worknotes

This submenu appears from the “Create Tank Worknotes” menu. The options only apply to empty vessels.

Menu Item	**Description**	**Enabled If...**
Rinse Tank	Displays the “Generate Vessel Jobs” dialog in “New” mode with the rinse tank job and vessel(s) preselected	User has winery locked. Vessel(s) are empty
Clean Tank	Displays the “Generate Vessel Jobs” dialog in “New” mode with the clean tank job and vessel(s) preselected	User has winery locked. Vessel(s) are empty
Sanitise Tank	Displays the “Generate Vessel	User has winery locked.

	Jobs" dialog in "New" mode with the sanitise tank job and vessel(s) preselected	Vessel(s) are empty
Move Vessel(s) to Other Winery	Displays the "Move Vessels" dialog to allow the user to move vessels to other wineries.	User has winery locked. Vessel(s) are empty and in Barrel Hall

Table 19. Create Tank Worknotes Menu

13.3.4.2.6 Create Wine Worknotes

This submenu (Table 20) is quite complex, and no doubt offers some room for improvement in any subsequent development of the model. It is divided into three main submenus; Additions, Processes, and Laboratory, which divide the tasks that can be applied to the wine into logical sections.

The Additions menu simply sets up pre-selections for the generic wine addition dialog. Depending on the vessels/wines selected from the winery map, the addition options are defaulted to the default value for the type of addition. The only exception to this is the Multiple option, which goes to a pre-selection dialog first, to allow the addition type(s) to be selected.

Menu Item	**Description**	**Enabled If...**
Multiple	Displays the Multiple Addition pre-selection dialog.	User has winery locked. Vessel(s) contains wine
Acid...	Displays the "Wine Additions" dialog in "New" mode with entries for each of the vessels defaulted to Acid adds	User has winery locked. Vessel(s) contains wine
Additives...	Displays the "Wine Additions" dialog in "New" mode with entries for each of the vessels defaulted to so-called "Other Additives" adds	User has winery locked. Vessel(s) contains wine
Fining...	Displays the "Wine Additions" dialog in "New" mode with entries for each of the vessels defaulted to Fining Agents additions	User has winery locked. Vessel(s) contains wine
Grape Concentrate...	Displays the "Wine Additions" dialog in "New" mode with entries for each of the vessels defaulted to Grape Concentrate	User has winery locked. Vessel(s) contains wine

	additions	
MLF Bacteria...	Displays the “Wine Additions” dialog in “New” mode with entries for each of the vessels defaulted to Malolactic Bacteria additions	User has winery locked. Vessel(s) contains wine
Nutrient...	Displays the “Wine Additions” dialog in “New” mode with entries for each of the vessels defaulted to Nutrient adds	User has winery locked. Vessel(s) contains wine
SO_2...	Displays the “Wine Additions” dialog in “New” mode with entries for each of the vessels defaulted to Sulphur Dioxide additions.	User has winery locked. Vessel(s) contains wine
Yeast...	Displays the “Wine Additions” dialog in “New” mode with entries for each of the vessels defaulted to Yeast additions	User has winery locked. Vessel(s) contains wine

Table 20. Create Wine Worknotes Menu

The Processes submenu is in two parts, divided by a menu divider. (Figure 49).

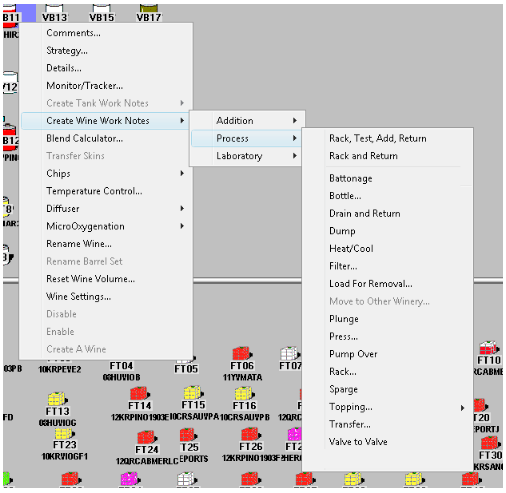


Figure 49. Process Menu

The reason for the divider is that it was decided to build some composite process jobs as well as supply just basic process jobs and these needed to be clearly separated. In this context, a composite job is a job that consists of several sequenced worknotes of different types that help achieve the overall aim of the job. The same result could be achieved by creating individual jobs with individual worknotes, but the overall aim might be lost because it was not explicitly stated in any of the job titles. It is expected that many more composite jobs will be required, perhaps providing even more incentive to redesign this particular aspect of the user interface in the future.

Menu Item	**Description**	**Enabled If...**
Rack, Test, Add, Return...	Loads the “Rack, Test , Add, Return” dialog in “New” mode	User has winery locked. Single selected Vessel contains wine
Rack and Return...	Loads the “Rack and Return” dialog in “New” mode	User has winery locked. Single selected Vessel contains wine
Battonage	Loads the “Create Wines Job” dialog in “New” mode with the selected vessels and Battonage job pre-selected	User has winery locked. Selected Vessel(s) contain wine
Bottle...	Loads the “Bottle Wine” dialog in “New” mode.	User has winery locked. Single selected Vessel contains wine
Drain and Return	Loads the “Create Wines Job” dialog in “New” mode with the selected vessels and Drain and Return job pre-selected	User has winery locked. Selected Vessel(s) contain wine
Dump	Loads the “Create Wines Job” dialog in “New” mode with the selected vessels and Dump Wine job pre-selected	User has winery locked. Selected Vessel(s) contain wine
Heat/Cool...	Loads the “Heat/Cool a Wine” dialog in “New” mode	User has winery locked. Selected Vessel(s) contain wine
Filter...	Loads the “Transfer” dialog in “New” mode with the selected vessels preselected with the mode set to Filter	User has winery locked. Selected Vessel(s) contain wine
Load for Removal	Loads the “Create Wines Job” dialog in “New” mode with the selected vessels and Load for Removal WIne job pre-selected	User has winery locked. Selected Vessel(s) contain wine
Move Wine to Other Winery...	Displays the “Move Wine to Other Winery” dialog	User has winery locked. Single selected vessel from the Barrel

		Hall only contains wine
Plunge	Loads the “Create Wines Job” dialog in “New” mode with the selected vessels and Plunge job pre-selected	User has winery locked. Selected Vessel(s) contain wine
Press	Loads the “Press Wines” dialog in “New” mode	User has winery locked. Selected Vessel(s) contain on-skins wine.
Pump Over	Loads the “Create Wines Job” dialog in “New” mode with the selected vessels and Pump-Over job pre-selected	User has winery locked. Selected Vessel(s) contain wine
Rack	Loads the “Transfer” dialog in “New” mode with the selected vessels preselected with the mode set to Rack	User has winery locked. Selected Vessel(s) contain wine
Sparge	Loads the “Create Wines Job” dialog in “New” mode with the selected vessels and Sparge job pre-selected	User has winery locked. Selected Vessel(s) contain wine
Topping...	Displays the Topping Dialog in “New” mode	User has winery locked. Selected Vessel(s) contain wine
Transfer...	Loads the “Transfer” dialog in “New” mode with the selected vessels preselected with the mode set to Transfer	User has winery locked. Selected Vessel(s) contain wine
Valve to Valve	Loads the “Create Wines Job” dialog in “New” mode with the selected vessels and Valve-to-Valve job pre-selected	User has winery locked. Selected Vessel(s) contain wine

Table 21. Create Wine Worknotes Process Menu

The Laboratory Menu under the Create Wine Worknotes menu is similar to the Processes menu in that it has space reserved at the top for Multiple and composite jobs involving laboratory worknotes as well as the simple jobs themselves. (Figure 50)

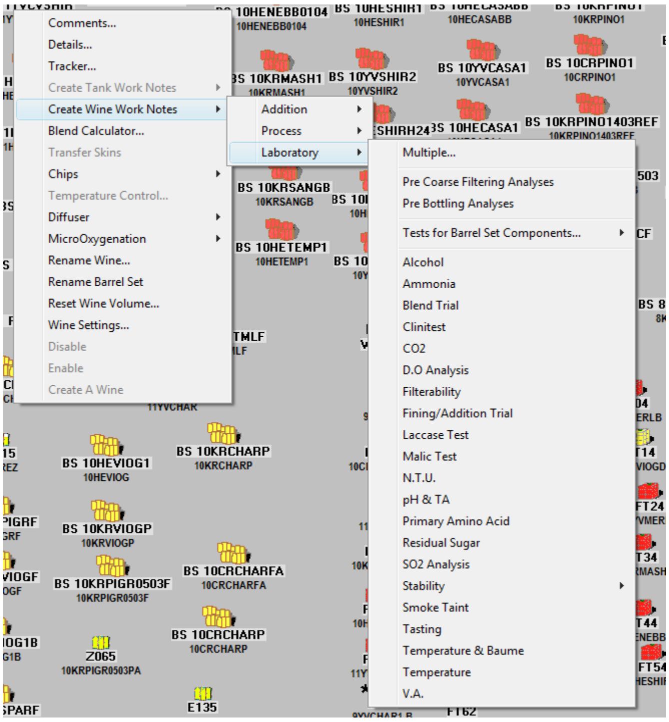


Figure 50. Laboratory Analyses menu

Menu Item	Description	Enabled If...
Multiple...	Displays the “Multiple Wine Analysis” dialog in “New” mode.	User has winery locked. Vessel(s) contains wine
Pre Coarse Filtering Analyses	Displays the Pre-Bottling dialog in New mode and with the Pre coarse filtering analyses job type set	User has winery locked. Vessel(s) contains wine.
Pre Bottling Analyses	Displays the Pre-Bottling dialog in New mode and with the Pre bottling analyses job type set	User has winery locked. Vessel(s) contains wine.
Tests for Barrel Set Components...	Displays the Barrel set components sub menu which consists of ... Clinitest, Malic Test, pH &TA, and Residual Sugar. All proceeing for these options are the same as those for whole wines/vessels	User has winery locked. Barrel Set (only) contains wine.

Table 22. Laboratory Menu

All remaining menu options cause the “Wine Jobs” dialog to be displayed in “New” mode with the appropriate wines and jobs preselected for the user. As usual, these options are only available if the user has winery locked and the vessel contains wine.

13.3.4.2.7 Chips

The treatment of chips added to wines is treated differently within the model as well as within the interface because the nature of the addition is fundamentally different, and based on a time exposure as well as amount added. The Chips sub-menu has options to add or remove chips to the wine. The Add option is always enabled, whereas the Remove option is only enabled if the wine currently has chips in it. The Add chips option also differs in that it causes the Add Chips dialog to be displayed which allows the user to select the amount and type of chips to be added, whereas the remove option simply generates the remove chips job, and does not discriminate between batches of chips to be removed if there is more than one batch present in the wine.

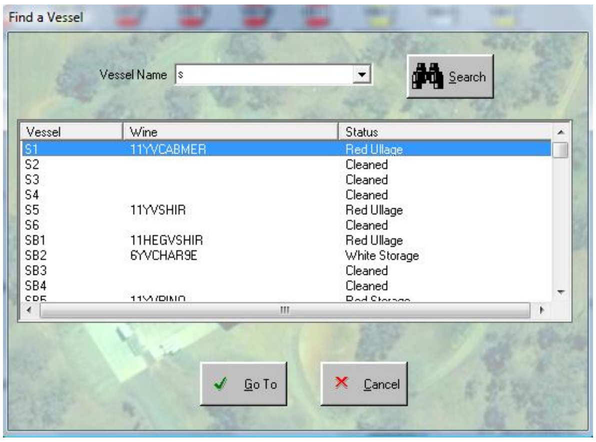
13.3.4.2.8 Diffuser

Similarly, the addition or removal of a diffuser is not regarded in the same way that a material addition to a wine is treated because the addition is to the process or treatment of the wine, not the wine. There are two simple menu options that appear; to put a diffuser into the wine, or to remove it. The former is only enabled if the wine currently has no diffuser in it and vice-versa for the latter.

13.3.4.2.9 Micro-Oxygenation

The process of micro-oxygenation is defined as a slow ingress of molecular oxygen into the wine at a specified rate, over a specified time. For this reason, it too is treated separately from the material additions that can be made to a wine. The menu options are to add, remove, or change the rate of MOX to the wine.

13.3.4.3 Find a Vessel



Vessel Name: s

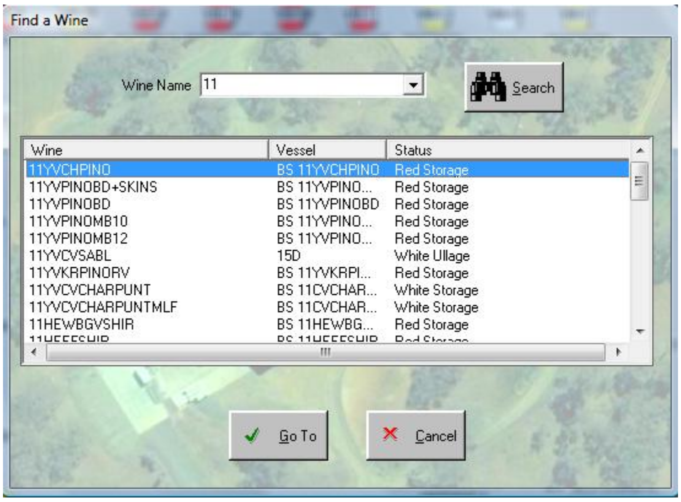
Vessel	Wine	Status
S1	11YVCABMER	Red Ullage
S2		Cleaned
S3		Cleaned
S4		Cleaned
S5	11YVSHIR	Red Ullage
S6		Cleaned
SB1	11HEGVSHIR	Red Ullage
SB2	6YVCHAR9E	White Storage
SB3		Cleaned
SB4		Cleaned
SB5	11YVPINO	Red Storage

Figure 51. Find A Vessel Dialog

Because vessels within the barrel hall are dynamic in position and membership of barrel sets, it is necessary to provide a function to find a vessel within the Winery Map screen. The dialog (Figure 51) allows the user to enter a string of text that is then used to compare to the names of the vessels defined in the winery. The comparison is case insensitive and can be from any part of the name of the vessel. The Vessel Name field is also a drop-down combo-box which lists all the vessels anyway so it acts as a text and selection field. When this field changes, the “Search” button become the default action so the user need only press enter to start the search process. The List field below will then list all the matching vessels, along with the wine it contains (if any) and the status of the vessel. The list includes vessels that are part of barrel sets, whose names match the criteria specified for the search. The search process simply loops through all the vessels in the winery object and looks for the text string specified in the Vessel Name field, anywhere in the name of the vessel, regardless of the case of the text. The “Go To” button becomes the default action when the list is populated and will dismiss the dialog and highlight the vessel selected from the list on the Winery Map.

13.3.4.4 Find a Wine

The “Find a Wine” dialog (Figure 52) is a clone of the “Find a Vessel” dialog with the only modification being the emphasis of the wine name instead of the vessel name in the search and presentation of the dialog. Internally, the search is a little different in that the search process must loop through all the wines in the wines collection of the winery object and first determine whether the wine is a current wine, then extract the name of the wine from the readings for that wine, and search that for any matching occurrences of the search string in the name. Similarly, the Wine Name field is a drop-down combo box, initially populated with the names of all the current wines defined in the winery which also requires this more laborious computation to build.



Wine Name: 11

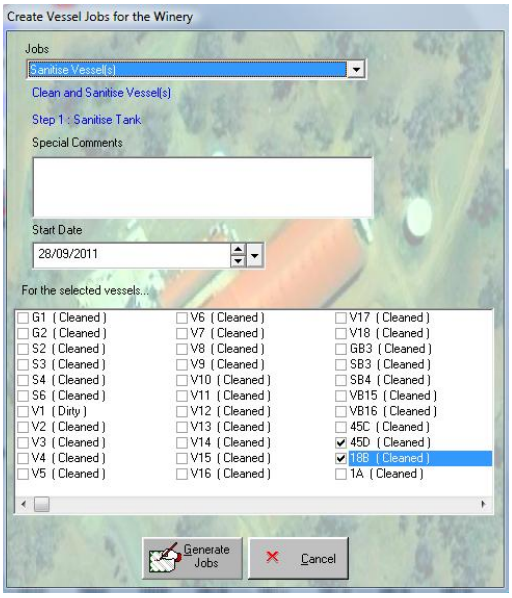
Wine	Vessel	Status
11YVCHPINO	BS 11YVCHPINO	Red Storage
11YVPINOBD+SKINS	BS 11YVPINO...	Red Storage
11YVPINOBD	BS 11YVPINOBD	Red Storage
11YVPINOMB10	BS 11YVPINO...	Red Storage
11YVPINOMB12	BS 11YVPINO...	Red Storage
11YVCVSABL	15D	White Ullage
11YVKRPINORV	BS 11YVKRPI...	Red Storage
11YVCVCHARPUNT	BS 11CVCHAR...	White Storage
11YVCVCHARPUNTMLF	BS 11CVCHAR...	White Storage
11HEWBGVSHIR	BS 11HEWBG...	Red Storage
11HEFFSHIR	BS 11HEFFSHIR	Red Storage

Figure 52. Find a Wine dialog

13.3.4.5 Generate Job for Vessel

This is a generic job creating dialog for jobs specific to vessels. (Figure 53) The list at the top is populated with job types that...

1. Apply to a winery facility type
2. Are flagged as “Generally Available”
3. Have the WorknoteTargetObjectType property set to vessel object types.
4. Are enabled.



Create Vessel Jobs for the Winery

Jobs: Sanitise Vessel(s)

Clean and Sanitise Vessel(s)

Step 1 : Sanitise Tank

Special Comments

Start Date: 28/09/2011

For the selected vessels...

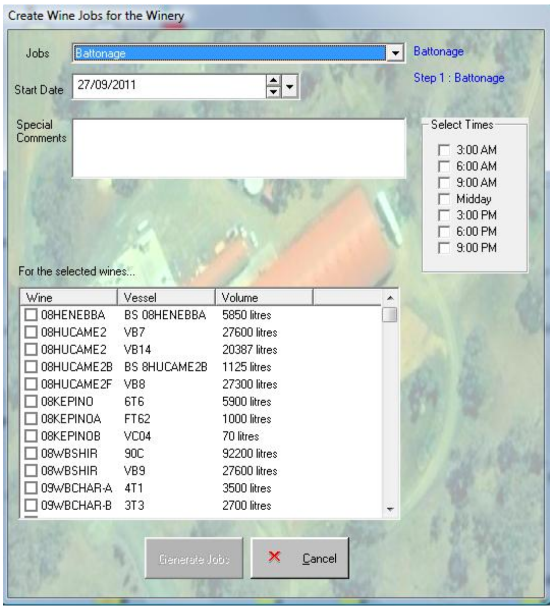
□ G1 (Cleaned)
□ G2 (Cleaned)
□ S2 (Cleaned)
□ S3 (Cleaned)
□ S4 (Cleaned)
□ S6 (Cleaned)
□ V1 (Dirty)
□ V2 (Cleaned)
□ V3 (Cleaned)
□ V4 (Cleaned)
□ V5 (Cleaned)
□ V6 (Cleaned)
□ V7 (Cleaned)
□ V8 (Cleaned)
□ V9 (Cleaned)
□ V10 (Cleaned)
□ V11 (Cleaned)
□ V12 (Cleaned)
□ V13 (Cleaned)
□ V14 (Cleaned)
□ V15 (Cleaned)
□ V16 (Cleaned)
□ V17 (Cleaned)
□ V18 (Cleaned)
□ GB3 (Cleaned)
□ SB3 (Cleaned)
□ SB4 (Cleaned)
□ VB15 (Cleaned)
□ VB16 (Cleaned)
□ 45C (Cleaned)
☑ 45D (Cleaned)
☑ 18B (Cleaned)
□ 1A (Cleaned)

Figure 53. Vessel Job dialog

These criteria ensure that only jobs that do not require any further specifications are displayed. The dialog can be invoked in either “New” or “Edit” mode, which causes the behaviour of the dialog to change accordingly. In “New” mode, the initial settings are based on the selection of vessels and the type of job that needs to be created. Also, the OK button appears as “Generate Jobs”, and when pressed, a new job and worknotes are created. In “Edit” mode, the dialog is also given the ID of the job that is being edited, and so can initialise itself with the details of the job. The “OK” button remains as “OK”. When Generate Jobs is pressed, a worknote is created for each jobtype-vessel combination. When OK is pressed the application needs to check whether it needs to create new or edit or delete existing worknotes to achieve the appropriate result. As can be seen from the image of the dialog, the user is also able to add special comments to the job and specify when the job should be started; both properties of the job object.

13.3.4.6 Create Wine Jobs

This dialog is similar in purpose and behaviour to the Create Vessel Jobs dialog described in the previous section. (Figure 54) The list at the top of the dialog contains job types that fit similar criteria to those for the vessel jobs, except that the WorknoteTargetObjectType needs to be for wine object types. The list of wines is in the “Details” format rather than a multi-columned list so that it can be sorted; given the fact that the names of the wines are user defined rather than static vessel names. Some jobs from this list can potentially have worknotes that require times to be set for their implementation, and so the “Select Times” panel will appear when such jobs are selected from the list. Once again, this dialog can be invoked in “New” or “Edit” mode, and the same specifications exist as far as processing when the Generate Jobs and OK buttons are pressed.



Create Wine Jobs for the Winery

Jobs: Battonage

Battonage

Step 1 : Battonage

Start Date: 27/09/2011

Special Comments:

Select Times

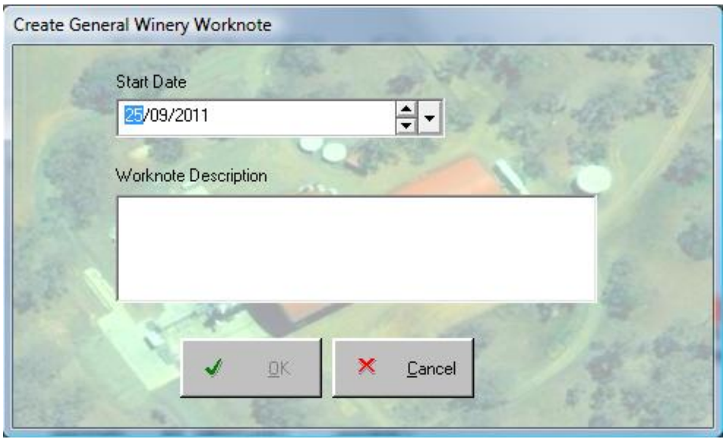
- □ 3:00 AM
- □ 6:00 AM
- □ 9:00 AM
- □ Midday
- □ 3:00 PM
- □ 6:00 PM
- □ 9:00 PM

For the selected wines...

Wine	Vessel	Volume
□ 08HENEBBA	BS 08HENEBBA	5850 litres
□ 08HUCAME2	VB7	27600 litres
□ 08HUCAME2	VB14	20387 litres
□ 08HUCAME2B	BS 8HUCAME2B	1125 litres
□ 08HUCAME2F	VB8	27300 litres
□ 08KEPINO	6T6	5900 litres
□ 08KEPINOA	FT62	1000 litres
□ 08KEPINOB	VC04	70 litres
□ 08WBSHIR	90C	92200 litres
□ 08WBSHIR	VB9	27600 litres
□ 09WBCHAR-A	4T1	3500 litres
□ 09WBCHAR-B	3T3	2700 litres

Figure 54. Wine Job Dialog

13.3.4.7 General Worknote



Create General Winery Worknote

Start Date: 25/09/2011

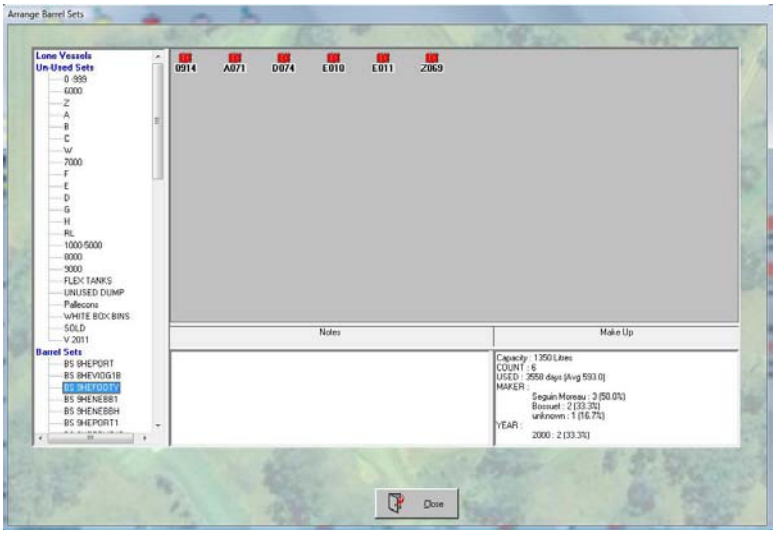
Worknote Description:

Figure 55. General Job Dialog

The need for a general worknote came about because it was deemed useful and beneficial to inform staff of work requirements in a consistent format, (i.e., the job would be printed as a report from the system with the familiar structure, letter-head and format). Originally, these jobs were deemed un-necessary because they served no purpose from the data modelling perspective; however they have been included to provide this functionality even though it cannot contribute to the data model's functionality. (Figure 55)

13.3.4.8 Barrel Set Management

The concept of barrel sets has been discussed earlier. (*9.1.2.4 Barrel Sets*). However, there was one issue that was left unresolved as far as ensuring that the users understood the concept, and the distinction between moving vessels and moving wine. It is hoped that this has been overcome by designing the dialog for management of barrels within barrel sets by employing an interface design that was deliberately cloned from the common Windows Explorer format of a tree control on the left representing the barrel sets (analogous to folders) and the display of the individual barrels to the right (analogous to files).



Capacity : 1350 Litres
COUNT : 6
USED : 3558 days (Avg 593.0)
MAKER :
Seguin Moreau : 3 (50.0%)
Bossuet : 2 (33.3%)
unknown : 1 (16.7%)
YEAR :
2000 : 2 (33.3%)

Figure 56. Arrange Barrel Sets Dialog

The dialog (Figure 56) was designed to allow barrels to be drag-and-dropped from one barrel set to another. The rules from *11.4 Barrel Sets* are easily administered by displaying messages when invalid actions are attempted. The movement of vessel from barrel set to barrel set is achieved by simply updating the BarrelSet property for the vessel. There are three main nodes in the barrel set tree...

"Lone Vessels" is used to contain all the vessels in the barrel hall that are used but are not members of a barrel set.

“Un-used Sets” contains the barrel sets that are flagged as un-used, which means they are not displayed in the Barrel Hall window, or available for selection in the rest of the system.

“Barrel Sets” obviously contains all the barrels sets that are defined in the system, that are used or at least that we want to have visible outside of this dialog.

There are two popup menus that can be invoked from the barrel set tree. If the user right-clicks on the background of the tree, a “New Barrel Set” menu appears, which will create a new barrel set, gives it the default name of “New Barrel Set” and displays the “Rename Barrel Set” dialog (below).

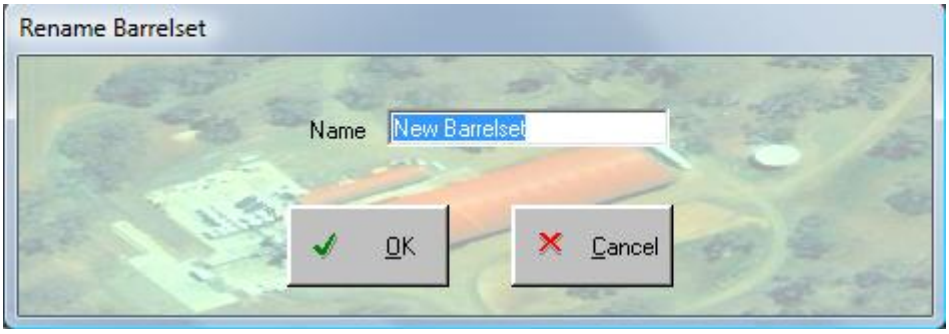


Figure 57. Rename Barrel Set Dialog

The second popup menu appears when a barrel set is right-clicked. This menu has three options...

- “Un-used Set” which is a check menu option to change the used/un-used toggle for the barrel set. This can only be changed for barrel sets that do not contain wine.
- “Rename” uses the rename dialog as shown above, to allow the user to change the name of a barrel set.
- “Delete” causes the barrel set to be tagged as Deleted which stops it from appearing in this list, but still allows the barrel set to appear in the context of wine tracking, and also ensures data integrity for historic data concerning barrel set membership for vessels. This cannot be done on barrel sets that contain wine.

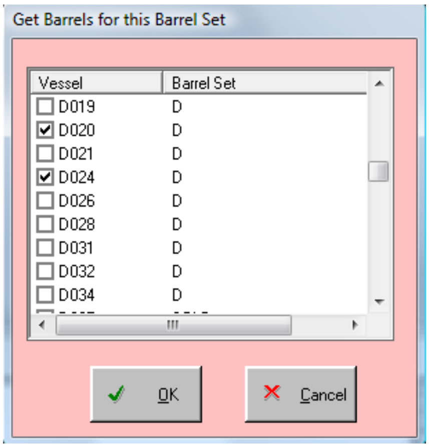
When a barrel set is selected from the tree, its details are displayed on the area to the right, and summarised in the “Make Up” panel below, as well as any notes that may have been made about the set. The Notes section is automatically updated as text is entered. The makeup summary utilises the BarrelMakeUp property of the barrel set object.

Each individual vessel that belongs to the barrel set is displayed as an icon in the display area in the same way that it would appear if it were a lone vessel in the barrel hall. Right clicking on an individual vessel displays a checked popup menu that allows the user to set the “UllageInBarrelSet” property for that vessel. This is taken into consideration when the computations are made in the BarrelSetNeedsAttention procedure that works out whether the barrel set may need attention to reconcile the specified volume of the wine versus the calculated capacity of the barrel set. Any message generated by this procedure, which is invoked as the barrel set list is generated, is displayed as a child node of the barrel set in the tree.

This dialog also allows the selection of multiple barrels from the barrels panel and the use of Ctrl-C and Ctrl-V (Copy and Paste) to move them from one barrel set to another.

There are also two options in popup menu form that appear when the background for the barrel set display area is right-clicked.

"Move Barrel s Here..." displays the following dialog.

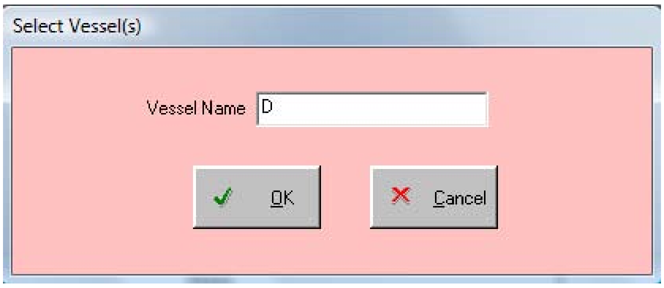


Vessel	Barrel Set
□ D019	D
☑ D020	D
□ D021	D
☑ D024	D
□ D026	D
□ D028	D
□ D031	D
□ D032	D
□ D034	D

Figure 58. Get Barrels Dialog

This dialog displays a checked list box of all the unused vessels defined in the system, allowing the user to grab vessels from any barrel set and move them to the selected barrel set without having to search or navigate to the source barrel set.

"Select" displays this dialog which allows the user to enter a character or string that is compared to



Vessel Name D

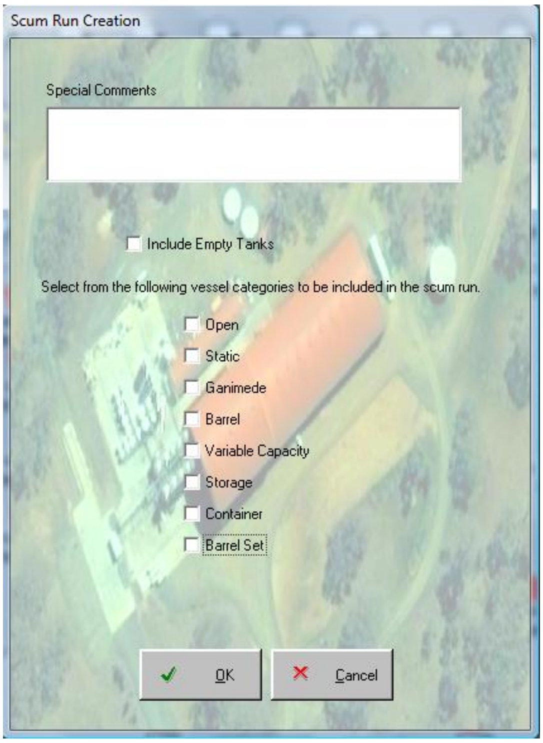
Figure 59. Select Vessels Dialog

the names of all the vessels in the set and selects those contain that string.

13.3.4.9 Scum Run

The concept of a scum run is not an original idea. The name is often given a less colloquial title in other settings, but is nonetheless best described as a periodic check of all the wines and/or vessels in the winery to ensure that they are in a proper condition. Within the context of this system, a scum run also represents an opportunity to check that the system is up to date with what is going on in the winery; a chance to reconcile the two.

The infrastructure within the model has already been described, so it is a simple matter, here, to describe how it is used. As was noted in the Jobs collection object, the system allows only one scum run job to be active at any one time. When a scum run is created from the `Tank farm” background popup menu, the first dialog to appear is the “Scum Run Creation” dialog. (below)



Scum Run Creation

Special Comments

□ Include Empty Tanks

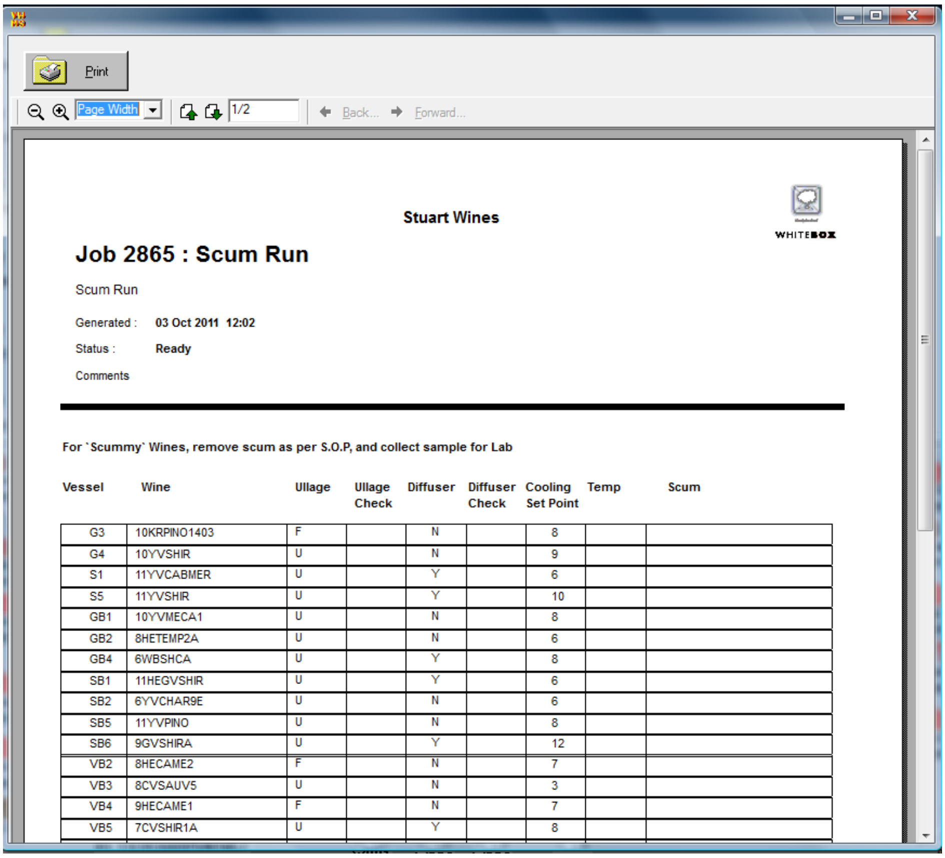
Select from the following vessel categories to be included in the scum run.

□ Open
□ Static
□ Ganimede
□ Barrel
□ Variable Capacity
□ Storage
□ Container
□ Barrel Set

Figure 60. Scum Run Pre-selection Dialog

This allows the user to specify which vessels need to be included in the job. When the OK button is pressed, the job is created along with individual worknotes for each of the vessels that fit the criteria as specified in the dialog. Needless to say, this can mean hundreds of worknotes in some enterprises, depending on the options they select.

The scum run can then be presented to the user in two forms; the printable version (Figure 61) which can be given to the cellar staff to actually perform the run, and the editable version (Figure 62) which allows the user to input the results of the run.



Stuart Wines

Job 2865 : Scum Run

Scum Run

Generated : 03 Oct 2011 12:02

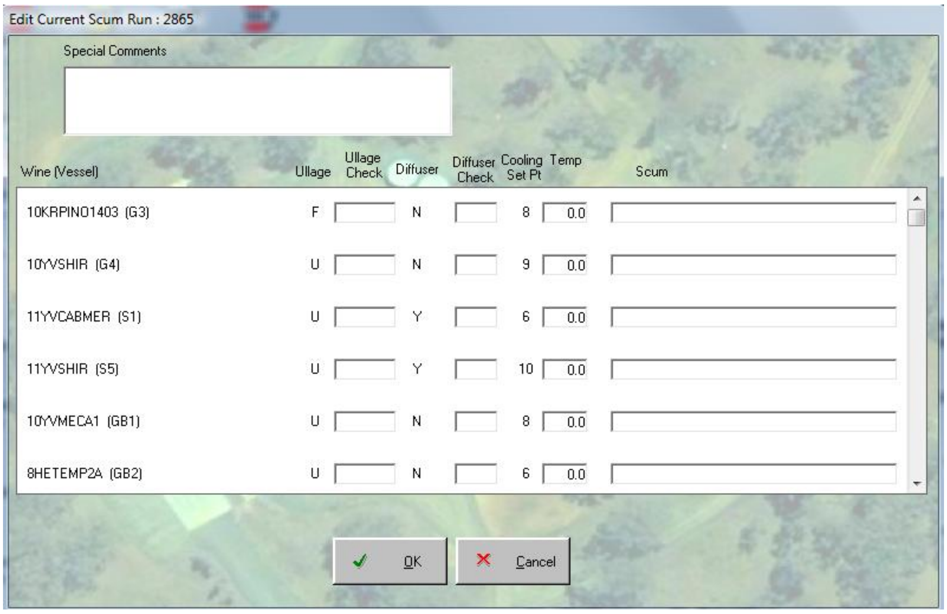
Status : Ready

Comments

For `Scummy` Wines, remove scum as per S.O.P, and collect sample for Lab

Vessel	Wine	Ullage	Ullage Check	Diffuser	Diffuser Check	Cooling Set Point	Temp	Scum
G3	10KRPINO1403	F		N		8		
G4	10YVSHIR	U		N		9		
S1	11YVCABMER	U		Y		6		
S5	11YVSHIR	U		Y		10		
GB1	10YVMECA1	U		N		8		
GB2	8HETEMP2A	U		N		6		
GB4	6WBSHCA	U		Y		8		
SB1	11HEGVSHIR	U		Y		6		
SB2	6YVCHAR9E	U		N		6		
SB5	11YVPINO	U		N		8		
SB6	9GVSHIRA	U		Y		12		
VB2	8HECAME2	F		N		7		
VB3	8CVSAUV5	U		N		3		
VB4	9HECAME1	F		N		7		
VB5	7CVSHIR1A	U		Y		8		

Figure 61. Scum Run Report



Edit Current Scum Run : 2865

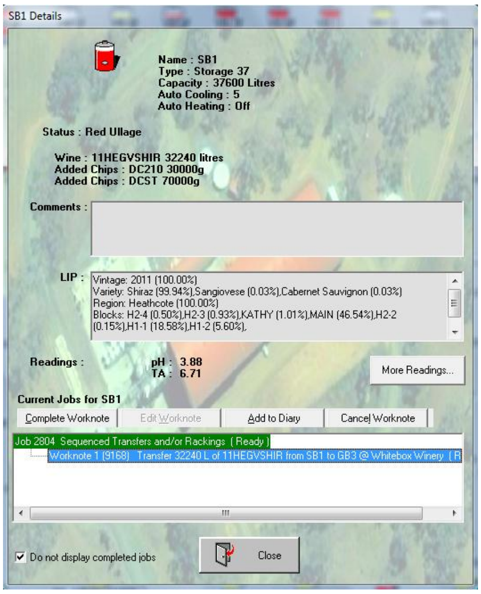
Special Comments

Wine (Vessel)	Ullage	Ullage Check	Diffuser	Diffuser Check	Cooling Set Pt	Temp	Scum
10KRPINO1403 (G3)	F		N		8	0.0	
10YVSHIR (G4)	U		N		9	0.0	
11YVCABMER (S1)	U		Y		6	0.0	
11YVSHIR (S5)	U		Y		10	0.0	
10YVMECA1 (GB1)	U		N		8	0.0	
8HETEMP2A (GB2)	U		N		6	0.0	

Figure 62. Scum Run Dialog

Details for the current settings are recorded in the Parameters section of the worknote structure upon creation so that when the report is printed for the cellar staff to use, or presented for editing in the edit dialog, it indicates what the system currently has recorded for the vessel. The fields in the scum run are, unfortunately, over-tailored for the purposes of the winery used for testing of the system. For example, the Ullage value for a vessel utilises a code system favoured by one particular winemaker but not by most others: (F = Full, OF = Overfull, U = Ullage, X = number of litres below full, etc.). However, the overall philosophy of the function is sound and has proven to be very useful in highlighting some oversights in the cellar. There is no doubt that this needs to be more generic, but without more examples of what might be required, this structure had to suffice.

13.3.4.10 DETAILS



SB1 Details

Name : SB1
Type : Storage 37
Capacity : 37600 Litres
Auto Cooling : 5
Auto Heating : Off

Status : Red Ullage

Wine : 11HEGVSHIR 32240 litres
Added Chips : DC210 30000g
Added Chips : DCST 70000g

Comments :

LIP : Vintage: 2011 (100.00%)
Variety: Shiraz (99.94%),Sangiovese (0.03%),Cabernet Sauvignon (0.03%)
Region: Heathcote (100.00%)
Blocks: H2-4 (0.50%),H2-3 (0.93%),KATHY (1.01%),MAIN (46.54%),H2-2 (0.15%),H1-1 (18.58%),H1-2 (5.60%),

Readings : pH : 3.88
TA : 6.71

Current Jobs for SB1

Job 2804 Sequenced Transfers and/or Rackings (Ready)
Worknote 1 (9168) Transfer 32240 L of 11HEGVSHIR from SB1 to GB3 @ Whitebox Winery (R

☑ Do not display completed jobs

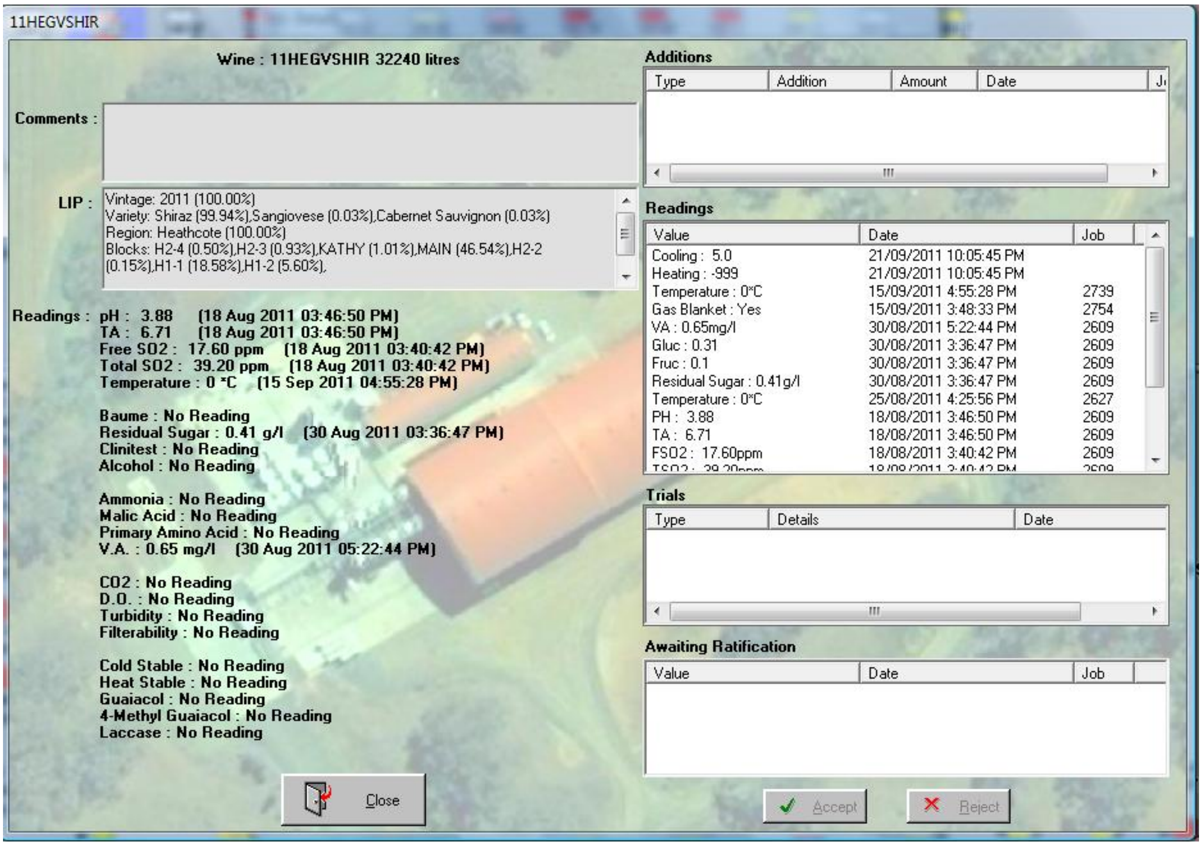
Figure 63. Vessel Details Dialog

When the user double clicks on an item in the Winery Map, a details screen (Figure 63) will appear for that object. This contains more information than can be described in a tooltip.

In this example, we show more details about the vessel and the wine in it. The LIP for the wine is displayed, and a list of the current jobs associated with the vessel. This list and associated buttons is duplicated form the Work Diary screen (*13.3.6)* and so will not be described here.

13.3.4.11 MORE READINGS

Further information regarding wine readings is available by pressing the "More Readings" button which causes the wine details window to appear. (Figure 64) This dialog is dedicated to the wine and displays several lists pertinent to the state and history of the wine.



Wine : 11HEGVSHIR 32240 litres

Comments :

LIP : Vintage: 2011 (100.00%)
Variety: Shiraz (99.94%),Sangiovese (0.03%),Cabernet Sauvignon (0.03%)
Region: Heathcote (100.00%)
Blocks: H2-4 (0.50%),H2-3 (0.93%),KATHY (1.01%),MAIN (46.54%),H2-2 (0.15%),H1-1 (18.58%),H1-2 (5.60%),

Readings : pH : 3.88 (18 Aug 2011 03:46:50 PM)
TA : 6.71 (18 Aug 2011 03:46:50 PM)
Free SO2 : 17.60 ppm (18 Aug 2011 03:40:42 PM)
Total SO2 : 39.20 ppm (18 Aug 2011 03:40:42 PM)
Temperature : 0 *C (15 Sep 2011 04:55:28 PM)

Baume : No Reading
Residual Sugar : 0.41 g/l (30 Aug 2011 03:36:47 PM)
Clinitest : No Reading
Alcohol : No Reading

Ammonia : No Reading
Malic Acid : No Reading
Primary Amino Acid : No Reading
V.A. : 0.65 mg/l (30 Aug 2011 05:22:44 PM)

CO2 : No Reading
D.O. : No Reading
Turbidity : No Reading
Filterability : No Reading

Cold Stable : No Reading
Heat Stable : No Reading
Guaiacol : No Reading
4-Methyl Guaiacol : No Reading
Laccase : No Reading

Additions

Type	Addition	Amount	Date	J

Readings

Value	Date	Job
Cooling : 5.0	21/09/2011 10:05:45 PM	
Heating : -999	21/09/2011 10:05:45 PM	
Temperature : 0*C	15/09/2011 4:55:28 PM	2739
Gas Blanket : Yes	15/09/2011 3:48:33 PM	2754
VA : 0.65mg/l	30/08/2011 5:22:44 PM	2609
Gluc : 0.31	30/08/2011 3:36:47 PM	2609
Fruc : 0.1	30/08/2011 3:36:47 PM	2609
Residual Sugar : 0.41g/l	30/08/2011 3:36:47 PM	2609
Temperature : 0*C	25/08/2011 4:25:56 PM	2627
PH : 3.88	18/08/2011 3:46:50 PM	2609
TA : 6.71	18/08/2011 3:46:50 PM	2609
FSO2 : 17.60ppm	18/08/2011 3:40:42 PM	2609
TSO2 : 39.20ppm	18/08/2011 3:40:42 PM	2609

Trials

Type	Details	Date

Awaiting Ratification

Value	Date	Job

Figure 64. Wine Readings Dialog

The readings section displays all the known analytical parameters and their values, if known. The additions list displays all the additions worknotes applied to this wine. The readings list displays all the historical readings applicable to this wine. The trial list displays all the completed tasting, blending or fining trials for the wine. The “Awaiting Ratification” list shows all the wine readings that need to be ratified before they are accepted. The functions to accept or reject these readings are duplicated from the “Readings Ratification” screen (see *13.3.5*) and so will not be described here.

13.3.4.12 COMMENTS

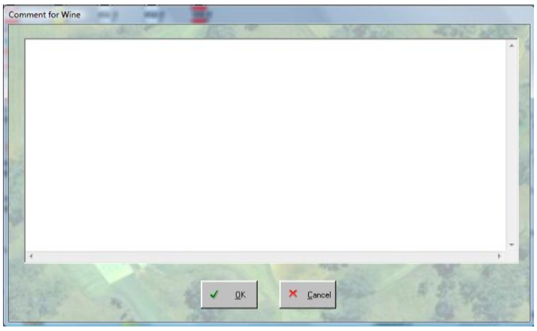
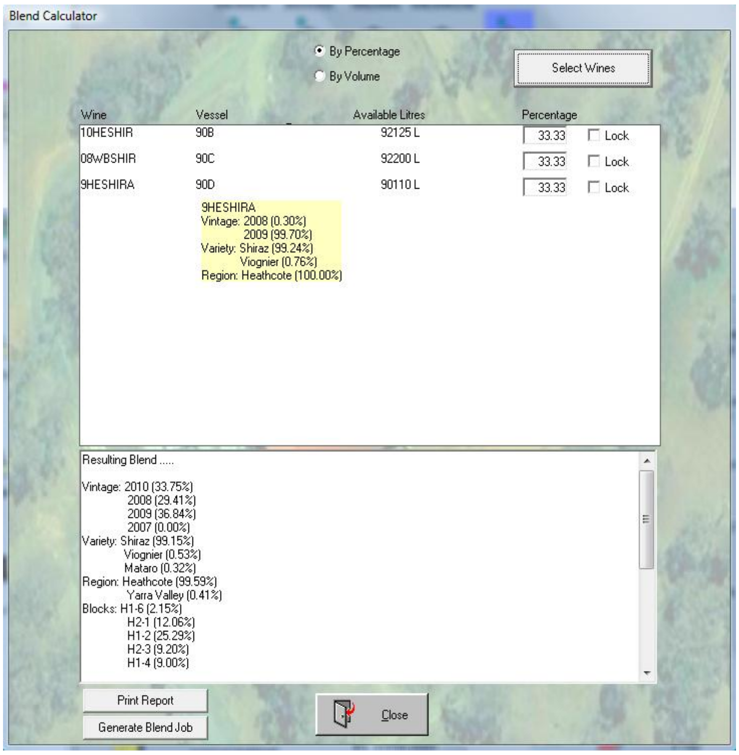


Figure 65. Wine Comments Dialog

Each wine has a comments field attached that allows the user to enter any relevant information about the wine. There is no structure to the field, and no size limit. The comments appear in many places for the wine, but can only be updated using the “Comments” popup menu and “Comments“ dialog (Figure 65) from the tank farm window.

13.3.4.13 BLEND CALCULATOR

The Blend Calculator (Figure 66) is an example of a tool required from first principles of winemaking philosophy for any self respecting winemaking software package. When a winemaker is looking to blend wines, there are often several aims. Sometimes, they need to know how much wine they can make of a particular blend, and how much more or less they could make if they tweaked the blend in some way. Other times they might be trying to hide a wine or blend it away, but need to know that they are not affecting the LIP claims for any wines. The motives can be superior, or they can be pragmatic. Nevertheless, during a blending trial, wine availability and LIP issues can be just as imperative as flavour profiles and other subjective considerations for wine quality.



Wine	Vessel	Available Litres	Percentage	
10HESHIR	90B	92125 L	33.33	□ Lock
08WBSHIR	90C	92200 L	33.33	□ Lock
9HESHIRA	90D	90110 L	33.33	□ Lock

```
9HESHIRA
Vintage: 2008 (0.30%)
         2009 (99.70%)
Variety: Shiraz (99.24%)
         Viognier (0.76%)
Region: Heathcote (100.00%)
```

```
Resulting Blend .....

Vintage: 2010 (33.75%)
         2008 (29.41%)
         2009 (36.84%)
         2007 (0.00%)
Variety: Shiraz (99.15%)
         Viognier (0.53%)
         Mataro (0.32%)
Region: Heathcote (99.59%)
        Yarra Valley (0.41%)
Blocks: H1-6 (2.15%)
        H2-1 (12.06%)
        H1-2 (25.29%)
        H2-3 (9.20%)
        H1-4 (9.00%)
```

Figure 66. Blend Calculator

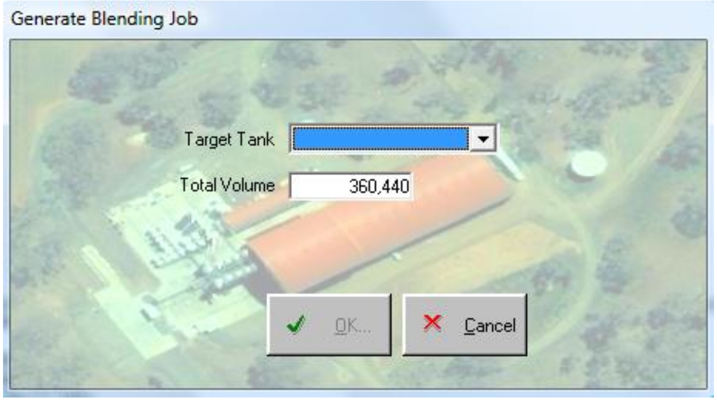
The aim of this dialog is to allow the user to quickly and simply manipulate the ratios of selected wines into a blend and then let the user know the LIP and volume calculations. It is available from the tank farm or barrel hall area of the winey map, as a general tool, and also from within the managed laboratory window as a means of doing the volume and LIP calculations during a formal blending trial worknote.

The wines in the list are fixed for a formal blending trial but can be changed when the tool is invoked from the winery map by pressing the “Select Wines” button to select and de-select wines to be included in the trial.

The means of specifying the ratio of each wine in the blend can be on a “By Percentage” or “By Volume” basis as indicated by the radio buttons at the top of the dialog. Each time the ratios are altered, they are corrected to ensure that the percentage adds up to 100 or the volumes do not exceed the available volume of each wine. The algorithm to correct the ratios for “By Percentage” is quite simple. The value being changed is maintained and the remainder is evenly distributed amongst the other wines. However, it is also possible to lock one or more wines to whatever value is specified, so that when the percentages are recalculated, only the unlocked wines can be altered. If it is not possible to reconcile the percentages, a message is generated and the blend is not calculated.

When the ratios or volume are correct, the blend is calculated. The process to do this is simply a matter of creating a temporary wine in memory, creating and setting the attributes of the LIP collection for that wine, then using the “BuildLIPString” method to generate the report. The maximum wine available that can be created for this blend is calculated as the temporary wine is built, by simply maintaining a static maximum allowed. As each percentage of source wine is added to the temporary wine, we work out how much wine could be created if this wine was the restricting factor. If this is less than the current maximum then make this the new maximum.

The resulting blend can be printed in report format, and the blend job can be generated by pressing the “Generate Blend Job” button which invokes the dialog shown...


Generate Blending Job

Target Tank

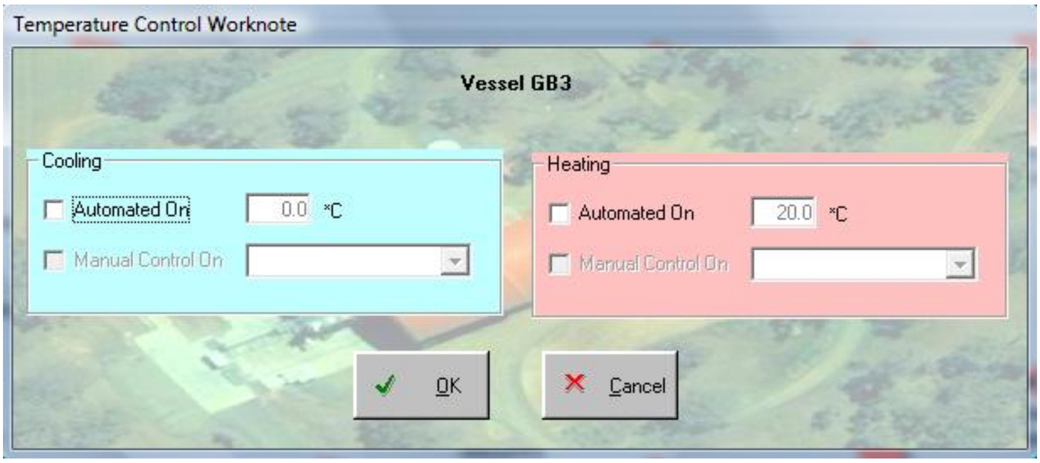
Total Volume 360,440

Figure 67. Blend Calculator - Blending Job Dialog

which allows the user to specify a target tank and volume, which are used to create the blending job.

13.3.4.14 Temperature Control

The “Temperature Control” dialog (Figure 68) is only available for vessels that have either manual or automated heating or cooling configured.



Temperature Control Worknote

Vessel GB3

Cooling

□ Automated On 0.0 °C

□ Manual Control On

Heating

□ Automated On 20.0 °C

□ Manual Control On

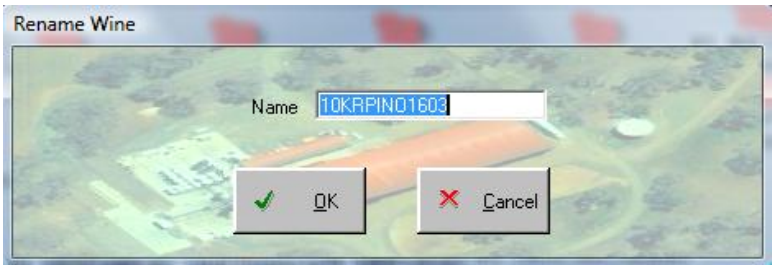
Figure 68. Temperature Control Dialog

The available options are reflected in their enabled status in the dialog when it appears.

Any changes result in the vessel’s HeatingOn or CoolingOn properties and the wine(if any) in the vessel having readings added indicating that they are being heated or cooled.

13.3.4.15 Rename Wine

The ability to change the name (Figure 69) of the wine reflects the principle that the wine is not identified in the system by its name but rather by its ID, so the name is free to be whatever the user wants it to be. It is up to the user to give the name as much meaning as they want. By default, wine names are generated according to the source fruit and data of harvest, but this has little meaning later when wines are blended and perhaps assigned to labels. The system always endeavours to make wine names unique but this function overrides that.



Rename Wine

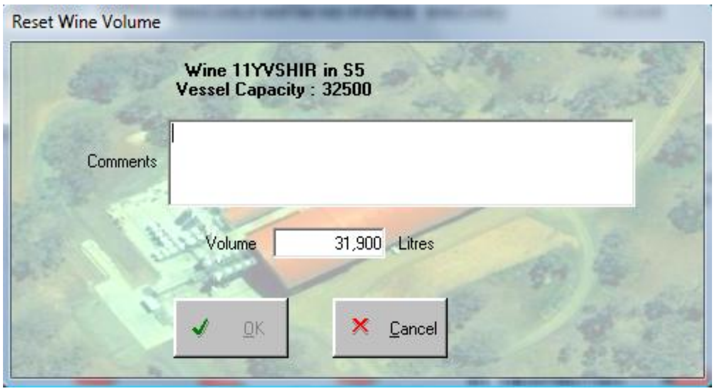
Name 10KRPINO1603

Figure 69. Wine Rename Dialog

13.3.4.16 Rename Barrel Set

Barrel sets are a construct of this project rather than an industry standard concept, so there are even fewer restrictions on the name a barrel set can have. However, it has been found that naming a barrel set after the wine it is or was used for is acceptable as long as the name is given the prefix "BS". This introduces the possibility of automatically prefixing barrel set names to ensure they are distinguishable from other vessels on sight, but this was complicated by suggestions later that the term "Barrel Set" was confusing and this should be resolved before a naming convention should be established.

13.3.4.17 Reset Wine Volume



Reset Wine Volume

Wine 11YVSHIR in S5
Vessel Capacity : 32500

Comments

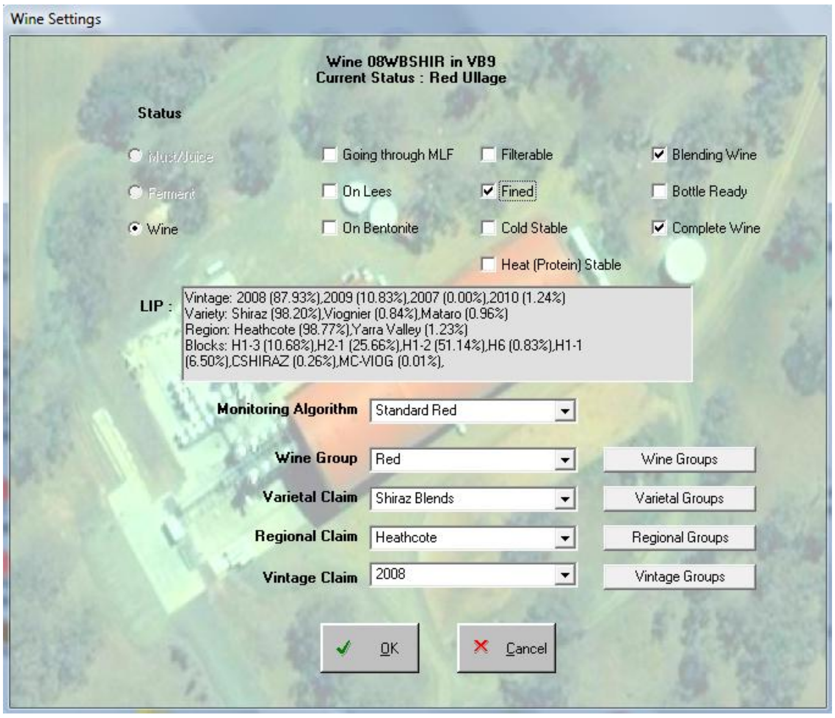
Volume 31,900 Litres

Figure 70. Reset Wine Volume Dialog

The function to reset the volume of a wine was found to be necessary as part of routine housekeeping in the system. The reasons for why the volumes sometimes need to be reset are many and varied, and so the function is associated with a job and worknote so that it appears as a task within the tracker function, and the explanation for the change can be stored in the comments for the job. (Figure 70)

13.3.4.18 Wine Settings

The Wine Settings dialog (Figure 71) simply sets the Groups, sub status flags, and monitoring algorithm for the selected wine.



Wine 08WBSHIR in VB9
Current Status : Red Ullage

Status

○ Must/Juice
○ Ferment
● Wine

□ Going through MLF
□ On Lees
□ On Bentonite

□ Filterable
☑ Fined
□ Cold Stable
□ Heat (Protein) Stable

☑ Blending Wine
□ Bottle Ready
☑ Complete Wine

LIP : Vintage: 2008 (87.93%),2009 (10.83%),2007 (0.00%),2010 (1.24%)
Variety: Shiraz (98.20%),Viognier (0.84%),Mataro (0.96%)
Region: Heathcote (98.77%),Yarra Valley (1.23%)
Blocks: H1-3 (10.68%),H2-1 (25.66%),H1-2 (51.14%),H6 (0.83%),H1-1 (6.50%),CSHIRAZ (0.26%),MC-VIOG (0.01%),

Monitoring Algorithm: Standard Red

Wine Group: Red

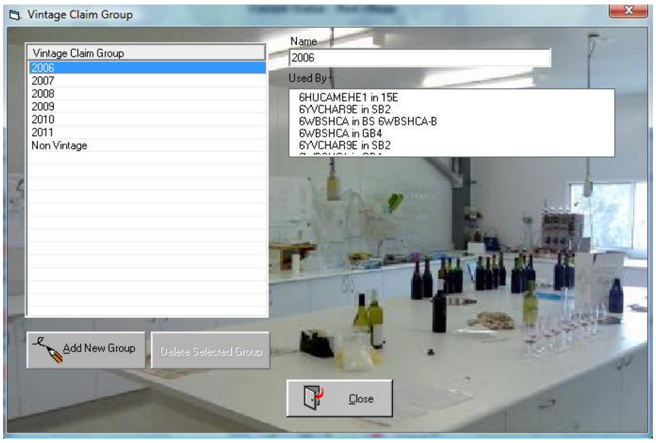
Varietal Claim: Shiraz Blends

Regional Claim: Heathcote

Vintage Claim: 2008

Figure 71. Wine Settings Dialog

The status for the wine can only be promoted from Must/Juice to Ferment to Wine but not in reverse. The other flags are set manually rather than being set as a result of any actions or events that might be associated with these flags. Monitoring Algorithms are created and edited in the Monitoring Algorithms screen (see *Figure 101. Monitoring Algorithm Dialog*, p.275). The remaining groups are maintained using the Wine Groups dialog (Figure 72) which is initialised to present the required group category; either Wine, Variety, Regional, or Vintage.



Vintage Claim Group
2006
2007
2008
2009
2010
2011
Non Vintage

Name: 2006

Used By:
6HUCAMEHE1 in 15E
6YVCHAR9E in SB2
6WBSHCA in BS 6WBSHCA-B
6WBSHCA in GB4
6YVCHAR9E in SB2

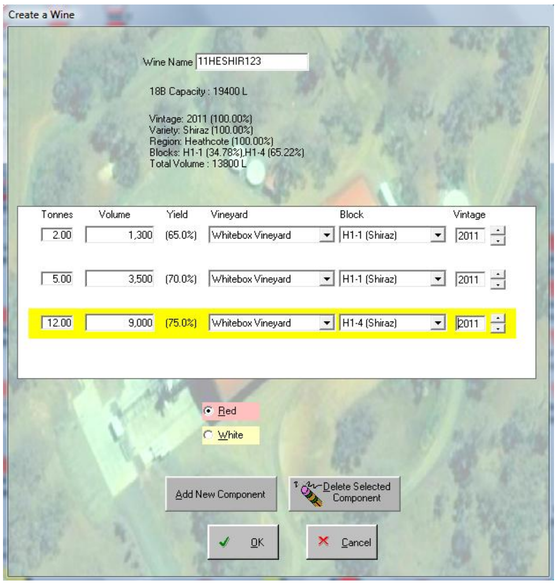
Figure 72. Wine Group Dialog

The name of the item can be changed by simply editing the name field to the right of the list. The “Delete Selected Item” button is only enabled when the selected item is not referred to by a wine.

The used by list is generated by using the UsedBy property of the WineGroup object. The "Add New" button simply adds a new item to the group with a default name.

13.3.4.19 CREATE A WINE

This function (Figure 73) was required when the system was being beta tested and needed to be initialised with data for all the wines that currently existed in the winery. It was decided that it should only be available for super-admin level users because it was not the sort of function that should be used on a whim. Of course, the history of the wine is not complete, and only the initial LIP information is recorded.


Create a Wine

Wine Name 11HESHIR123

18B Capacity : 19400 L

Vintage: 2011 (100.00%)
Variety: Shiraz (100.00%)
Region: Heathcote (100.00%)
Blocks: H1-1 (34.78%),H1-4 (65.22%)
Total Volume : 13800 L

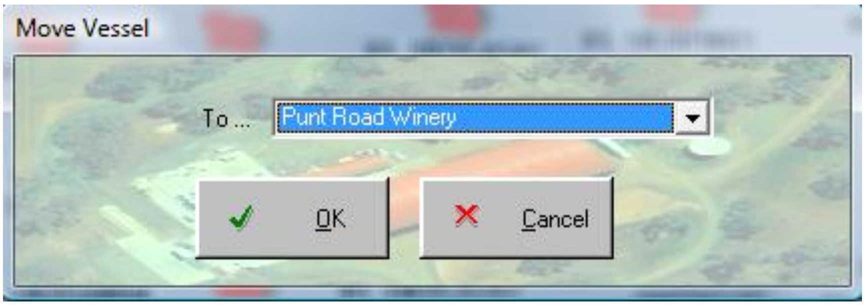
Tonnes	Volume	Yield	Vineyard	Block	Vintage
2.00	1,300	(65.0%)	Whitebox Vineyard	H1-1 (Shiraz)	2011
5.00	3,500	(70.0%)	Whitebox Vineyard	H1-1 (Shiraz)	2011
12.00	9,000	(75.0%)	Whitebox Vineyard	H1-4 (Shiraz)	2011

● Red
○ White

Figure 73. Create a Wine Dialog

13.3.4.20 MOVE A VESSEL

The vessel movement dialog (Figure 74) allows the user to specify the winery to which the selected vessel should be moved.


Move Vessel

To ... Punt Road Winery

Figure 74. Move Vessel Dialog

When OK'd, the wineryid property of the vessel is updated. This causes a vessel reading to be added to its reading collection with the winery id of the selected winery, thereby maintaining an historic record of the location of the vessel.

13.3.4.21 Additions

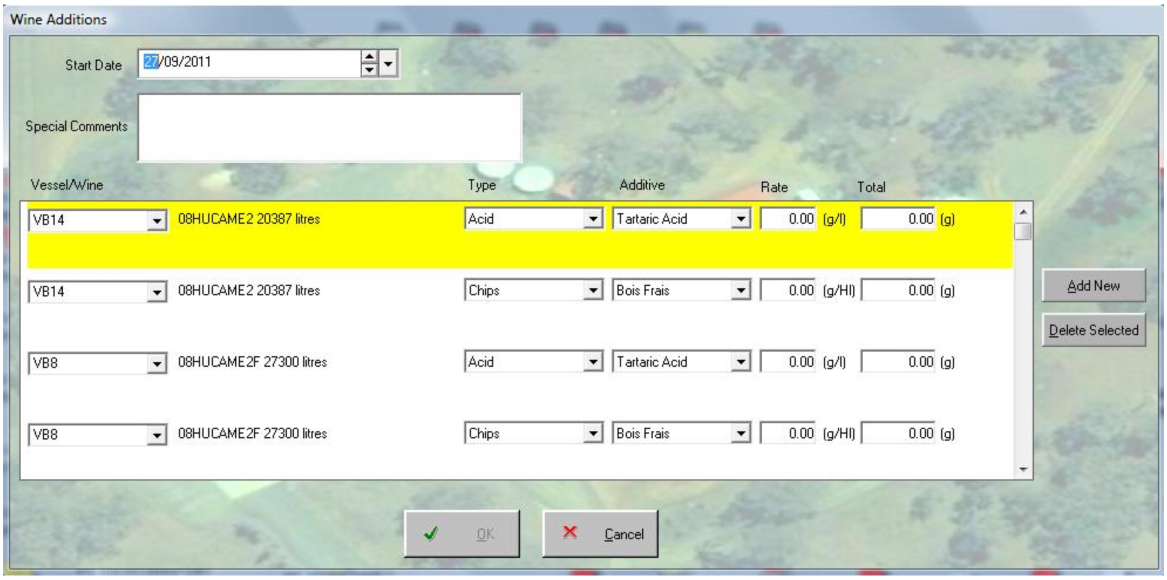
Additions to wines are given special treatment because of their impact on a wine and significance in the history and development of a wine. Worknotes for additions appear as individual jobs and as part of complex jobs so the worknotes need to be completed individually and we need to provide appropriate dialogs for their creation, modification and completion. The completion step is particularly important because it also includes the nomination of batch numbers for the actual material used, and the need to check the impact that the additive has on readings for the wine using the AffectspH, AffectsTA, AffectsSO2, and AffectsSugars flags.

When additions are displayed, the system needs to display the rates and amounts in appropriate units. These are referenced from the additives object which includes the parameters...

- Rate: the default rate of addition of the additive, for initialising dialogs
- Rate Factor: 1 = g/l, 100 = g/HL, 1000 = mg/l or ppm (the number is used to divide the rate to convert to grams.
- ByVolume: a flag that changes the basis of addition from mass in grams to volume in millilitres.

All these factors are used to change headings and calculations in all aspects of the dialogs used to administrator addition is the application.

The general dialog (Figure 75) for specifying an addition to a wine allows the user to nominate multiple additions to multiple wines in one job. This is because, in practice, this is how additions jobs are often created and completed, either multiple additions for a single wine, the same material is added to multiple wines, or multiple additions made to multiple wines. Each additive gets a single worknote for each wine.



Wine Additions

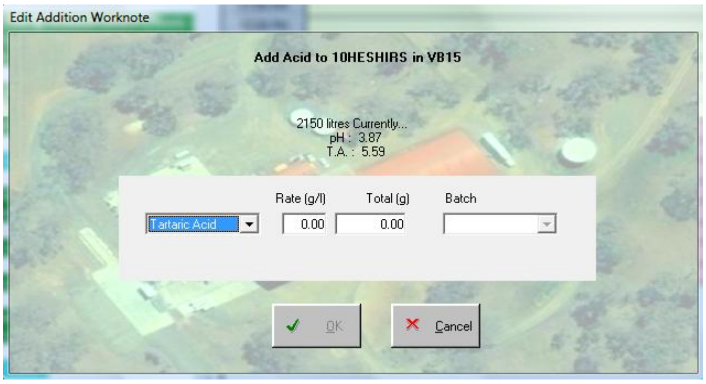
Start Date 27/09/2011

Special Comments

Vessel/Wine		Type	Additive	Rate		Total	
VB14	08HUCAME2 20387 litres	Acid	Tartaric Acid	0.00	(g/l)	0.00	(g)
VB14	08HUCAME2 20387 litres	Chips	Bois Frais	0.00	(g/Hl)	0.00	(g)
VB8	08HUCAME2F 27300 litres	Acid	Tartaric Acid	0.00	(g/l)	0.00	(g)
VB8	08HUCAME2F 27300 litres	Chips	Bois Frais	0.00	(g/Hl)	0.00	(g)

Figure 75. Multiple Wine/Additive Job dialog

The same dialog is used when the “Wine Addition” job is edited, however if an individual worknote is edited, the single wine addition worknote dialog is displayed (*Figure 76)*. This is done to avoid confusion for the user if they choose to edit a single worknote and do not want to be confronted with the details of the other worknotes in the job. This dialog is also used to complete all addition worknotes.



Edit Addition Worknote

Add Acid to 10HESHIRS in VB15

2150 litres Currently...
pH : 3.87
T.A. : 5.59

	Rate (g/l)	Total (g)	Batch
Tartaric Acid	0.00	0.00	

Figure 76. Edit Addition Worknote dialog

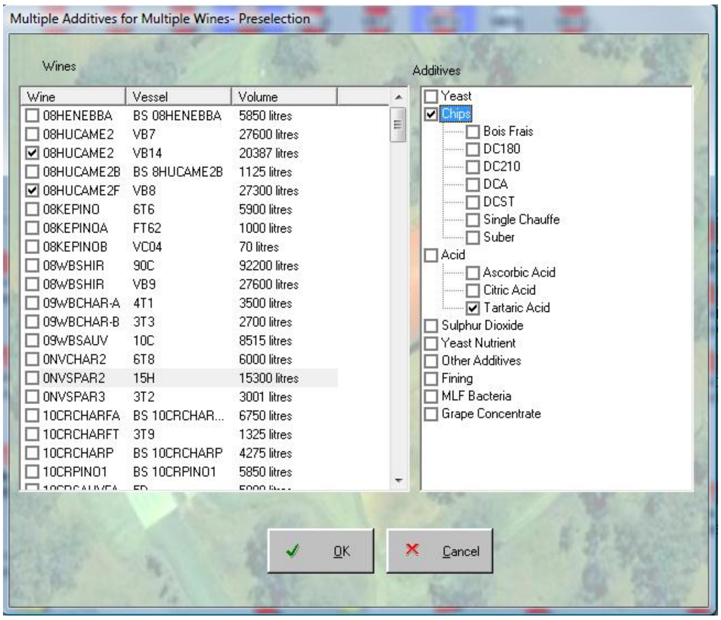
In this case, the OK button is not enabled until the batch number is supplied. As the worknote is completed, a check is made to see if any of the “Affects...” flags are set for the additive, and if so, relevant readings for the wine are cloned and flagged as suspect. These include...

AffectspH	pH and T.A readings
AffectsTA	pH and T.A readings
AffectsSO2	Free and Total SO_2 readings
AffectsSugars	Baume, Clinitest, Residual Fructose, Residual Glucose, Residual Sugar readings

When an additive has the "AffectsVolume" flag set, the volume of additive is also added to the LIP object for the wine, with the BlockID for the extra LIP object being set to 0. The additive type can be derived from the worknote ID also stored in the LIP object. The "Fortify" method is used to perform this function; its name is an historical reference to the only perceived (at the time) additive that could affect the volume being ethanol used to fortify wines into ports etc.

13.3.4.21.1 Multiple Additions

A composite job was created to allow the specification of multiple additions to multiple wines in a single job. This simply required an extra pre-selection dialog to be available to allow the user to select the wines and additive and/or additive types *(Figure 77. Multiple Wine/Addition Pre-selection dialog*) before presenting the Wine additions dialog as illustrated above (*Figure 75. Multiple Wine/Additive Job dialog*).



Wine	Vessel	Volume
□ 08HENEBBA	BS 08HENEBBA	5850 litres
□ 08HUCAME2	VB7	27600 litres
☑ 08HUCAME2	VB14	20387 litres
□ 08HUCAME2B	BS 8HUCAME2B	1125 litres
☑ 08HUCAME2F	VB8	27300 litres
□ 08KEPINO	6T6	5900 litres
□ 08KEPINOA	FT62	1000 litres
□ 08KEPINOB	VC04	70 litres
□ 08WBSHIR	90C	92200 litres
□ 08WBSHIR	VB9	27600 litres
□ 09WBCHAR-A	4T1	3500 litres
□ 09WBCHAR-B	3T3	2700 litres
□ 09WBSAUV	10C	8515 litres
□ 0NVCHAR2	6T8	6000 litres
□ 0NVSPAR2	15H	15300 litres
□ 0NVSPAR3	3T2	3001 litres
□ 10CRCHARFA	BS 10CRCHAR...	6750 litres
□ 10CRCHARFT	3T9	1325 litres
□ 10CRCHARP	BS 10CRCHARP	4275 litres
□ 10CRPINO1	BS 10CRPINO1	5850 litres

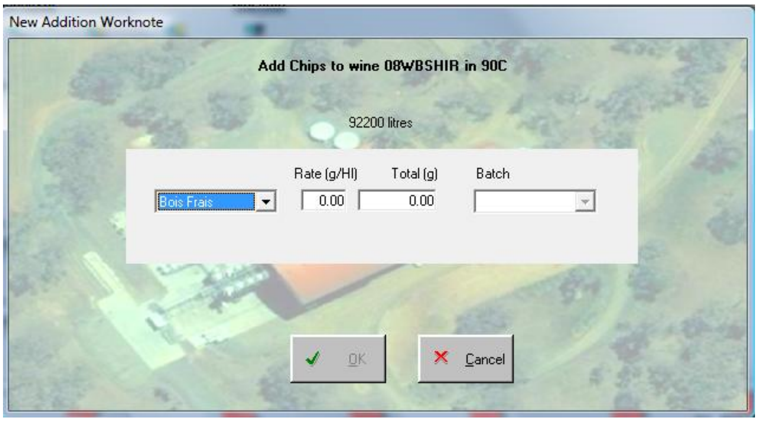
- □ Yeast
- ☑ Chips
  - □ Bois Frais
  - □ DC180
  - □ DC210
  - □ DCA
  - □ DCST
  - □ Single Chauffe
  - □ Suber
- □ Acid
  - □ Ascorbic Acid
  - □ Citric Acid
  - ☑ Tartaric Acid
- □ Sulphur Dioxide
- □ Yeast Nutrient
- □ Other Additives
- □ Fining
- □ MLF Bacteria
- □ Grape Concentrate

Figure 77. Multiple Wine/Addition Pre-selection dialog

This allows the user to select either additive types or specific additives so that when the Wine Additions dialog is displayed, it is pre-loaded with all the wines and additives. If only a type is selected then the default additive is preselected for that type.

13.3.4.21.2 Temporal additions and removal (Chips, Diffusers, Mox)

There are a few winemaking techniques that fall into an ambiguous descriptive area because they are partially additions and partially processes. The overriding aspect they have in common is that involve a time based application that needs to be considered when we analyse the impact of the addition/process on the wine. In other words, they neither involve a simple addition of a known amount of an additive or chemical, nor a one-off process applied to the wine, but rather a “temporal addition” that involves exposure to a material of some kind.



New Addition Worknote

Add Chips to wine 08WBSHIR in 90C

92200 litres

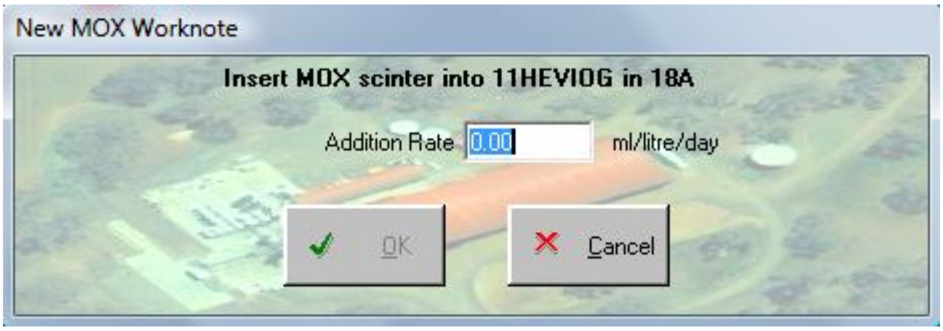
	Rate (g/Hl)	Total (g)	Batch
Bois Frais	0.00	0.00	

Figure 78. Addition Worknote Dialog

The three known materials are Chips, Inert Gas Blankets via diffusers, and Micro-oxygenation. We will not go into any detail about the specific chemical or oenological characteristics of these processes other than to say that the period of exposure as well as the nature and amount of the material is important to have record from a winemaking point of view. However, there are significant differences in some details, so each type needs to be described separately, to justify the implementation within the application.

Chips additions do involve the physical addition of the additive to the wine, and so this additive is available for selection using the same mechanisms for adding other additives within the multiple additive/wine scenario mentioned above. However it is also has its own menu structure (*13.3.4.2.7 Chips*). This is the only way the user can explicitly have the chips removed from a wine using a job/worknote. The addition of chips is recorded in the Chips collection object for the wine because there is insufficient detail in simply recording it as a reading in the readings/parameters collection

An inert gas blanket is neither an addition, nor a one-off process, but rather a time based process. It differs from a dry ice addition in that it is a constant application lasting indefinitely rather than a one-off temporary protective technique. This process also has its own menu structure for initiating and removing a diffuser from a vessel. (*13.3.4.2.8 Diffuser*) The presence and removal of a diffuser is recorded as a reading for the wine and flagged for the vessel in the “GasBlanket” property.



New MOX Worknote

Insert MOX scinter into 11HEVIOG in 18A

Addition Rate 0.00 ml/litre/day

Figure 79. MOX Worknote Dialog

Micro-oxygenation sinters introduce oxygen to the wine in very low doses. The dissolved oxygen levels in the wine are not impacted, but it is believed that the low concentration of oxygen simulates the maturation process that occurs in oak vessels. The rate that the oxygen is added is critical, and usually controlled by a Programmable Logic Controller (PLC) of some kind. The rate of addition is measured in millilitres of oxygenation, per litre of wine, per month. Just like the diffuser, micro-oxygenation is recorded in the wine reading as a rate and also in the Vessel's MOX property which simply records the presence or absence of a sinter.

13.3.4.22 Processes

There are many processes available to the user to apply to wines. Some are quite simple, and require no other parameters other than a record that they were done. Others require greater detail at both the creation and completion stages of the job lifecycle and are detailed next.

13.3.4.22.1 General rules for transferring wine

The underlying business rules for transferring wine are by far the most complex and important functions in the system. They draw together all the major aspects of the data model and illustrate one of the most important functions of the model; the ability to trace the history of the nature and make-up of the wines defined in the system. This section describes how this information is processed and how the model behaves when a transfer is completed. These rules become the basis for the business rules that are applied to the dialog that follow after this section.

Whenever wine is transferred from one tank to another, whether it is a simple transfer, a racking, a filtration job, a pressing, or just topping a wine, it is necessary to first categorise the transfer into one of four types which help identify the steps necessary to process the information.

1. Target tank is empty and no wine left in source tank
2. Target tank is empty but some wine left in source tank
3. Target tank already has wine in it and no wine left in source tank

4. Target tank already has wine in it but some wine left in source tank

These four options differ in what happens to the source wine, and what makes up the final wine. The first two options are simplest because initially there is no wine in the target tank and so the resulting wine(s) will only come from a single source. The latter two are complicated by the presence of wine already in the target tank. This requires extra processing to combine the LIP and details of each wine in an appropriate way. It should be noted that there are three main flags that are set as a result of the processing of the transfer that deal with how existing worknotes should be changed to take the transfer into account. Worknotes that have a wine target object type may need to have their target object changed to point to the new wine objects that may result from the process. The flags are bRollOldSourceToNewTarget, bRollOldSourceToNewSource, and bRollOldTargetToNewTarget, and their names clearly indicate the required changes. Other settings such as new names for wines, flags for cleaning vessels, removal of mox sinters or diffusers, remaining and transferred volumes, and other parameters are provided by the dialogs employing these functions.

13.3.4.22.2 Option 1. Target tank is empty and no wine left in source tank

Simply transferring a wine from one tank to another is the simplest option because it does not involve the creation of a new wine, and the source wine object remains active. Therefore, the pseudo code looks fairly simple...

```
Add wine reading to source wine setting the vessel to the target
Add wine reading to source wine setting the name to the new name if it has changed.
Add new volume information to the LIPS collection of source wine
Duplicate current lab readings and set as assumed
Set the status of the target vessel equal to the status of the source vessel
Add a reading to the target vessel setting the wineid
If the target vessel is a barrel set then add a vessel reading for each vessel
If the source vessel is flagged to be cleaned
       Set the state of the source vessel to "cleaned"
       Add a reading to the source vessel setting the wineid to 0
       If the vessel is a barrel set
              Set state of each vessel to "cleaned"
              Add wine reading for each vessel for wineid = 0
              Set Ullage flag to false for each vessel
       Set the MOX property to false
       Set the GasBlanket property to false
Otherwise
       Set the state of the source vessel to "dirty"
       Add a reading to the source vessel setting the wineid to 0
       If the vessel is a barrel set
              Set state of each vessel to "dirty"
              Add wine reading for each vessel for wineid = 0
              Set Ullage flag to false for each vessel
Set the wineid property of the source vessel to 0
Set target vessel GasBlanket property appropriately
Add reading to source wine for GasBlanket setting
If chips are not transferred
       Remove chips from the wine
IF MOX sinter is transferred
       Set mox property of source vessel to false
```

```
      Set mox property of target vessel to true
If target tank contained skins
      If source wine did not contain skins
            Add target vessel skins ID reading to wine
            Set wine skins property to true
            Set target vessel-skin-wineid property to wine id
            Set target vessel-skin-status property to "In Wine"
            Save the wine-skin object
      Otherwise
            Append skin description from target vessel to wine skin description
            Save the wine-skin object
            Set the target vessel skin object status to "dumped"
            Save the wine-skin object
Invoke the SetStatus method for the source vessel
Save the wine object
```

The most critical part of all of these functions is the sequence in which they occur. The model and the objects themselves react to changes in parameters so when a change is made, it may cause other parameters to change too, which might otherwise cause issues for consequent steps in the process if not recognised earlier and taken into consideration during processing. For example, empty tanks are "cleaned" of data after the information they held for the wine has been used and transferred.

13.3.4.22.3 Option 2. Target tank is empty but some wine left in source tank

The only significant difference between this option and the first option is that we need to create two new wines; one to replace the original source wine, and one to represent the new wine in the target vessel. The obvious difference in the pseudo code is that it refers to three wines; the original source, the new source and the new target...

```
Create and initialise a new wine object for wine left in source vessel (NSW)
Set the winery for NSW to the id of the current winery
Add the name to NSW  as a reading
Add the source vessel id to NSW as a reading
Add the colour to NSW as a reading, copied from the original source wine (OSW)
Add the status of NSW as a reading, copied from OSW
Set NSW lees setting to equal OSW lees setting
Duplicate Current Lab Readings from OSW for NSW
Copy groups from OSW to NSW
If the remaining wine in the source is lees
      Set bRollOldSourceToNewTarget to True
      Set Lees property of NSW = true
      If the source vessel is to be cleaned
            Set NSW Wine status to "Dumped"
            Set source vessel wine id = 0
            Set source vessel state to "cleaned"
            Set source vessel mox to false
            Set source vessel chips to false
            Set source vessel GasBlanket to false
            Add reading to source vessel, wine id = 0
            If source vessel is a barrel set
                  Add reading to each vessel, wine id = 0
                  Add reading to each vessel, state is "cleaned"
                  Add reading to each vessel, ullage = false
      Otherwise
            For NSW, Set Current Lab Readings As Assumed
Otherwise
      Set bRollOldSourceToNewSource = True
```

```
       For NSW, Set Current Lab Readings As Assumed
Add reading to NSW for GasBlanket setting
Set source vessel GasBlanket property appropriately
Copy Onskins flag from OSW to NSW
Add ID of OSW to NSW Source wine collection
Copy current LIPs from OSW to NSW, correcting volume for new amount

Create and initialise a new wine object for wine left in target vessel (NTW)
Set the winery for NTW to the id of the current winery
Add the name to NTW as a reading
Add the source vessel id to NTW as a reading
Add the colour to NTW as a reading, copied from the original source wine (OSW)
Add the status of NTW as a reading, copied from OSW
Set NTW lees setting to equal OSW lees setting
Copy Onskins flag from OSW to NTW
Add ID of OSW to NTW Source wine collection
Duplicate Current Lab Readings from OSW for NTW
Copy groups from OSW to NTW
For NTW, Set Current Lab Readings As Assumed
If target vessel is a barrel set
       Add reading to each vessel, wine id = 0
Otherwise
       Add reading to target vessel, wine id = 0
Add reading to NTW for GasBlanket setting
Set target vessel GasBlanket property appropriately
Copy current LIPs from OSW to NTW, correcting volume for new amount

Add reading to OSW, status = "transferred away"
Save OSW
IF MOX sinter is transferred
       Set mox property of source vessel to false
       Set mox property of target vessel to true
If chips are transferred
       Loop through all current chips in OSW and add them to NTW
Otherwise
       Loop through all current chips in OSW wine and add them to NSW

If Target vessel has skins but none in source
       Add wine reading to NTW, skins from Original Target Wine (OTW)
       Set Onskins property for NTW to true
       Set wineid for skins in target vessel to NTW
       Set status for skins in target vessel to "In Wine"
       Save skins in NTW
Otherwise if Skins in source wine and target
       Append skins description from OSW to skins on target vessel
       Set wineid for skins in target vessel to NTW
       Set status for skins in target vessel to "In Wine"
       Save skins in NTW
       Add reading to NTW, skins
       Save skins in NSW
Otherwise if Skins in source but not in target
       Create new skin object
       Set PressWN to value in Source skin
       Copy Description from source skin
       Set status to inwine
       Set wineid to NWT
       Save the new skin
       Add reading to NWT, skin = new skin
       Set Onskins property for NTW to true
Save NSW
Save NTW
```

13.3.4.22.4 Option 3. Target tank already has wine in it and no wine left in source tank

This third option introduces the concept of transferring wine onto another wine. The processing becomes more complicated because we need to turn two wines into one.

```
Set bRollOldSourceToNewTarget = True
Set bRollOldTargetToNewTarget = True
Use CombineWines method to combine the source and target wines into one
Set source vessel wine = 0
If the source vessel is flagged to be cleaned
    Set the state of the source vessel to “cleaned”
    Add a reading to the source vessel setting the wineid to 0
    If the vessel is a barrel set
        Set state of each vessel to “cleaned”
        Add wine reading for each vessel for wineid = 0
        Set Ullage flag to false for each vessel
    Set the MOX property to false
    Set the GasBlanket property to false
Otherwise
    Set the state of the source vessel to “dirty”
    Add a reading to the source vessel setting the wineid to 0
    If the vessel is a barrel set
        Set state of each vessel to “dirty”
        Add wine reading for each vessel for wineid = 0
        Set Ullage flag to false for each vessel

Set the source vessel Status
Save OSW
Add chips from OTW to NTW
If source chips are to be moved then move them from source to target wine
Set target vessel GasBlanket property appropriately
Add reading to target wine for GasBlanket setting
If flagged, transfer MOX from Source Vessel to Target Vessel
If Target skins but none in source
    Change wineid for OTW skins from OTW to NTW
    Add reading for NTW, skins
    Save skins at NTW
Otherwise if Skins in source wine and target
    Change wineid for OTW skins from OTW to NTW
    Add reading for NTW, skins
    Set NTW onskins = true
    Append skin description from OSW to NTW
    Save skins at NTW
Otherwise if Skins in source, target no skins
    Set OSW skin wineid to NTW
    Add reading to NTW. Skin
    Set NTW OnSkins = true
    Save NTW skins object
Save NTW
```

13.3.4.22.5 Option 4. Target tank already has wine in it but some wine left in source tank

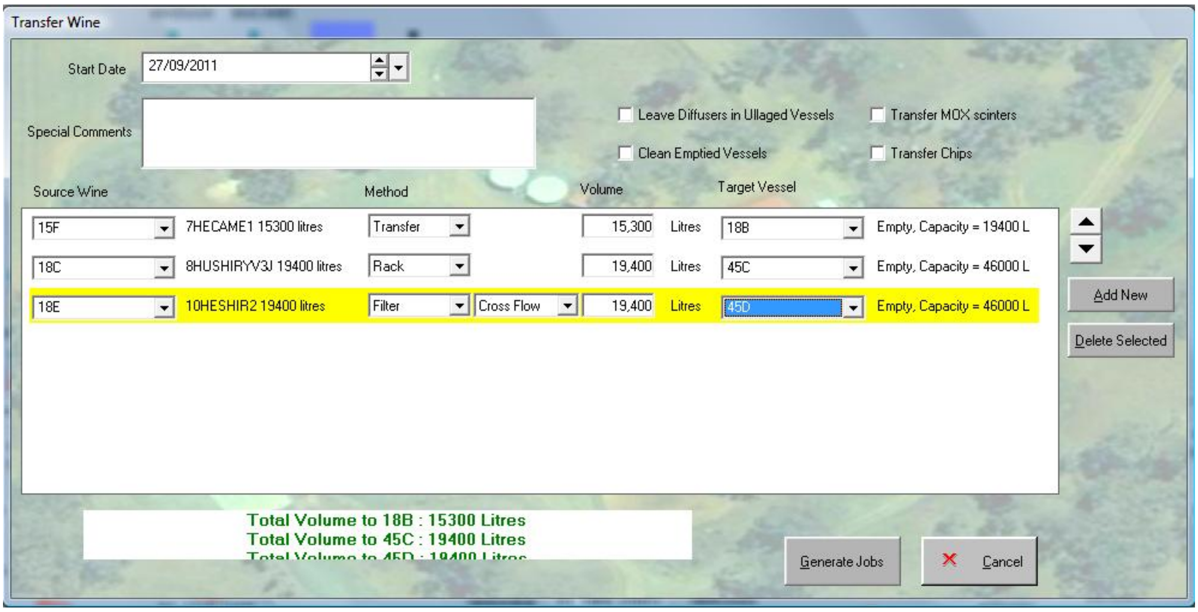
This option is used for topping wines as well as common transfers where wine or lees is left at the source and wine is already in the target vessel. In this case we create two new wines to replace the two original wines. It should be noted that, in the event that the remaining wine in the source tanks is lees, then it may be flagged to be immediately dumped, and therefore have a very short life.

```
Set bRollOldSourceToNewSource = True
Set bRollOldTargetToNewTarget = True
Create new wine at source(NSW)
Set NSW name, colour, vessel, and status readings
If NSW is lees
    Set NSW lees flag
    If Source vessel is to be cleaned
        Set Source Vessel state to cleaned
        Set NSW status to “Dumped”
        Add a reading to the source vessel setting the wineid to 0
        If the vessel is a barrel set
```

```
                    Set state of each vessel to "cleaned"
                    Add wine reading for each vessel for wineid = 0
                    Set Ullage flag to false for each vessel
             Set the MOX property to false
             Set the GasBlanket property to false
             Set Chips flag to False
Otherwise
       Copy lees flag from OSW to NSW
       Duplicate Lab readings from OSW to NSW
       Copy groups from OSW to NSW
       Set all lab readings for NSW to Assumed
If diffuser remains in source vessel, set flag and add reading
Copy skins flag from OSW to NSW
Add OSW as source for NSW
Duplicate LIP (adjusting volume) for NSW from OSW
Save NSW
Create NTW
Set Lees for NTW based on OTW is lees and OSW is lees
Set Onskins for NTW based on OTW is onskins or OSW is onskins
Set colour reading for wine as red if either OSW or OTW is red, otherwise white
Set name, vessel, readings to NTW
Set status for NTW to greater of values for OSW and OTW
Duplicate OSW LIPS in NTW, with volume set to amount transferred
Duplicate OTW LIPS in NTW
Add OSW and OTW as sources for NTW
Add chips from OTW to NTW
Set GasBlanket for Target Vessel and add reading for NTW
Set MOX for Target Vessel and add reading for NTW
Move chips from Source to NTW if required
If Target has skins but none in source
       Set skins object in OTW to have WineID = NTW
       Add skins reading to NTW
       Set NTW OnSkins to true
       Save NTW Skin object
Otherwise if Skins in source wine and target
       Set skins object in OSW to have WineID = NSW
       Set skins object in OTW to have WineID = NTW
       Append Skin description from NSW to NTW Skin
       Add skins reading to NSW
       Add skins reading to NTW
       Save NSW Skin object
       Save NTW Skin object
Otherwise if Skins in source, target no skins
       Create new skin object
       Copy PressWN and Description from skin in OSW
       Set Status to InWine
       Set WineID to NTW
       Save Skin object
       Add skin reading to NTW
       Set NTW to Onskins true
       Set OSW skin object's wineid to NSW
       Save NSW skin object
Save NTW
Set Target Vessel Status
Add status reading to OSW, TRANSFERRED AWAY
Add status reading to OTW, TRANSFERRED AWAY
Save OTW and OSW
```

13.3.4.22.6 Transfer, Rack, and Filter

The standard method for transferring, racking or filtering a wine from one vessel to another utilises the same dialog. The concept of a "sequenced transfer" job is to allow the user to schedule a sequence of transfers of wine that would not otherwise be possible because individual jobs would overlap. For example, if the user wanted to transfer all of wine X from vessel A to vessel B , then all of wine Y from vessel C to vessel A, this would not be possible to create as separate jobs because vessel A currently has wine in it and there is insufficient space for wine Y. A sequenced transfer takes into account the fact that previous transfers in the sequence alter the capacity and availability of vessels, and so the system can allow a complex sequence of events to be specified.



Transfer Wine

Start Date 27/09/2011

Special Comments

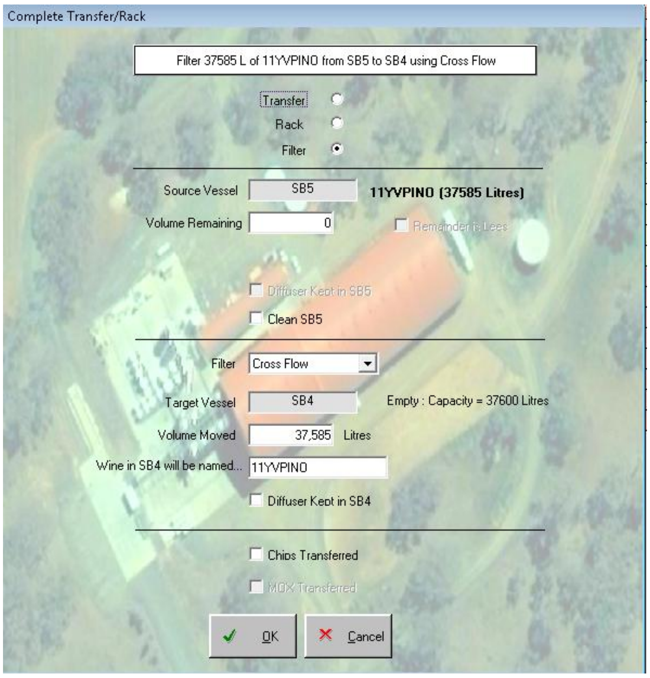
□ Leave Diffusers in Ullaged Vessels
□ Transfer MOX scinters
□ Clean Emptied Vessels
□ Transfer Chips

Source Wine		Method		Volume		Target Vessel	
15F	7HECAME1 15300 litres	Transfer		15,300	Litres	18B	Empty, Capacity = 19400 L
18C	8HUSHIRYV3J 19400 litres	Rack		19,400	Litres	45C	Empty, Capacity = 46000 L
18E	10HESHIR2 19400 litres	Filter	Cross Flow	19,400	Litres	45D	Empty, Capacity = 46000 L

Total Volume to 18B : 15300 Litres
Total Volume to 45C : 19400 Litres
Total Volume to 45D : 19400 Litres

Figure 80. Transfer/Filter/Rack Wine Dialog

The important thing to note is that the dialog and business rules allow any transfers to be specified and sequenced. The consequences for the LIP are calculated but the system takes no responsibility for the consequences of an incorrect sequence, from a winemaking perspective. It will simply respond to the events as they are completed. The dialog (Figure 80) shows the creation of the sequence whereas, each worknote, when completed, uses the following dialog. (Figure 81)


Complete Transfer/Rack

Filter 37585 L of 11YVPINO from SB5 to SB4 using Cross Flow

Transfer ○
Rack ○
Filter ●

Source Vessel SB5 11YVPINO (37585 Litres)

Volume Remaining 0 □ Remainder is Lees

□ Diffuser Kept in SB5

□ Clean SB5

Filter Cross Flow

Target Vessel SB4 Empty : Capacity = 37600 Litres

Volume Moved 37,585 Litres

Wine in SB4 will be named... 11YVPINO

□ Diffuser Kept in SB4

□ Chips Transferred

□ MOX Transferred

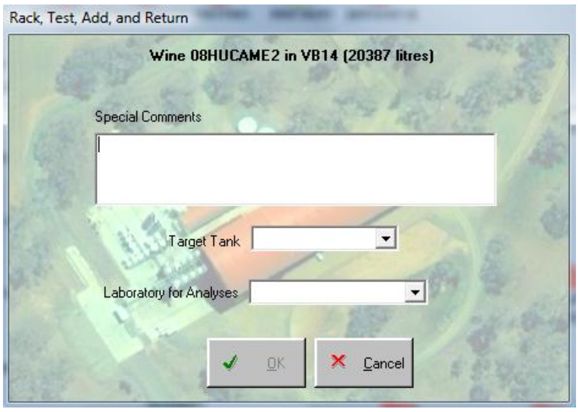
Figure 81. Complete Transfer/Filter/Rack Wine Dialog

Here we see how the application allows the user to specify all the parameters needed to complete the process according to the rules for transferring wine specified in *13.3.4.22.1 General rules for transferring wine.*

The dialog automatically attempts to name any new wines but also offers to restore the names back to any original settings if appropriate.

13.3.4.22.7 RACK TEST ADD RETURN

This is an example of a composite job involving different types of worknotes for different target facilities in a single job.


Rack, Test, Add, and Return

Wine 08HUCAME2 in VB14 (20387 litres)

Special Comments

Target Tank

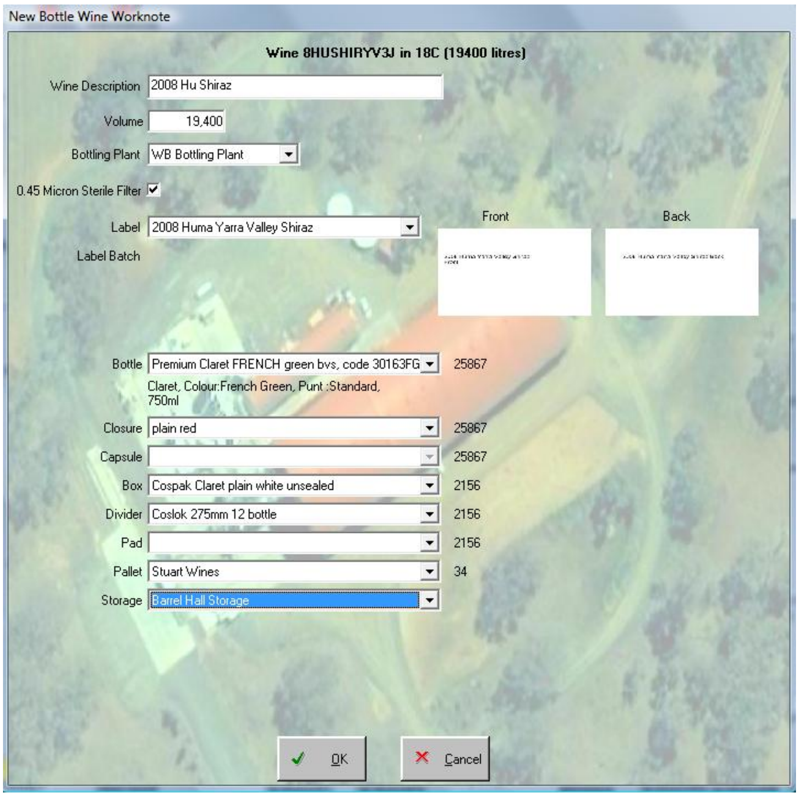
Laboratory for Analyses

Figure 82. Rack, Test Add, Return Dialog

The user nominates the target tank that the source wine should be racked to and the laboratory that the analyses should be performed. (Figure 82) The analyses are for pH and T.A. and for SO_2. Because each worknote is then treated individually when being completed, the racking can become a complete or partial transfer, and the analyses can be cancelled if not required. There is also an option to simply do a rack and return without the analyses.

13.3.4.22.8 Bottle Wine

Bottling wine is also regarded as a transfer in that it uses similar rules to treat the wine as specified in the general rules for transfers, only that the target wine gets a status of bottled, and a separate bottled wine object is also created.



New Bottle Wine Worknote

Wine 8HUSHIRYV3J in 18C (19400 litres)

Wine Description: 2008 Hu Shiraz

Volume: 19,400

Bottling Plant: WB Bottling Plant

0.45 Micron Sterile Filter ☑

Label: 2008 Huma Yarra Valley Shiraz

Label Batch

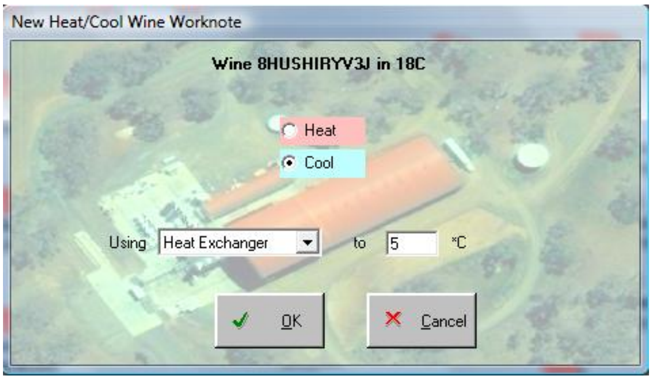
Bottle	Premium Claret FRENCH green bvs, code 30163FG Claret, Colour:French Green, Punt :Standard, 750ml	25867
Closure	plain red	25867
Capsule		25867
Box	Cospak Claret plain white unsealed	2156
Divider	Coslok 275mm 12 bottle	2156
Pad		2156
Pallet	Stuart Wines	34
Storage	Barrel Hall Storage	

Figure 83. Create Wine Bottling Worknote Dialog

The dialog (Figure 83) allows the complete selection of all major dry goods for the bottling session and also calculates the numbers required based on the volume of wine to be bottled. Upon completion of the job, batch numbers are required for all dry goods selected. Of course, not all dry goods types are mandatory; for example cleanskin bottling without labels.

If there is wine left in the source tank, then two new wines are completed; one for the remainder in the tank and one which gets a status of bottled. The bottled wine object, on the other hand, is used to store all the dry goods information more specific to the bottling process rather than representing the wine object.

13.3.4.22.9 Heat/Cool a Wine



New Heat/Cool Wine Worknote

Wine 8HUSHIRYV3J in 18C

○ Heat
● Cool

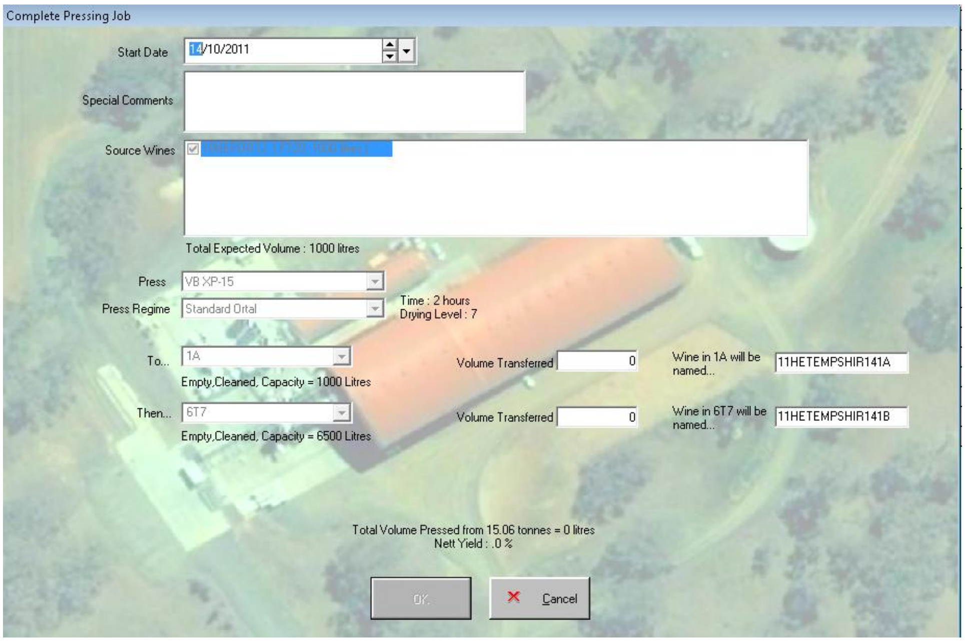
Using Heat Exchanger to 5 °C

Figure 84. Heat/Cool Wine dialog

Aside from temperature control of some vessels, it may be necessary to heat or cool a wine as part of a process rather than setting a storage temperature for a wine. For example, wine needs to be bottled at a minimum of 16°C, whereas cold stabilisation needs to be performed at near 0°C. Various heating and cooling methods are configured for each winery, depending on the equipment available; some can be used for both heating and cooling a wine. This dialog (Figure 84) allows the specification of how and to what temperature a wine should be heated or cooled as an individual worknote and job.

13.3.4.22.10 Press

Pressing a wine or wines is another form of transfer that also follows the same general rules for transfers of wine (*13.3.4.22.1 General rules for transferring wine*)



Complete Pressing Job

Start Date: 14/10/2011

Special Comments:

Source Wines: ☑ [illegible]

Total Expected Volume : 1000 litres

Press: VB XP-15

Press Regime: Standard Ortal

Time : 2 hours
Drying Level : 7

To...: 1A
Empty,Cleaned, Capacity = 1000 Litres

Volume Transferred: 0

Wine in 1A will be named...: 11HETEMPSHIR141A

Then...: 6T7
Empty,Cleaned, Capacity = 6500 Litres

Volume Transferred: 0

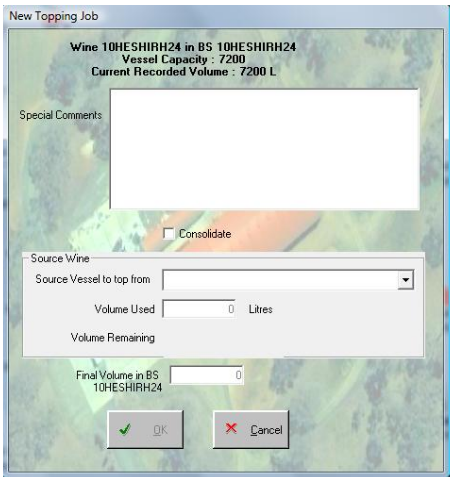
Wine in 6T7 will be named...: 11HETEMPSHIR141B

Total Volume Pressed from 15.06 tonnes = 0 litres
Nett Yield : .0 %

Figure 85. Wine Pressing Dialog

The process must also include specification of a press and a pressing regime. The dialog (Figure 85) allows for any number of source wines (only on skins wines are listed) to be selected and pressed into up to 3 target vessels. Upon completion, the respective volumes and wine names are specified and the standard rules are applied to create the required number of new wines and to retire the source wine objects.

13.3.4.22.11 TOPPING



New Topping Job

Wine 10HESHIRH24 in BS 10HESHIRH24
Vessel Capacity : 7200
Current Recorded Volume : 7200 L

Special Comments:

□ Consolidate

Source Wine

Source Vessel to top from:

Volume Used: 0 Litres

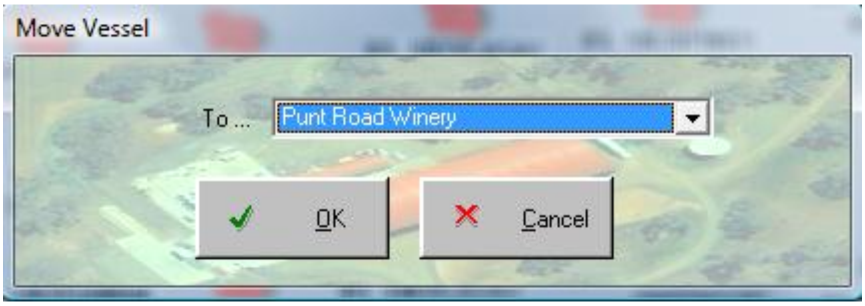
Volume Remaining

Final Volume in BS 10HESHIRH24: 0

Figure 86. Wine Topping Dialog

Topping a wine is typically necessary for wine in barrels in order to minimise the wine's exposure to the air because of losses from evaporation, leaks, or other processes. Often, in order to keep the LIP of a wine constant, a single barrel is nominated and sacrificed to top up other vessels, then this barrel is protected with argon gas or put into a variable capacity vessel. This effectively consolidates the wine within its vessels. Upon completion, all we need to know is the total volume of the wine which tells us the loss, and we simply adjust the volume of the wine without effecting LIP ratios. In other situations when it is not a concern to maintain the “independence” of a wine, the vessels are individually topped from another wine. The important parameters are the final volume of the target wine and the amount of wine used from the donor wine. From these numbers we can deduce both the losses of the original wine and the new LIP by first adjusting the volume of the wine to the new reduced volume, then adding the appropriate volume of wine from the donor, using the general rules option 4 (or 3 if all the source wine is used)

13.3.4.22.12 MOVE A WINE



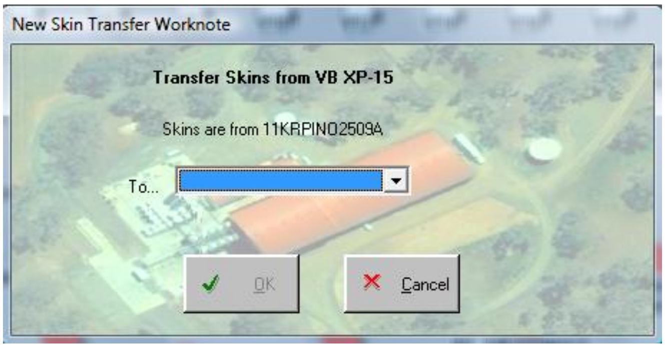
Move Vessel

To ... Punt Road Winery

Figure 87. Move a Wine Dialog

Moving a wine from one winery to another is generally not necessary for wineries of the scale within the scope of this project. However, this function became necessary on more than one occasion during the development of the data model, despite earlier scepticism. The function of the dialog (Figure 86) is very simple, requiring the change in WineryID for the vessel and the removal and subsequent addition of the vessel from the respective winery.vessel collections, with the only complication being if the vessel is a barrel set.

13.3.4.22.13 TRANSFER SKINS



New Skin Transfer Worknote

Transfer Skins from VB XP-15

Skins are from 11KRPINO2509A

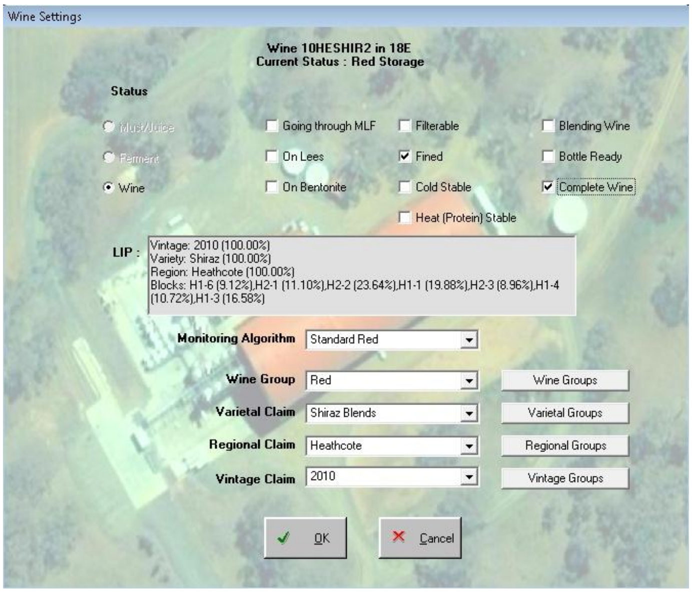
To...

Figure 88. Transfer Skins Dialog

This is the dialog (Figure 88) that is displayed for the “Transfer Skins To...” option. Only vessels of a vessel type that has “OnSkinsFermentor” flag set to true appears in the target list, so that skins are not transferred into inappropriate vessels.

13.3.4.23 SETTINGS

The Wine “Settings” dialog (Figure 89) allows the generic wine status levels, so-called “sub status” flags, and groups to be modified for an individual wine.



Wine Settings

Wine 10HESHIR2 in 18E
Current Status : Red Storage

Status

○ Must/Juice
○ Ferment
● Wine

□ Going through MLF
□ On Lees
□ On Bentonite
□ Filterable
☑ Fined
□ Cold Stable
□ Heat (Protein) Stable
□ Blending Wine
□ Bottle Ready
☑ Complete Wine

LIP :

Vintage: 2010 (100.00%)
Variety: Shiraz (100.00%)
Region: Heathcote (100.00%)
Blocks: H1-6 (9.12%),H2-1 (11.10%),H2-2 (23.64%),H1-1 (19.88%),H2-3 (8.96%),H1-4 (10.72%),H1-3 (16.58%)

Monitoring Algorithm: Standard Red
Wine Group: Red
Varietal Claim: Shiraz Blends
Regional Claim: Heathcote
Vintage Claim: 2010

Figure 89. Wine Settings Dialog

The substatus flags, which vary based on the status of the wine, were accumulated and designed from observing the significant concepts held by winemakers during the post-fermentation, maturation process. These flags (Table 23) are simple milestones presented in order of the generally accepted sequence of achievement, but this is not enforced. The prospect of additional flags is easily achieved because these sub-status states are stored as a single reading (called substatus) using a bitmap for each.

Status	**Substatus**
Must/Juice	None
Ferment	Early = 1, Mid = 2, Late = 4
Wine	MLF = 1, ONLEES = 2, ONBENTONITE = 4, FILTERABLE = 8, FINED = 16, COLDSTABLE = 32, HEATSTABLE = 64, BLENDINGWINE = 128, BOTTLEREADY = 256, COMPLETEWINE = 512

Table 23. Status and Substatus flags

Once a wine is set to a new status (from juice to ferment or ferment to wine) it cannot be undone, so the meaning of the substatus reading is interpretable by getting the status flag at the same time.

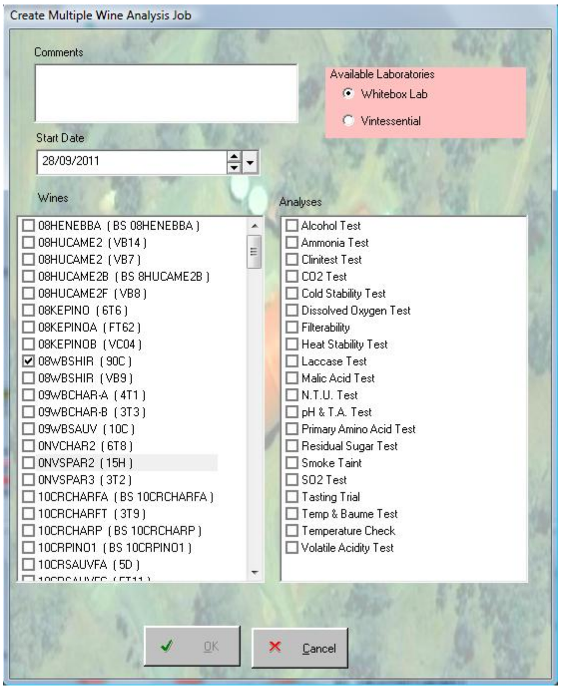
Monitoring Algorithms are discussed in greater detail later (*13.3.7 Monitoring and Algorithms*) but this dialog offers another way of changing the algorithm setting of the wine without having to navigate away from the Winery Map screen.

The wine groups are four categories for completely free-form categorising of the wines. The claims type groups are there to allow wines to be grouped by the user rather than by strict observance of LIP information. These groups are not mandatory. This apparently vague structure became necessary because it allows complete flexibility for the user to categorise the wines using whatever conceptual criteria they like. It was noticed that wines tended to be grouped by one of the three LIP groups, but also occasionally by some other criteria such as intended label, and usually by more than one criteria at a time. It was not appropriate to enforce any more hard-coded structure than this.

13.3.4.24 Laboratory Tests

Menu options to generate laboratory analyses are either single or composite jobs. The composite options include a basic multiple analyses job, as well as pre-bottling and pre-coarse-filtering jobs.

13.3.4.24.1 Multiple Analyses



Create Multiple Wine Analysis Job

Comments:

Available Laboratories
● Whitebox Lab
○ Vintessential

Start Date: 28/09/2011

Wines
- □ 08HENEBBA (BS 08HENEBBA)
- □ 08HUCAME2 (VB14)
- □ 08HUCAME2 (VB7)
- □ 08HUCAME2B (BS 8HUCAME2B)
- □ 08HUCAME2F (VB8)
- □ 08KEPINO (6T6)
- □ 08KEPINOA (FT62)
- □ 08KEPINOB (VC04)
- ☑ 08WBSHIR (90C)
- □ 08WBSHIR (VB9)
- □ 09WBCHAR-A (4T1)
- □ 09WBCHAR-B (3T3)
- □ 09WBSAUV (10C)
- □ 0NVCHAR2 (6T8)
- □ 0NVSPAR2 (15H)
- □ 0NVSPAR3 (3T2)
- □ 10CRCHARFA (BS 10CRCHARFA)
- □ 10CRCHARFT (3T9)
- □ 10CRCHARP (BS 10CRCHARP)
- □ 10CRPINO1 (BS 10CRPINO1)
- □ 10CRSAUVFA (5D)

Analyses
- □ Alcohol Test
- □ Ammonia Test
- □ Clinitest Test
- □ CO2 Test
- □ Cold Stability Test
- □ Dissolved Oxygen Test
- □ Filterability
- □ Heat Stability Test
- □ Laccase Test
- □ Malic Acid Test
- □ N.T.U. Test
- □ pH & T.A. Test
- □ Primary Amino Acid Test
- □ Residual Sugar Test
- □ Smoke Taint
- □ SO2 Test
- □ Tasting Trial
- □ Temp & Baume Test
- □ Temperature Check
- □ Volatile Acidity Test

Figure 90. Multiple Analyses Selection Dialog

The multiple analysis job only allows analyses that do not require further parameters (other than the identification of the wine) to be created so that the creation process is performed within the function of the dialog. The job type flags “GenerallyAvailable” and “Enabled” are used to test this criterion. There are only a few analytical processes that require greater specification, and these are detailed later in this section.

This function is very popular for winemakers when they wish to select a group of wines and give then the same treatment without having to repeat the selection process for each wine. The dialog (Figure 90) allows the user to specify which laboratory should perform the analyses, and the test date can be set to book the analysis ahead of time.

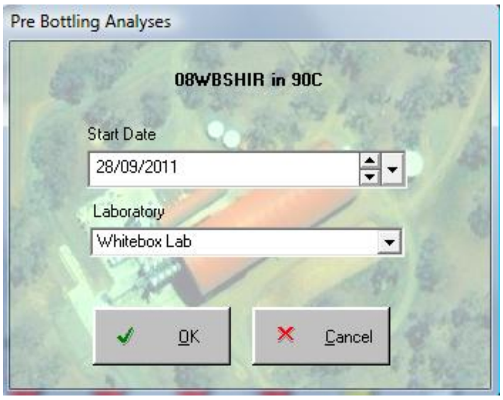
13.3.4.24.2 PRE COARSE FILTERING / BOTTLING

The two other composite laboratory jobs included in the application are good examples of user requirements that might not normally be picked up by standard business analysis techniques. In fact these two examples are derived from a checklist used at the pilot site. They differ in the number of tests to be performed and are used as a means of taking a snapshot of the wine to see what needs to be done to it before either being filtered or bottled. The tests are...

	Pre-Coarse Filtering	Pre Bottling
Alcohol	✓	
Free and Total SO_2	✓	✓
pH & TA	✓	✓
Residual Sugar	✓	
Volatile Acidity	✓	
Malic Acid	✓	
Cold Stability	✓	
Heat Stability	✓	
N.T.U (Turbidity)	✓	✓
Filterability		✓
Dissolved Oxygen	✓	✓
CO_2	✓	✓
Temperature	✓	✓
TasteTrial	✓	✓

Table 24. Pre bottling job details

Both jobs use the same dialog (Figure 91) when they are generated, and this dialog acts as a potential generic dialog for all composite laboratory jobs.



Pre Bottling Analyses

08WBSHIR in 90C

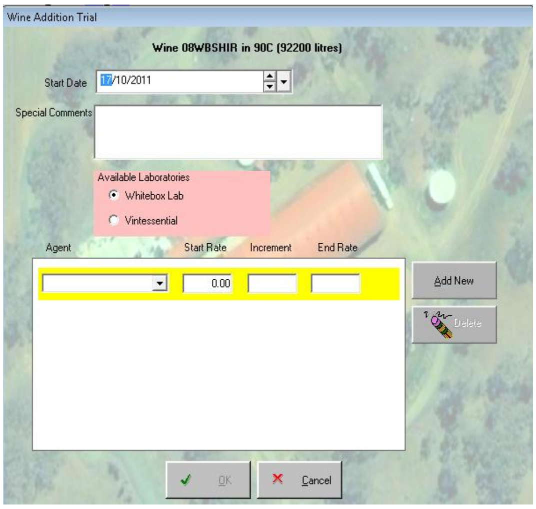
Start Date: 28/09/2011

Laboratory: Whitebox Lab

Figure 91. Pre-bottling Analsis Selection Dialog

13.3.4.24.3 ADDITION TRIALS

One of the other laboratory jobs that require more information is Addition/Fining Trials. Each trial also needs to specify the material being trialled and the start, end, and incremental amounts of being trialled so that the scope and range of the test can be fulfilled.



Wine Addition Trial

Wine 08WBSHIR in 90C (92200 litres)

Start Date: 17/10/2011

Special Comments:

Available Laboratories
● Whitebox Lab
○ Vintessential

Agent	Start Rate	Increment	End Rate
	0.00		

Figure 92. Wine Addition Trial Dialog

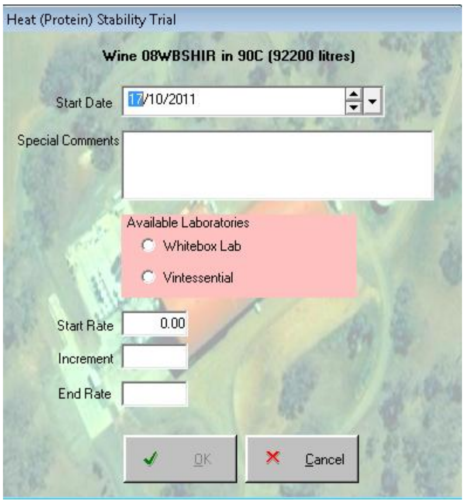
The dialog (Figure 92) allows the user to specify any number of trials with different materials for the same wine. The dimensions of amounts added are based on the rate factor for the selected additive which follows the same rules as additions to wines. (Table 25)

Rate Factor	Mass Rate	Volume Rate
1	grams per litre	millilitres per litre
100	grams per hectalitre	millilitres per hectalitre
1000	milligrams per litre	microlitres per litre

Table 25. Rate Factor conversions

These parameters allow the computation of all the additions required to samples within the laboratory screen.

13.3.4.24.4 Heat Stability Trial



Heat (Protein) Stability Trial

Wine 08WBSHIR in 90C (92200 litres)

Start Date 17/10/2011

Special Comments

Start Rate 0.00

Increment

End Rate

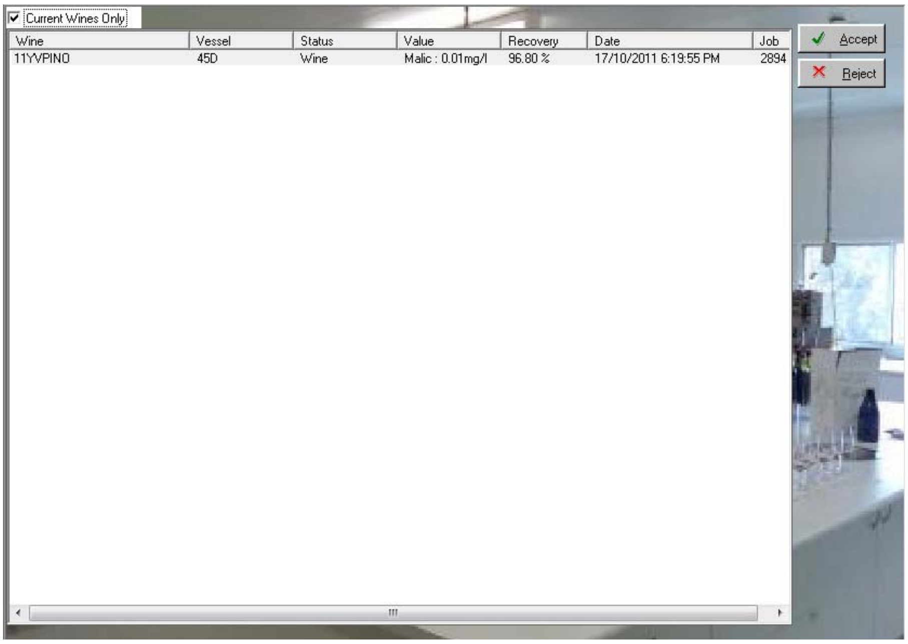
Figure 93. Heat Stability Trial Dialog

The Heat Stability trial is just a specialised bentonite addition trial that includes extra steps to filter and heat treat the samples before measuring increases in turbidity. It is therefore the same as an addition trial in that it requires the user to specify start, end and incremental rates in order to generate the worknote. (Figure 93) All the other jobs in the Laboratory menu are generated via the create wine Jobs dialog (*see 13.3.4.6 Create Wine Jobs*)

13.3.5 Reading Ratification

The ratification of analyses and the need for this type of functionality is perhaps more revealing of the nature of some winemakers than it is of the winemaking process. This was never perceived as necessary during the requirements collection phase of this project, nor was it even perceived as an issue during development of the model. It came about as a suggestion from one winemaker when it was noticed that there were some apparent systematic errors being generated in some of the laboratory results during the vintage period. Rather than simply trusting the laboratory staff to enter the results of the tests, it was felt that it was necessary to have them vetted and ratified by the winemaker before they became officially part of the records for the wine. When other winemakers were asked about this they regarded it as a good idea because it allowed obviously erroneous results to be detected, and retests to be carried-out before the error could be promulgated into the official record. The possibility that such conditions in the laboratory could exist indicates the degree of pragmatism that exists in the nature of some winemakers and their approach to making wine. It seemed that erroneous analyses were just another contingency that the winemaking process would dish-up to winemakers, particularly during vintage.

It was decided that the ratification process should be optional and switchable for each analysis test. The user ought to be able to accept or reject any analytical result, and order a retest of rejected.



☑ Current Wines Only

Wine	Vessel	Status	Value	Recovery	Date	Job
11YVPINO	45D	Wine	Malic : 0.01mg/l	96.80 %	17/10/2011 6:19:55 PM	2894

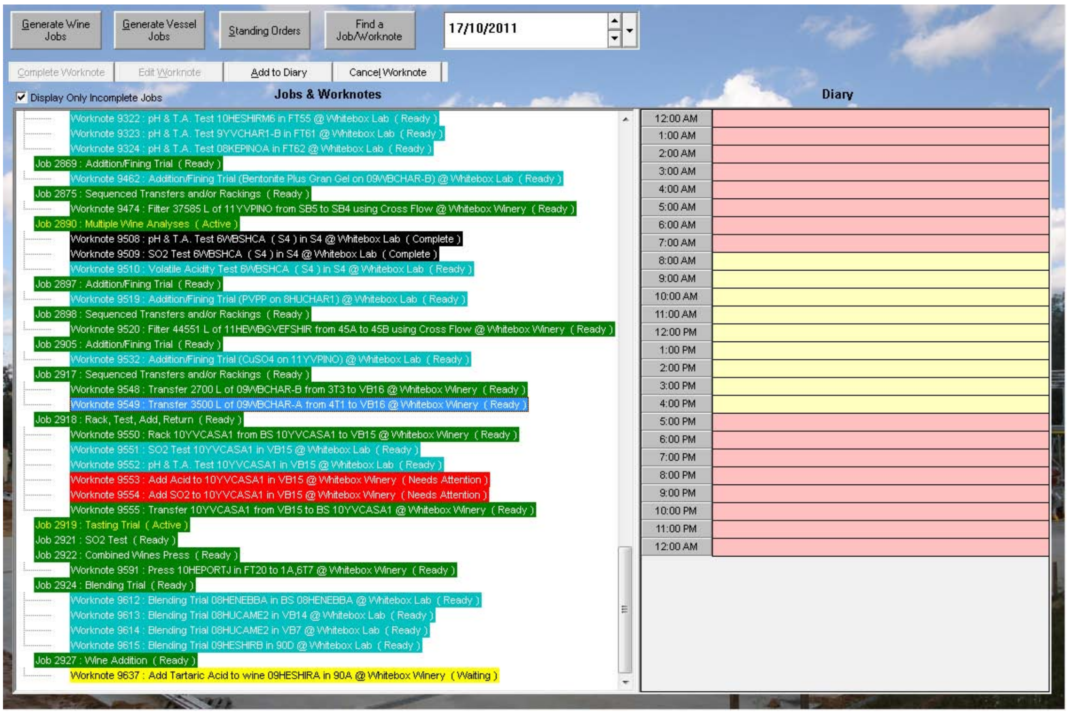
Figure 94. Reading Ratification Screen

The Ratification node was added as a child node to the Winery Map node in the main menu tree. All readings to be ratified are listed on the screen (Figure 94), along with other relevant information. This includes the recovery rate for any enzymatic assays. The date is the sample date as entered by the laboratory user, and so the time that the ratification is done has no bearing on the data and time of the reading if it is accepted. If the reading is accepted, the “AwaitingRatification” flag is set to false and the reading is saved.

In the event that a result is rejected, the reading is deleted from the collection and the interface then checks if a retest is required, and if so it resets the worknote to “Ready”. The job status is then reset, and saved so the job may reappear in the work-list. Rejecting a reading and then requesting a retest may be problematic if the test appeared in the middle of a job where the remaining tasks have been completed. This is where the system relies on the user to enter data that is not contradictory to reality (to not lie). If a retest is not possible because the wine no longer exists in the state that it is assumed in the job then the retest ought to be cancelled.

13.3.6 Work Diary

The Work Diary screen (Figure 95) is the screen of choice for most users when they become comfortable with the system and the way it operates. It is a direct reflection of the requirements as specified in *10.2.5 Work Diary.*



Worknote 9322 : pH & T.A. Test 10HESHIRM6 in FT55 @ Whitebox Lab (Ready)
Worknote 9323 : pH & T.A. Test 9YVCHAR1-B in FT61 @ Whitebox Lab (Ready)
Worknote 9324 : pH & T.A. Test 08KEPINOA in FT62 @ Whitebox Lab (Ready)
Job 2869 : Addition/Fining Trial (Ready)
Worknote 9462 : Addition/Fining Trial (Bentonite Plus Gran Gel on 09WBCHAR-B) @ Whitebox Lab (Ready)
Job 2875 : Sequenced Transfers and/or Rackings (Ready)
Worknote 9474 : Filter 37585 L of 11YVPINO from SB5 to SB4 using Cross Flow @ Whitebox Winery (Ready)
Job 2890 : Multiple Wine Analyses (Active)
Worknote 9508 : pH & T.A. Test 6WBSHCA (S4) in S4 @ Whitebox Lab (Complete)
Worknote 9509 : SO2 Test 6WBSHCA (S4) in S4 @ Whitebox Lab (Complete)
Worknote 9510 : Volatile Acidity Test 6WBSHCA (S4) in S4 @ Whitebox Lab (Ready)
Job 2897 : Addition/Fining Trial (Ready)
Worknote 9519 : Addition/Fining Trial (PVPP on 8HUCHAR1) @ Whitebox Lab (Ready)
Job 2898 : Sequenced Transfers and/or Rackings (Ready)
Worknote 9520 : Filter 44551 L of 11HEWBGVEFSHIR from 45A to 45B using Cross Flow @ Whitebox Winery (Ready)
Job 2905 : Addition/Fining Trial (Ready)
Worknote 9532 : Addition/Fining Trial (CuSO4 on 11YVPINO) @ Whitebox Lab (Ready)
Job 2917 : Sequenced Transfers and/or Rackings (Ready)
Worknote 9548 : Transfer 2700 L of 09WBCHAR-B from 3T3 to VB16 @ Whitebox Winery (Ready)
Worknote 9549 : Transfer 3500 L of 09WBCHAR-A from 4T1 to VB16 @ Whitebox Winery (Ready)
Job 2918 : Rack, Test, Add, Return (Ready)
Worknote 9550 : Rack 10YVCASA1 from BS 10YVCASA1 to VB15 @ Whitebox Winery (Ready)
Worknote 9551 : SO2 Test 10YVCASA1 in VB15 @ Whitebox Lab (Ready)
Worknote 9552 : pH & T.A. Test 10YVCASA1 in VB15 @ Whitebox Lab (Ready)
Worknote 9553 : Add Acid to 10YVCASA1 in VB15 @ Whitebox Winery (Needs Attention)
Worknote 9554 : Add SO2 to 10YVCASA1 in VB15 @ Whitebox Winery (Needs Attention)
Worknote 9555 : Transfer 10YVCASA1 from VB15 to BS 10YVCASA1 @ Whitebox Winery (Ready)
Job 2919 : Tasting Trial (Active)
Job 2921 : SO2 Test (Ready)
Job 2922 : Combined Wines Press (Ready)
Worknote 9591 : Press 10HEPORTJ in FT20 to 1A,6T7 @ Whitebox Winery (Ready)
Job 2924 : Blending Trial (Ready)
Worknote 9612 : Blending Trial 08HENEBBA in BS 08HENEBBA @ Whitebox Lab (Ready)
Worknote 9613 : Blending Trial 08HUCAME2 in VB14 @ Whitebox Lab (Ready)
Worknote 9614 : Blending Trial 08HUCAME2 in VB7 @ Whitebox Lab (Ready)
Worknote 9615 : Blending Trial 09HESHIRB in 90D @ Whitebox Lab (Ready)
Job 2927 : Wine Addition (Ready)
Worknote 9637 : Add Tartaric Acid to wine 09HESHIRA in 90A @ Whitebox Winery (Waiting)

12:00 AM
1:00 AM
2:00 AM
3:00 AM
4:00 AM
5:00 AM
6:00 AM
7:00 AM
8:00 AM
9:00 AM
10:00 AM
11:00 AM
12:00 PM
1:00 PM
2:00 PM
3:00 PM
4:00 PM
5:00 PM
6:00 PM
7:00 PM
8:00 PM
9:00 PM
10:00 PM
11:00 PM
12:00 AM

Figure 95. Work Diary Screen

The main feature of this screen is yet another tree control. This time the main nodes represent jobs, and the child nodes represent worknotes. The colour of the background and text of each node represent the status of the object.

Waiting worknote
Worknote or Job needs attention
Ready Job or worknote
Ready Laboratory worknote
Active Job
Complete
Cancelled

The screen also has the following push buttons...

Generate Wine Jobs	Displays the “Create Wine Jobs” dialog (*13.3.4.6*)
Generate Vessel Jobs	Displays the “Create Vessel Jobs” dialog (*13.3.4.5)*
Standing Orders	Displays “Standing Orders” dialog *(13.3.6.1*)
Find Job/Worknote	Displays “Find Job/Worknote” dialog (*13.3.6.2*)

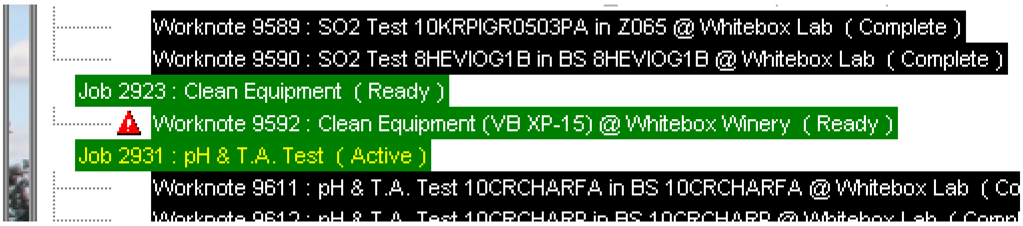
Near the command buttons is a check box to “Display Only Incomplete Jobs”. This is selected by default and causes only relevant jobs and worknotes to be displayed, and excludes cancelled or completed jobs. The date control at the top of the screen causes the while screen to be refreshed when it is changed. If the date set is in the past then only jobs completed on that date are displayed. If the date is set to the future, then only jobs booked for that date are displayed. In these two cases the “Display Only Incomplete Jobs” checkbox is invisible.

As well as these features, the following buttons (Table 26) appear when a job or worknote is selected from the tree. The same options are available from a popup menu when a node is right-clicked.

Edit Job	Displays the same dialog that was used to create the job, in EDIT mode. Only available if the original job was created using a dialog (if the Job type has NoDialog flag set to false)
Cancel Job	Cancels the job and all its worknotes.
View Job Report	Displays the Job Viewer dialog (13.2.5 Other System Wide Dialogs)
Confirm Job	Used to turn standing orders into confirmed and ready jobs.
Complete Worknote	Displays a dialog to allow the worknote to be completed, unless this is not required. This dialog will differ depending on the worknote type. The worknote is completed and the status of its parent job is reset. The Completion Date and time is set when the Completion Date dialog (see *13.2.5 Other System Wide Dialogs*) is OK’d immediately after the completion process.
Edit Worknote	Some individual worknotes can also be edited if the “Editable” flag is set on worknote type object. If so, each worknote type has its own edit screen.
Add to Diary	Copies the details of the selected worknote to the selected diary space
Cancel Worknote	Cancels the individual worknote and causes the parent job status to be reset.

Table 26. Buttons for processing Jobs and Worknotes

There is an additional popup menu option for worknotes called “Set Priority” to allow the “HighPriority” flag to be set on the worknote object. There are two options; High and Low; the current setting being disabled. When a high priority worknote is displayed in the tree, a red exclamation mark is displayed next to it to highlight its importance. (see below)



Worknote 9589 : SO2 Test 10KRPIGR0503PA in Z065 @ Whitebox Lab (Complete)
Worknote 9590 : SO2 Test 8HEVIOG1B in BS 8HEVIOG1B @ Whitebox Lab (Complete)
Job 2923 : Clean Equipment (Ready)
Worknote 9592 : Clean Equipment (VB XP-15) @ Whitebox Winery (Ready)
Job 2931 : pH & T.A. Test (Active)
Worknote 9611 : pH & T.A. Test 10CRCHARFA in BS 10CRCHARFA @ Whitebox Lab (Co
Worknote 9612 : pH & T.A. Test 10CRCHARP in BS 10CRCHARP @ Whitebox Lab (Compl

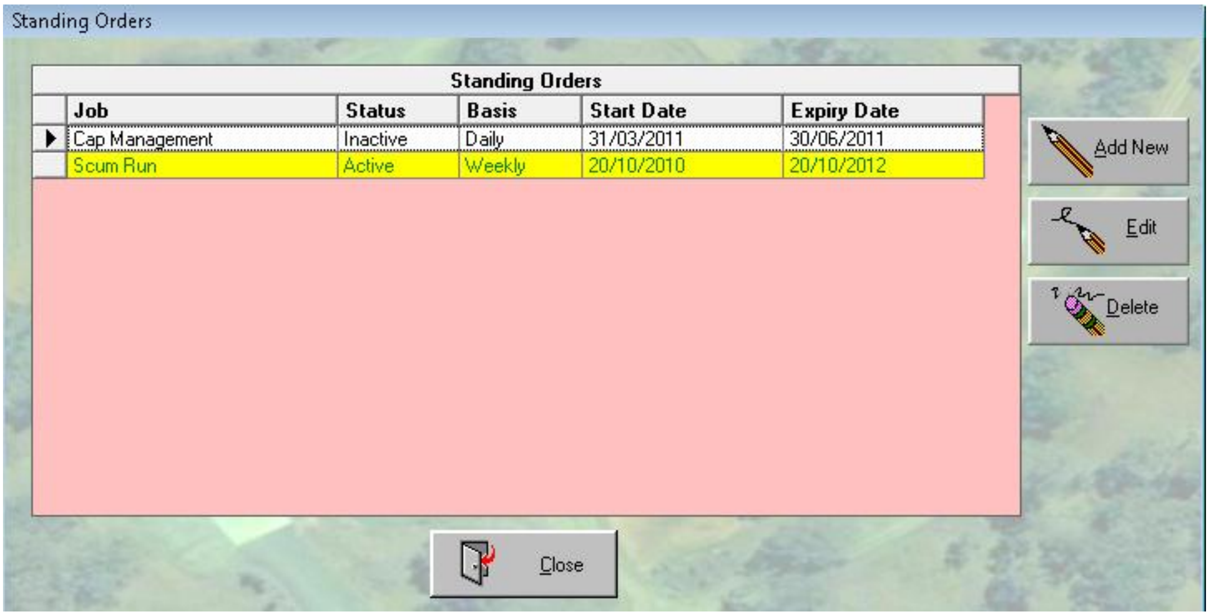
Figure 96. High Priority Worknote example

High priority lab worknotes also receive special highlighting treatment when displayed (see *13.7 Laboratories*).

The other main part of the Work Diary screen is the Diary. The manifestation of this concept in the application is rudimentary. Diary entries are user specific so personal entries are not displayed for all to see. The user need only select a slot on the diary then enter text and it will be recorded immediately. If the entry is cleared, the diary entry object is deleted.

13.3.6.1 Standing Orders

The standing orders screen (Figure 97) is also a direct reflection of the requirements process. The dialog allows the user to maintain a list of standing orders, and the ability to activate and de-activate them when required. As described earlier, a standing order can be thought of as a template for a job that is offered up periodically by the system so that the user can either ignore it or confirm it.



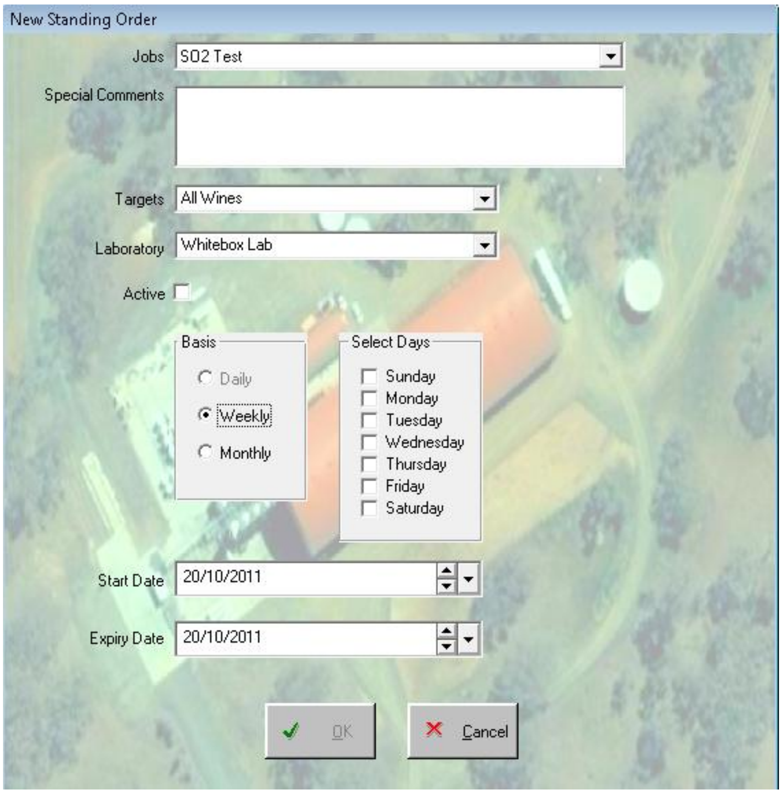
Standing Orders

Standing Orders

Job	Status	Basis	Start Date	Expiry Date
Cap Management	Inactive	Daily	31/03/2011	30/06/2011
Scum Run	Active	Weekly	20/10/2010	20/10/2012

Figure 97. Standing Orders Dialog

When a user loads the Work Diary and has it locked out, and any subsequent interaction with the dialog causes the system to go through all the standing orders and work out whether any new jobs need to be displayed. Any jobs previously created from standing orders for that day are recognisable from the "StandingOrder" parameter which gets set when a standing order is confirmed and the job is created. Standing orders appearing in the Work Diary are recognisable by the red background, and the negative Job ID.



New Standing Order

Jobs: SO2 Test

Special Comments:

Targets: All Wines

Laboratory: Whitebox Lab

Active ☐

Basis: ○ Daily ● Weekly ○ Monthly

Select Days: ☐ Sunday ☐ Monday ☐ Tuesday ☐ Wednesday ☐ Thursday ☐ Friday ☐ Saturday

Start Date: 20/10/2011

Expiry Date: 20/10/2011

Figure 98. New Standing Order Dialog

The "New Standing Order" dialog (Figure 98) that appears when the "Add New" button is pressed, provides an insight into the anatomy of a standing order. The list box at the top includes only job types that are flagged as "StandingOrder". The targets list contains the following options...

- All Wines
- Reds
- Whites
- All Ferments
- Red Ferments
- White Ferments

The Basis option selection causes the dialog to display a corresponding set of qualifiers for that basis. If "Daily" is selected, then it displays a set of times in 3 hour increments. "Weekly" displays a list of the days of the week, and “Monthly” displays the list of months with a date field to specifiy the day of the month that the standing order is to be offered. The Start Date and Expiry Date are self-explanatory. The standing order can be activated or de-activated by simply checking or un-checking the “Active” checkbox. The same dialog is used to edit an existing standing order selected from the list.

13.3.6.2 Find Job/Worknote

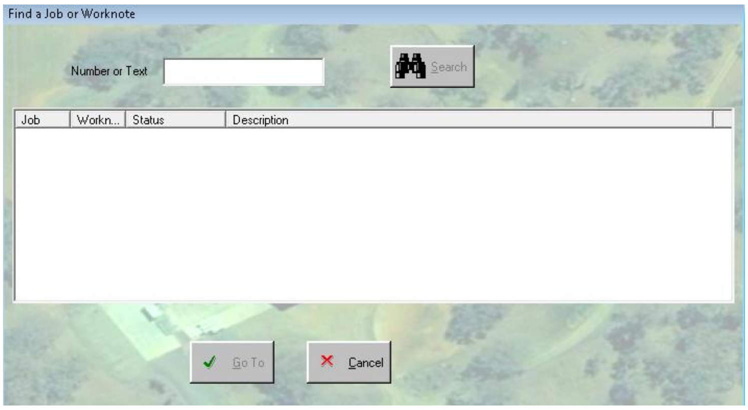


Figure 99. Find a Job/Worknote Dialog

This dialog (Figure 99) is a clone of the “Find a Vessel” and “Find a Wine“ dialogs. The major difference is that it allows the input of any text or number, and all job numbers, worknote numbers and titles are searched to find jobs or worknotes that fit the criteria. All the jobs and worknotes that match appear in the list below the entry field, and when one is selected and the “Go To” button pressed, the date is automatically selected on the Work Diary screen to display the desired job/worknote.

13.3.7 Monitoring and Algorithms

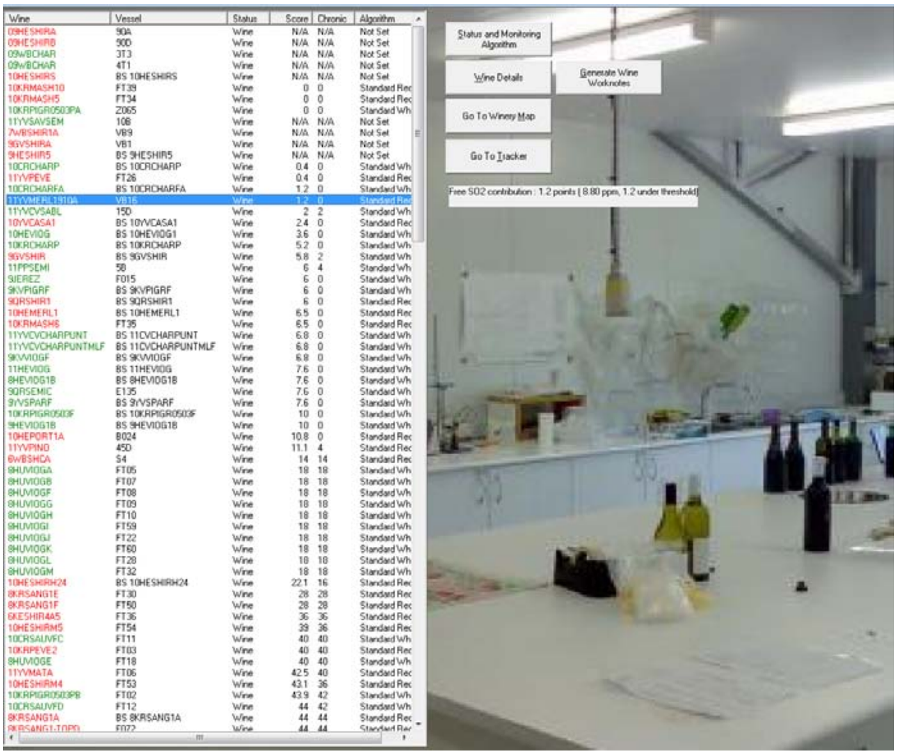
The Monitoring screen provides the winemaker with a means of highlighting wines that require attention according to criteria they can set themselves. The wine object has a Monitoring Algorithm property that is a key to the Monitoring Algorithms collection for the winery. The “MonitorScore” method for the wine object generates a score based on accumulated points from tests linked to the nominated algorithm. The tests themselves have a generic structure that allow any reading for a wine to be used as a parameter to be measured and compared in one of three ways.

1. Days since last measurement after certain number of days
2. Amount over a specified level
3. Amount Under a specified level

Pseudo-code for the MonitorScore method in the wine object looks like this...

```
If the Algorithm is set
      Set the score to 0
      Set the Score description to blank
      For each test in the algorithm
            Retrieve the reading for the wine that this test is using
            If there is a reading
                  If test is type 1
                        Determine the age of the reading
                        If this is greater than the threshold
                              +Score = (excess age/Increment) X Points
                              Add description to score description
                  If test is type 2
                        If reading is greater than the threshold
                              +Score = (excess/Increment) X Points
                              Add description to score description
                  If test is type 3
                        If reading is less than the threshold
                              +Score = (deficit/Increment) X Points
                              Add description to score description
            Otherwise
                  Add 100 points to score for no reading
```

The score and score description results are displayed on the Monitoring screen (Figure 100) as well as the chronic score which is the part of the score attributable to type 1 tests. The list can be sorted by any column and the buttons to the right of the list help the user navigate to other parts of the system to attempt to rectify any issues with the wine that might be causing a large score.



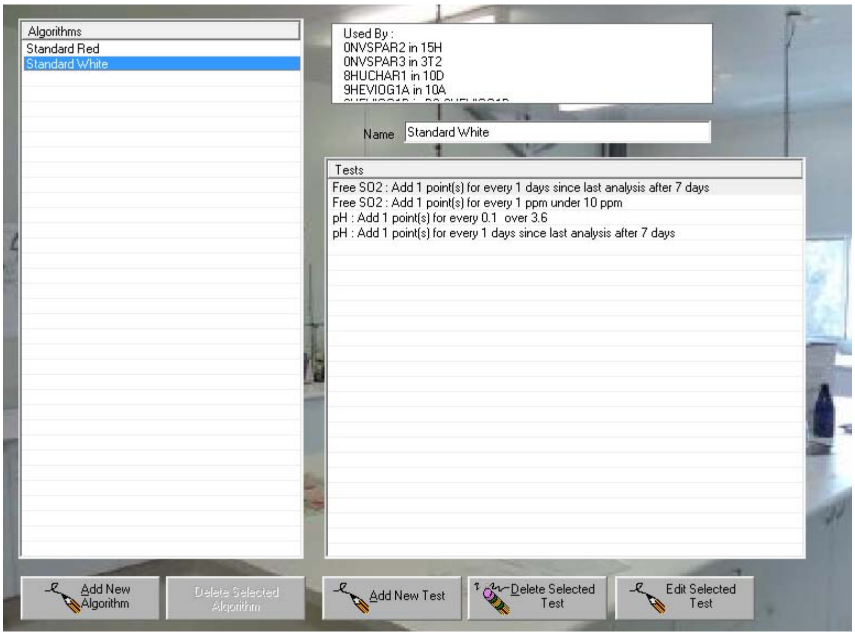
Wine	Vessel	Status	Score	Chronic	Algorithm
09HESHIRA	90A	Wine	N/A	N/A	Not Set
09HESHIRB	90D	Wine	N/A	N/A	Not Set
09WBCHAR	3T3	Wine	N/A	N/A	Not Set
09WBCHAR	4T1	Wine	N/A	N/A	Not Set
10HESHIRS	BS 10HESHIRS	Wine	N/A	N/A	Not Set
10KRMASH10	FT39	Wine	0	0	Standard Rec
10KRMASH5	FT34	Wine	0	0	Standard Rec
10KRPIGR0503PA	Z065	Wine	0	0	Standard Wh
11YVSAVSEM	10B	Wine	N/A	N/A	Not Set
7WBSHIR1A	VB9	Wine	N/A	N/A	Not Set
9GVSHIRA	VB1	Wine	N/A	N/A	Not Set
9HESHIR5	BS 9HESHIR5	Wine	N/A	N/A	Not Set
10CRCHARP	BS 10CRCHARP	Wine	0.4	0	Standard Wh
11YVPEVE	FT26	Wine	0.4	0	Standard Rec
10CRCHARFA	BS 10CRCHARFA	Wine	1.2	0	Standard Wh
11YVMERL1910A	VB16	Wine	1.2	0	Standard Rec
11YVCVSABL	15D	Wine	2	2	Standard Wh
10YVCASA1	BS 10YVCASA1	Wine	2.4	0	Standard Rec
10HEVIOG	BS 10HEVIOG1	Wine	3.6	0	Standard Wh
10KRCHARP	BS 10KRCHARP	Wine	5.2	0	Standard Wh
9GVSHIR	BS 9GVSHIR	Wine	5.8	2	Standard Wh
11PPSEMI	5B	Wine	6	4	Standard Wh
9JEREZ	F015	Wine	6	0	Standard Wh
9KVPIGRF	BS 9KVPIGRF	Wine	6	0	Standard Wh
9QRSHIR1	BS 9QRSHIR1	Wine	6	0	Standard Rec
10HEMERL1	BS 10HEMERL1	Wine	6.5	0	Standard Rec
10KRMASH6	FT35	Wine	6.5	0	Standard Rec
11YVCVCHARPUNT	BS 11CVCHARPUNT	Wine	6.8	0	Standard Wh
11YVCVCHARPUNTMLF	BS 11CVCHARPUNTMLF	Wine	6.8	0	Standard Wh
9KVVIOGF	BS 9KVVIOGF	Wine	6.8	0	Standard Wh
11HEVIOG	BS 11HEVIOG	Wine	7.6	0	Standard Wh
8HEVIOG1B	BS 8HEVIOG1B	Wine	7.6	0	Standard Wh
9QRSEMIC	E135	Wine	7.6	0	Standard Wh
9YVSPARF	BS 9YVSPARF	Wine	7.6	0	Standard Wh
10KRPIGR0503F	BS 10KRPIGR0503F	Wine	10	0	Standard Wh
9HEVIOG1B	BS 9HEVIOG1B	Wine	10	0	Standard Wh
10HEPORT1A	B024	Wine	10.8	0	Standard Rec
11YVPINO	45D	Wine	11.1	4	Standard Rec
6WBSHCA	S4	Wine	14	14	Standard Rec
8HUVIOGA	FT05	Wine	18	18	Standard Wh
8HUVIOGB	FT07	Wine	18	18	Standard Wh
8HUVIOGF	FT08	Wine	18	18	Standard Wh
8HUVIOGG	FT09	Wine	18	18	Standard Wh
8HUVIOGH	FT10	Wine	18	18	Standard Wh
8HUVIOGI	FT59	Wine	18	18	Standard Wh
8HUVIOGJ	FT22	Wine	18	18	Standard Wh
8HUVIOGK	FT60	Wine	18	18	Standard Wh
8HUVIOGL	FT28	Wine	18	18	Standard Wh
8HUVIOGM	FT32	Wine	18	18	Standard Wh
10HESHIRH24	BS 10HESHIRH24	Wine	22.1	16	Standard Rec
8KRSANG1E	FT30	Wine	28	28	Standard Rec
8KRSANG1F	FT50	Wine	28	28	Standard Rec
6KESHIR4A5	FT36	Wine	36	36	Standard Rec
10HESHIRM5	FT54	Wine	39	36	Standard Rec
10CRSAUVFC	FT11	Wine	40	40	Standard Wh
10KRPEVE2	FT03	Wine	40	40	Standard Rec
8HUVIOGE	FT18	Wine	40	40	Standard Wh
11YVMATA	FT06	Wine	42.5	40	Standard Rec
10HESHIRM4	FT53	Wine	43.1	36	Standard Rec
10KRPIGR0503PB	FT02	Wine	43.9	42	Standard Wh
10CRSAUVFD	FT12	Wine	44	42	Standard Wh
8KRSANG1A	BS 8KRSANG1A	Wine	44	44	Standard Rec
8KRSANG1-TOPD	F072	Wine	44	44	Standard Rec

Free SO2 contribution : 1.2 points [8.80 ppm, 1.2 under threshold]

Figure 100. Wine Monitoring Screen

It should be noted from the pseudo-code that if a wine has no reading for a specified test then a default 100 points is added to the score.

The Algorithms node on the main menu tree displays the algorithms defined for the winery and allows the user to create new, edit existing, and delete algorithms.



Algorithms
Standard Red
Standard White

Used By :
0NVSPAR2 in 15H
0NVSPAR3 in 3T2
8HUCHAR1 in 10D
9HEVIOG1A in 10A

Name: Standard White

Tests
Free SO2 : Add 1 point(s) for every 1 days since last analysis after 7 days
Free SO2 : Add 1 point(s) for every 1 ppm under 10 ppm
pH : Add 1 point(s) for every 0.1 over 3.6
pH : Add 1 point(s) for every 1 days since last analysis after 7 days

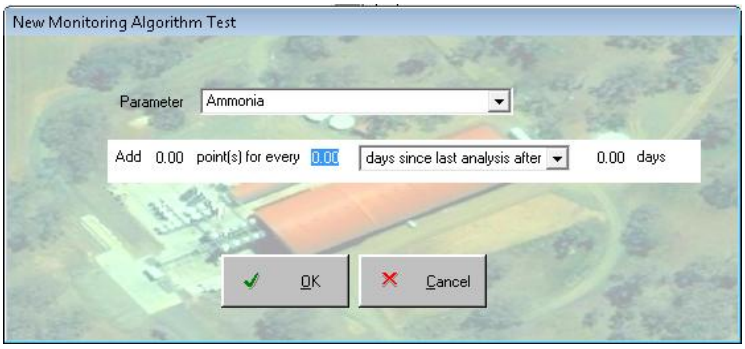
Figure 101. Monitoring Algorithm Dialog

This screen (Figure 101) was originally going to be within the administration application for the system rather than being part of the main application. However it was found that manipulating these algorithms was more common than originally thought and it made sense to include it in the main application and available from the main menu as a child node of the main Monitoring screen.

The command buttons underneath the list of algorithms allow the user to manipulate algorithms and their constituent tests. The “Add New Test” button displays the “New Test” dialog that allows the user to specify all the test object properties which include...

Parameter	Drop down list containing all possible reading types that can be tested
TestType	Drop down list with appropriate text for types 1,2,and 3
TestIncrement	The incremental amount of units that add test points to the score. This is the second number field in the test description.
TestThreshold	The critical point at which scoring begins. This is the third number field in the test description.
TestPoints	The number of points added to the score for each increment beyond the threshold. This is the first number field in the test description.

Table 27. Monitoring Algorithm Test Properties


New Monitoring Algorithm Test

Parameter: Ammonia

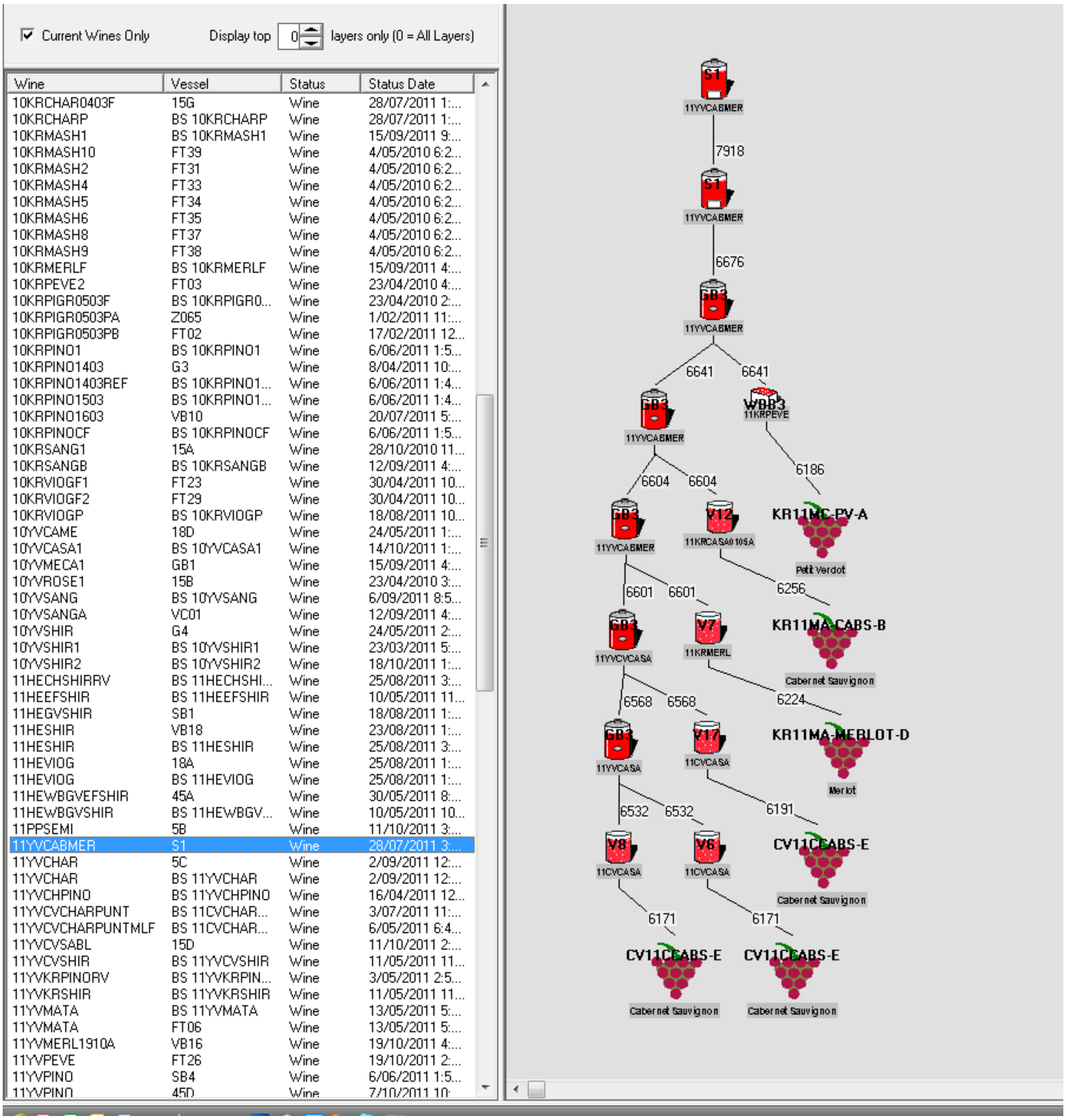
Add 0.00 point(s) for every 0.00 days since last analysis after 0.00 days

Figure 102. New Monitoring Algorithm Dialog

This dialog (Figure 102) is also used in "Edit" mode to modify existing tests using the "Edit Selected Test" button.

13.3.8 Wine Tracker

The Tracker screen (Figure 103) was originally envisaged as one of the most important outputs of the data model structure for wines. The design of the model always had the tracker function on the forefront of consideration throughout the process. The simple way that wines are related to each other, and the way that each wine's history is easily accessible and derivable make it a relatively simple concept to build a history and ancestry of a wine. The difficulty comes in presenting what can be quite a complex family tree in a way that can be understood.



☑ Current Wines Only   Display top 0 layers only (0 = All Layers)

Wine	Vessel	Status	Status Date
10KRCHAR0403F	15G	Wine	28/07/2011 1:...
10KRCHARP	BS 10KRCHARP	Wine	28/07/2011 1:...
10KRMASH1	BS 10KRMASH1	Wine	15/09/2011 9:...
10KRMASH10	FT39	Wine	4/05/2010 6:2...
10KRMASH2	FT31	Wine	4/05/2010 6:2...
10KRMASH4	FT33	Wine	4/05/2010 6:2...
10KRMASH5	FT34	Wine	4/05/2010 6:2...
10KRMASH6	FT35	Wine	4/05/2010 6:2...
10KRMASH8	FT37	Wine	4/05/2010 6:2...
10KRMASH9	FT38	Wine	4/05/2010 6:2...
10KRMERLF	BS 10KRMERLF	Wine	15/09/2011 4:...
10KRPEVE2	FT03	Wine	23/04/2010 4:...
10KRPIGR0503F	BS 10KRPIGR0...	Wine	23/04/2010 2:...
10KRPIGR0503PA	Z065	Wine	1/02/2011 11:...
10KRPIGR0503PB	FT02	Wine	17/02/2011 12...
10KRPINO1	BS 10KRPINO1	Wine	6/06/2011 1:5...
10KRPINO1403	G3	Wine	8/04/2011 10:...
10KRPINO1403REF	BS 10KRPINO1...	Wine	6/06/2011 1:4...
10KRPINO1503	BS 10KRPINO1...	Wine	6/06/2011 1:4...
10KRPINO1603	VB10	Wine	20/07/2011 5:...
10KRPINOCF	BS 10KRPINOCF	Wine	6/06/2011 1:5...
10KRSANG1	15A	Wine	28/10/2010 11...
10KRSANGB	BS 10KRSANGB	Wine	12/09/2011 4:...
10KRVIOGF1	FT23	Wine	30/04/2011 10...
10KRVIOGF2	FT29	Wine	30/04/2011 10...
10KRVIOGP	BS 10KRVIOGP	Wine	18/08/2011 10...
10YVCAME	18D	Wine	24/05/2011 1:...
10YVCASA1	BS 10YVCASA1	Wine	14/10/2011 1:...
10YVMECA1	GB1	Wine	15/09/2011 4:...
10YVROSE1	15B	Wine	23/04/2010 3:...
10YVSANG	BS 10YVSANG	Wine	6/09/2011 8:5...
10YVSANGA	VC01	Wine	12/09/2011 4:...
10YVSHIR	G4	Wine	24/05/2011 2:...
10YVSHIR1	BS 10YVSHIR1	Wine	23/03/2011 5:...
10YVSHIR2	BS 10YVSHIR2	Wine	18/10/2011 1:...
11HECHSHIRRV	BS 11HECHSHI...	Wine	25/08/2011 3:...
11HEEFSHIR	BS 11HEEFSHIR	Wine	10/05/2011 11...
11HEGVSHIR	SB1	Wine	18/08/2011 1:...
11HESHIR	VB18	Wine	23/08/2011 1:...
11HESHIR	BS 11HESHIR	Wine	25/08/2011 3:...
11HEVIOG	18A	Wine	25/08/2011 1:...
11HEVIOG	BS 11HEVIOG	Wine	25/08/2011 1:...
11HEWBGVEFSHIR	45A	Wine	30/05/2011 8:...
11HEWBGVSHIR	BS 11HEWBGV...	Wine	10/05/2011 10...
11PPSEMI	5B	Wine	11/10/2011 3:...
11YVCABMER	S1	Wine	28/07/2011 3:...
11YVCHAR	5C	Wine	2/09/2011 12:...
11YVCHAR	BS 11YVCHAR	Wine	2/09/2011 12:...
11YVCHPINO	BS 11YVCHPINO	Wine	16/04/2011 12...
11YVCVCHARPUNT	BS 11CVCHAR...	Wine	3/07/2011 11:...
11YVCVCHARPUNTMLF	BS 11CVCHAR...	Wine	6/05/2011 6:4...
11YVCVSABL	15D	Wine	11/10/2011 2:...
11YVCVSHIR	BS 11YVCVSHIR	Wine	11/05/2011 11...
11YVKRPINORV	BS 11YVKRPIN...	Wine	3/05/2011 2:5...
11YVKRSHIR	BS 11YVKRSHIR	Wine	11/05/2011 11...
11YVMATA	BS 11YVMATA	Wine	13/05/2011 5:...
11YVMATA	FT06	Wine	13/05/2011 5:...
11YVMERL1910A	VB16	Wine	19/10/2011 4:...
11YVPEVE	FT26	Wine	19/10/2011 2:...
11YVPINO	SB4	Wine	6/06/2011 1:5...
11YVPINO	45D	Wine	7/10/2011 10:

Figure 103. Wine Tracker Screen

All the current wines are displayed in the list but there is an option to include all the wines in the system, to allow the user to track wines that have been bottled or dumped. When a wine is selected, the system clears the screen, generates the tracking data for the selected wine, and then presents is on the screen. This process can take anywhere from a couple of seconds up to a minute for some wines with more complex histories. For this reason, the user can restrict the number of levels or "layers" of history to track back by selecting the layer count on the spin box at the top of the screen. A value of "0" means that there is no restriction. This value is stored in the system registry so it does not reset each time the application is started. The list can be sorted by any of the columns.

13.3.8.1 Generating the data

The process of generating the tracker data involves recursively looping through the wine information in the data model and dumping it into a database table, then reloading the sorted data into so-called Tracker Items objects of the data model (see 12.4.6 Tracker Items*)*. This method is not ideal but it simplifies the process by utilising the database system's ability to sort data far faster than the data model. The tracker screen is available to all users at the same time so the data dumped to the database is also tagged with the id of the user.

This generation process uses a recursive function called TrackerDump. This function first tries to identify any "reset volume" jobs for the wine or any other jobs that would result in the wine changing vessel but still remaining current. These actions result in new layers being added to the tracker data but no change in the id of the wine.

The function then loops through all the source wines for this wine, and calls itself recursively. If the wine has no source wines then it must have come from source fruit or it was created prior to the installation of the system by the Super Admin function "Create Wine". In these cases, the source fruit is added to the tracker dump. The function keeps track of which layer is relevant by passing it in as a parameter to the TrackerDump function

When the function bubbles back up to the top, the data is resorted into layer order and loaded into memory as Trackeritems

13.3.8.2 Presenting the Data

The first thing to notice on this screen is that the hierarchy of the wine is a pyramid that has the latest entity at the top and all the predecessors appear below. This is a convention that avoids the need to scroll down on the screen to see the latest state of the wine. As the tracker is laid out, the system centres the pyramid by working out the maximum number of entities that will appear in any single layer, and finding a mid point.

Each TrackerItem object is an instance of the wine or one of its source wines at a particular time. They are linked to each other by the fact that a worknote caused one instance to become another. This is why the tracker screen has a worknote number between each entity. The tracker screen uses objects cloned from the Winery Map, so many of the features of these objects are maintained such as the tooltip popup box that summarises the wine. If the user double-clicks on a wine, the Wine Readings dialog appears (*13.3.4.11 More Readings*). Similarly, if the user double-clicks on the worknote number linking two wines, the report for the parent job for that worknote is displayed in the "View Job" dialog (see *13.2.5 Other System Wide Dialogs*)

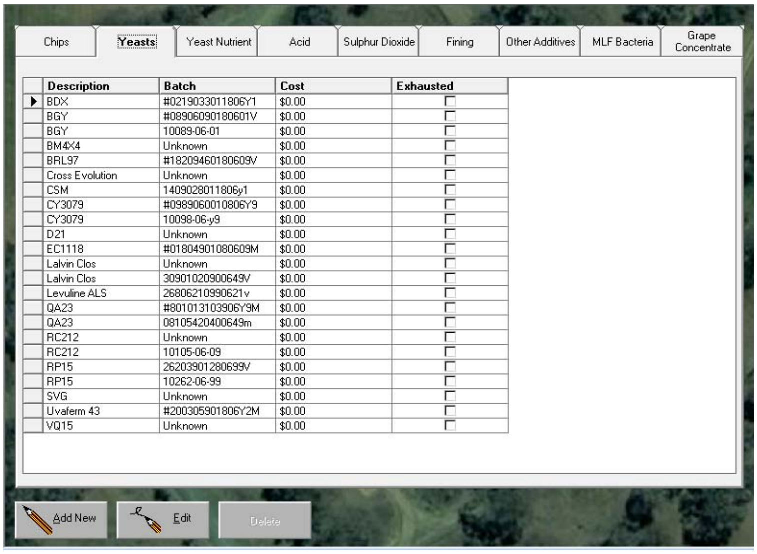
The example shown in the diagram is a relatively simple case, but even this can highlight some of the problems encountered in finding a way to present the information logically and simply. The events leading up to the final wine, as they occurred, are as follows...

Date	Worknote	Event
28/4/2011	6171	20 Tonnes Cabernet sauvignon fruit is processed into vessels V6 and V8
29/4/2011	6186	1.5 Tonnes Petit verdot fruit processed into vessel WBB1
29/4/2011	6191	9 Tonnes Cabernet sauvignon fruit is processed into vessel V17. Note that this fruit comes from the same Plan batch as that in worknote 6171; it just arrived the next day.
30/4/2011	6224	Merlot fruit is processed into vessel V7
1/5/2011	6256	7 Tonnes Cabernet sauvignon fruit is processed into vessel V12
9/5/2011	6532	Pressed V6 and V8 into GB3
10/5/2011	6568	Pressed V17 into GB3
11/5/2011	6601	Pressed V7 into GB3
11/5/2011	6604	Pressed V12 into GB3
13/5/2011	6641	Pressed WBB1 into GB3 (and another vessel)
18/5/2011	6676	Racked wine into S1
28/7/2011	7918	80 Litres used to top barrels

Here we see the Petit Verdot fruit is processed on the 29th April but does not get blended into the final wine until the 13th May. So, although the sequence can be derived, it is not immediately apparent. Similarly, only a small amount of Petit Verdot went into the blend, but this is not apparent in the tracker because it does not show a link to the other wine. These points may seem trivial, but this is just a simple case, and other wines can become very complicated over time as more and more actions happen to them.

The problem here is not with the model but with the interface. This, therefore, becomes a matter for further research rather than an issue to be resolved in this context.

13.3.9 Materials Storage (Additives Stock)

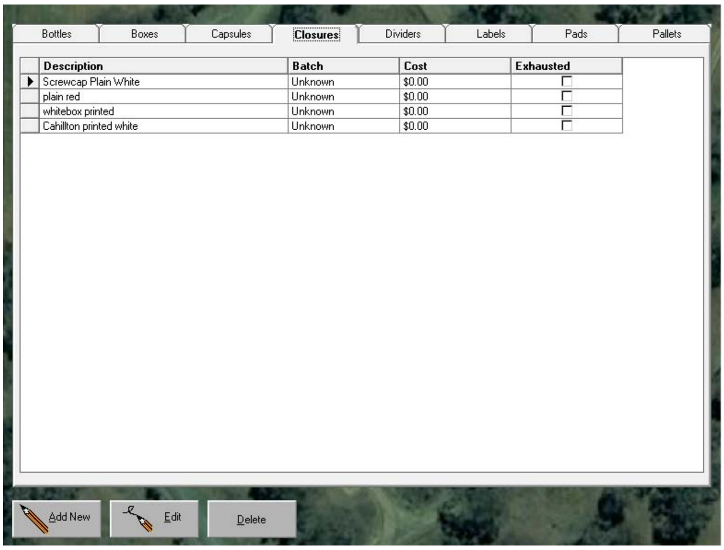


Description	Batch	Cost	Exhausted
BDX	#0219033011806Y1	$0.00	□
BGY	#08906090180601V	$0.00	□
BGY	10089-06-01	$0.00	□
BM4X4	Unknown	$0.00	□
BRL97	#18209460180609V	$0.00	□
Cross Evolution	Unknown	$0.00	□
CSM	1409028011806y1	$0.00	□
CY3079	#0989060010806Y9	$0.00	□
CY3079	10098-06-y9	$0.00	□
D21	Unknown	$0.00	□
EC1118	#01804901080609M	$0.00	□
Lalvin Clos	Unknown	$0.00	□
Lalvin Clos	30901020900649V	$0.00	□
Levuline ALS	26806210990621v	$0.00	□
QA23	#801013103906Y9M	$0.00	□
QA23	08105420400649m	$0.00	□
RC212	Unknown	$0.00	□
RC212	10105-06-09	$0.00	□
RP15	26203901280699V	$0.00	□
RP15	10262-06-99	$0.00	□
SVG	Unknown	$0.00	□
Uvaferm 43	#200305901806Y2M	$0.00	□
VQ15	Unknown	$0.00	□

Figure 104. Materials Stocks Screen

This screen (Figure 104) is used to manage details of the additives held in stock, including batch numbers and unit costs. This information is then used for completing worknotes that require batch numbers of any materials used in the worknote. The exhausted flag on each item excludes that item from being displayed in the batch lists. The tabs across the top divide the screen into the various additive types. The dialog directly updates the AdditivesStocks collection of the Additives objects in the Winery object.

13.3.10 Dry Goods



Description	Batch	Cost	Exhausted
Screwcap Plain White	Unknown	$0.00	□
plain red	Unknown	$0.00	□
whitebox printed	Unknown	$0.00	□
Cahillton printed white	Unknown	$0.00	□

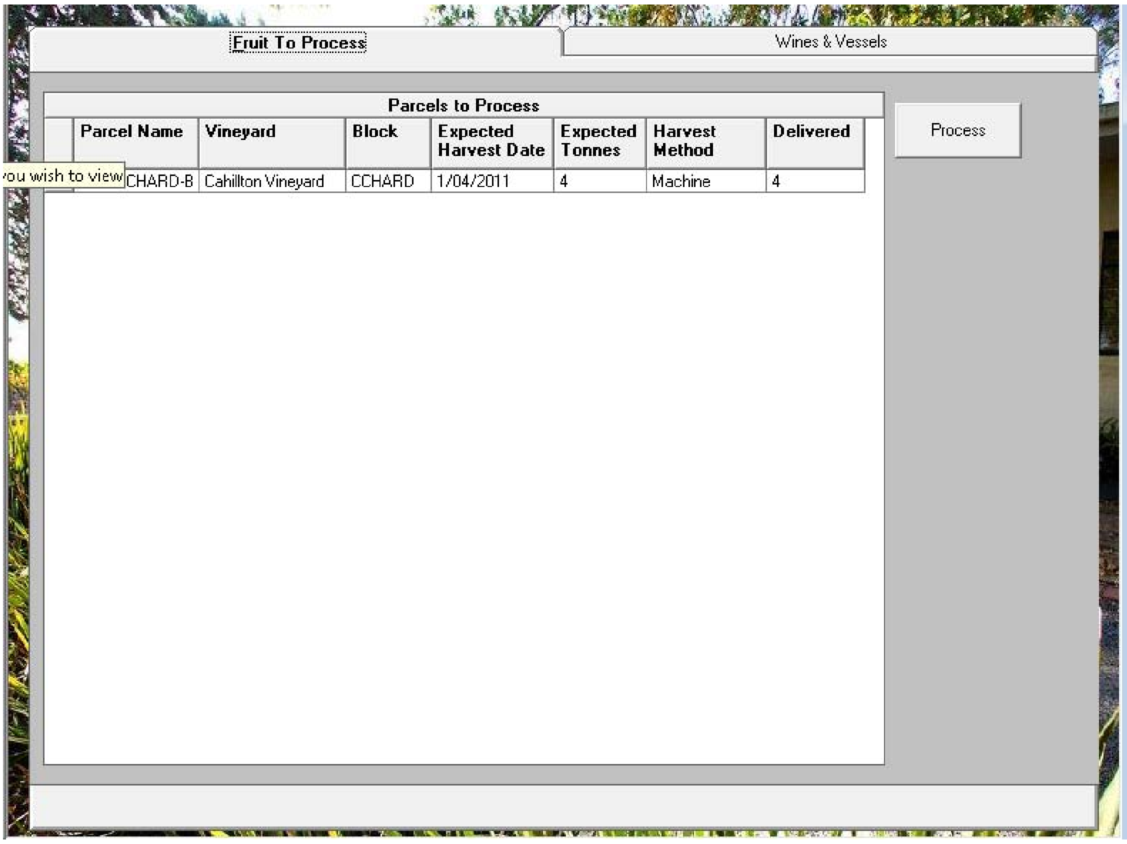
Figure 105. Dry Goods Stock Screen

This is a clone of the Additives Stock dialog, but refers to the dry goods in the winery rather than additives.

13.4 The Unmanaged Winery

An unmanaged winery can be part of any enterprise. If the enterprise contracts or supplies fruit to a winery it does not directly control, it may still want to keep records (albeit at a higher and less granular level) of the activities in the winery.

In essence, the unmanaged winery only needs to manage fruit processing and allow basic records of wines to be maintained. Inevitably, these records will not be available until well after the events occurred. However, they must be entered before the wine is moved from the winery. This is because when a wine is moved, it effectively becomes a new wine, and so information can no longer be attached to a non-current wine; by design.

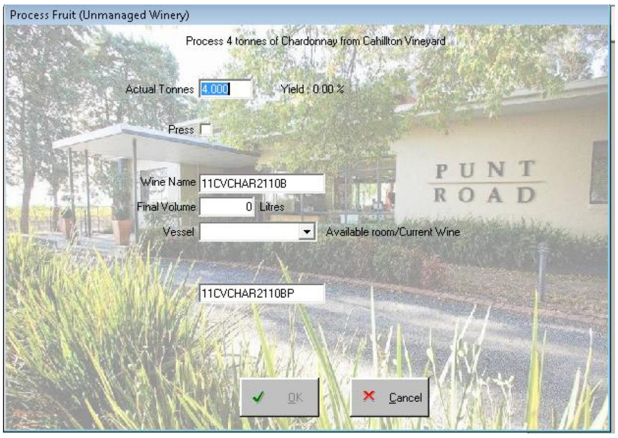


Parcels to Process

Parcel Name	Vineyard	Block	Expected Harvest Date	Expected Tonnes	Harvest Method	Delivered
CHARD-B	Cahillton Vineyard	CCHARD	1/04/2011	4	Machine	4

Figure 106. Unmanaged Winery - Fruit To Process Screen

So the unmanaged winery screen (Figure 106) is fairly simple in design, having only two major tabs; one for fruit processing and the other for wine management. The Fruit processing tab is an abbreviated version of the Weighbridge screen on the managed vineyard. It displays all the fruit parcels that have been designated for processing at the selected winery. The Process button causes the Process Fruit dialog (Figure 107) to be displayed with details of the selected fruit parcel.



Process Fruit (Unmanaged Winery)

Process 4 tonnes of Chardonnay from Cahillton Vineyard

Actual Tonnes 4.000 Yield : 0.00 %

Press □

Wine Name 11CVCHAR2110B

Final Volume 0 Litres

Vessel Available room/Current Wine

11CVCHAR2110BP

Figure 107. Unmanaged Winery - Process Fruit Dialog

This dialog allows the basic wine processing parameters to be entered. Note that there are no fields for additives or laboratory analyses, which will need to be added later, when they are known, in the Wines and Vessels tab. This is the key difference between a managed and unmanaged winery; we never know what actually happened until after the action was performed, so it makes sense to only enter known parameters and add extra information later, when they become apparent. It was found that if more detail was available and/or required then perhaps the winery ought to be converted into a managed facility.

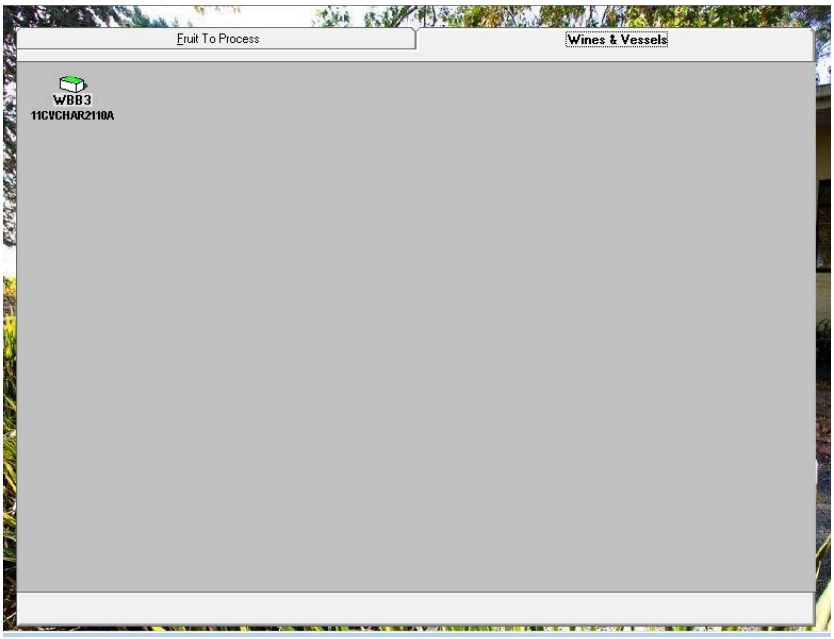


Figure 108. Unmanaged Winery - Wine & Vessels Screen

The Wines and Vessels tab (Figure 108) operates in the same way that the tank map part of the winery map works. All vessels and their wines (at least that are of concern to the enterprise) are displayed here. The menu that appears when a wine is right-clicked is similar to the menu for tank map items; the obvious difference being the details and functions these options perform. (Table 28)

Top Layer

Menu Item	Description	Enabled If...
Comments ...	Displays the "Comments" dialog.	User has winery locked. Only one vessel selected. Vessel contains wine.
Details...	Displays the "Details" dialog.	Single vessel selected
Operations...	Displays the Unmanaged Wineries Operations dialog	User has winery locked. Vessel(s) contains wine
Rename Wine...	Displays the "Rename Wine" dialog.	User has winery locked. Only one vessel selected. Vessel contains wine
Reset Wine Volume	Displays the "Reset Wine Volume" dialog.	User has winery locked. Super Administrator. Single Vessel contains wine.
Wine Settings...	Displays the "Wine Settings" dialog.	
Move Vessel(s) to Other Winery	Displays the "Move Vessels" dialog to allow the user to move vessels to other wineries.	User has winery locked. Vessel(s) are empty.

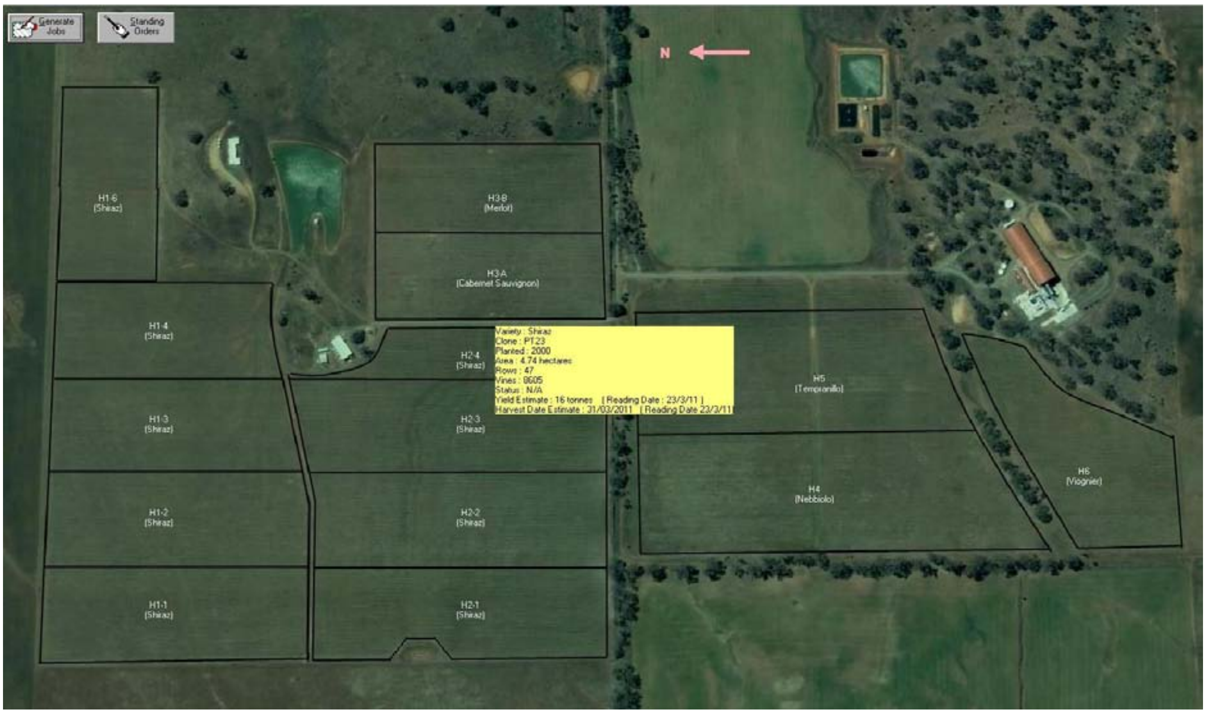
Table 28. Unmanaged Winery Top Layer Menu

The only new feature here is the Unmanaged Wineries Operations dialog which basically replaces all the job/worknote infrastructure in the managed winery with a single dialog allowing operations in the form of processes, additions, or lab work to be entered as completed tasks.

13.5 The Managed Vineyard

The screens for vineyards are a direct reflection of the requirements specified in the "Wish List" chapter *9.1.1 The Vineyard* and the GUI specifications chapter *10.1 Vineyards*

The main screen (Figure 109) which appears when the title node is selected form the main tree menu is a map of the vineyard with the blocks highlighted and labelled. The way the screen operates is similar to the Winery Map screen from the winery. For example, the mouse pointer will generate a tooltip box when it hovers over a block label as shown in Figure 109. Also, individual or multiple blocks can be selected from the screen and functions performed based on these selections and the use of a popup menu.



Variety : Shiraz
Clone : PT23
Planted : 2000
Area : 4.74 hectares
Rows : 47
Vines : 8605
Status : N/A
Yield Estimate : 16 tonnes [Reading Date : 23/3/11]
Harvest Date Estimate : 31/03/2011 [Reading Date 23/3/11

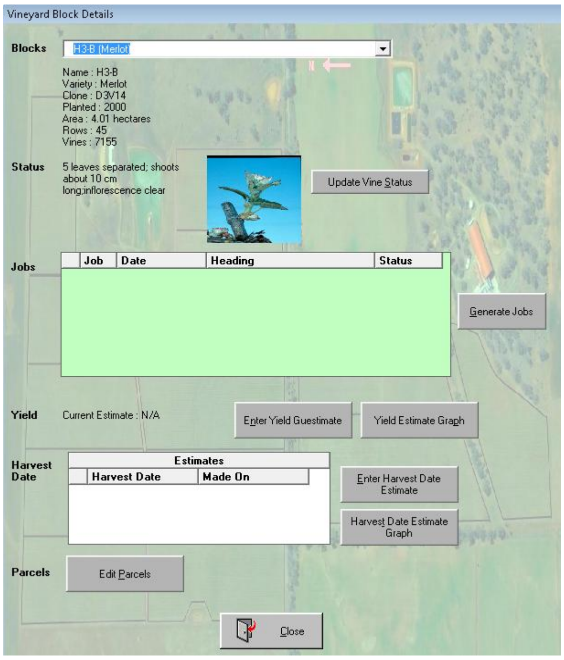
Figure 109. A Managed Vineyard Main Screen

The menu is far simpler because many functions are covered by two main dialogs; the Details dialog which is the main screen for each block, and the Vineyard Jobs dialog that covers all but three of the jobs to be generated in the vineyard. The menu options are...

Details	Displays the Vineyard Block Details dialog. This option is only available if only one vineyard block is selected.
Generate Jobs	This option displays the Jobs sub menu
Spray	Displays the Spray Job Dialog
Fertilising	Displays the Fertilisation Job dialog
Irrigation	Displays the Irrigation Job dialog
Other	Displays the “Vineyard Jobs” dialog

13.5.1 Vineyard Block Details

This dialog (Figure 110) is designed to be a summarising screen, to act as a point of entry for viewing and setting parameters and function on an individual vineyard block, Apart from the usual details concerning area, variety and clone etc, this dialog allows the user to set the Eichorn-Lorenz number for the current status of the vines, edit and graph yield and harvest date estimates, and edit parcels defined for the current vintage.



Vineyard Block Details

Blocks: H3-B (Merlot)

Name : H3-B
Variety : Merlot
Clone : D3V14
Planted : 2000
Area : 4.01 hectares
Rows : 45
Vines : 7155

Status: 5 leaves separated; shoots about 10 cm long;inflorescence clear

Jobs

Job	Date	Heading	Status

Yield: Current Estimate : N/A

Harvest Date

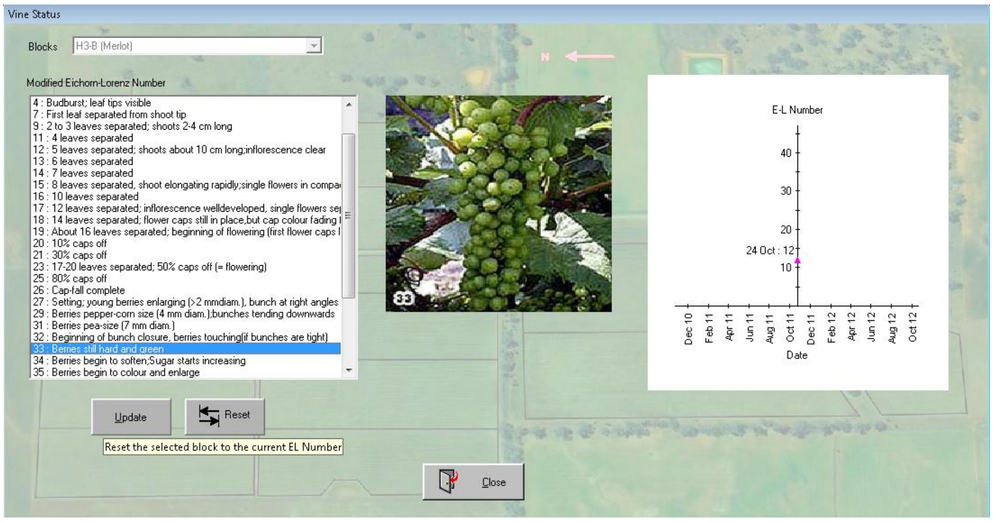
Estimates	
Harvest Date	Made On

Parcels

Figure 110. Vineyard Block Details Dialog

13.5.1.1 Vine Status

The Update Vine Status button invokes this dialog (Figure 111) which allows the user to specify the status of the block by selecting an entry from the list of so-called "Modified Eichorn-Lorenz Numbers". Each number has a corresponding description and a pictorial representation.



Vine Status

Blocks: H3-B (Merlot)

Modified Eichorn-Lorenz Number

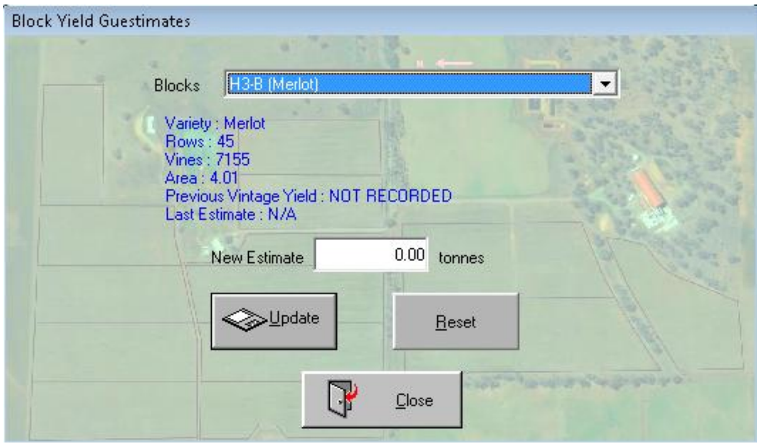
4 : Budburst; leaf tips visible
7 : First leaf separated from shoot tip
9 : 2 to 3 leaves separated; shoots 2-4 cm long
11 : 4 leaves separated
12 : 5 leaves separated; shoots about 10 cm long;inflorescence clear
13 : 6 leaves separated
14 : 7 leaves separated
15 : 8 leaves separated, shoot elongating rapidly;single flowers in compa
16 : 10 leaves separated
17 : 12 leaves separated; inflorescence welldeveloped, single flowers se
18 : 14 leaves separated; flower caps still in place,but cap colour fading
19 : About 16 leaves separated; beginning of flowering (first flower caps l
20 : 10% caps off
21 : 30% caps off
23 : 17-20 leaves separated; 50% caps off (= flowering)
25 : 80% caps off
26 : Cap-fall complete
27 : Setting; young berries enlarging (>2 mmdiam.), bunch at right angles
29 : Berries pepper-corn size (4 mm diam.);bunches tending downwards
31 : Berries pea-size (7 mm diam.)
32 : Beginning of bunch closure, berries touching(if bunches are tight)
33 : Berries still hard and green
34 : Berries begin to soften;Sugar starts increasing
35 : Berries begin to colour and enlarge

Reset the selected block to the current EL Number

Figure 111. Vine Status Dialog

The Update button causes the current selection from the list to be added as a reading to the vineyard block, date-time stamped with the current date and time. The graph to the left represents the change in state over time for the current vintage.

13.5.1.2 Block Yield Estimates



Block Yield Guestimates

Blocks: H3-B (Merlot)

Variety : Merlot
Rows : 45
Vines : 7155
Area : 4.01
Previous Vintage Yield : NOT RECORDED
Last Estimate : N/A

New Estimate 0.00 tonnes

Figure 112. Block Yield Estimate Dialog

Block yield estimates techniques and preferences differ from person to person, and different techniques are used at different stages of the season. Initially, numbers might be based purely on previous vintages, whereas later estimates are based on empirical analyses. This dialog (Figure 112) allows an estimate to be entered without necessarily justifying or specifying the technique used. Yield estimates are also stored as readings attached to the block object. The empirical analyses alluded to before are initiated by formal job/worknote requests and require samples to be collected as well as other statistical observations to be made. When an estimate changes, there are repercussions for the parcels defined for the block in terms of the expected fruit defined for each. A message is generated when a change is made, which alludes to the fact that some parcels might need to be redefined in the parcel definition dialog.

13.5.1.3 Yield Estimate Graph

The graph shown here is generated when the "Yield Estimate Graph" command button is pressed.

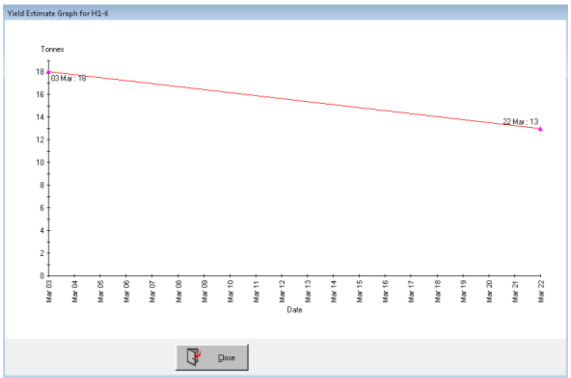
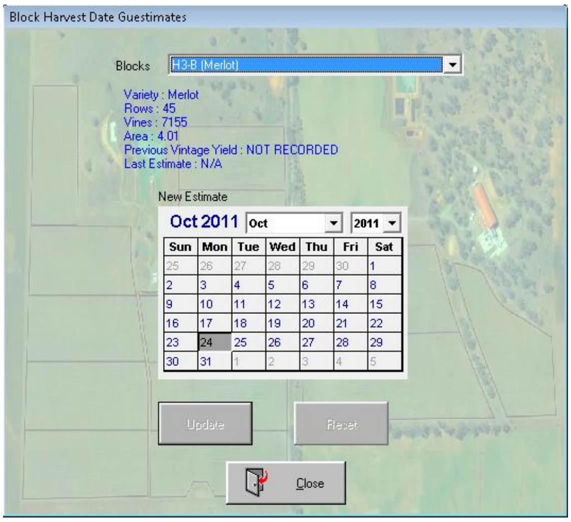


Figure 113. Yield Estimate Graph

It uses the estimate readings stored in the block object.

13.5.1.4 Harvest Date Estimate


Block Harvest Date Guestimates

Blocks: H3-B (Merlot)

Variety : Merlot
Rows : 45
Vines : 7155
Area : 4.01
Previous Vintage Yield : NOT RECORDED
Last Estimate : N/A

New Estimate

Oct 2011 | Oct | 2011

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Figure 114. Harvest Date Estimate Dialog

The harvest date estimate process is also a matter of preferred technique rather than precise science. Particularly susceptible to weather changes late in the growing season, the better the estimate, the better the plans and preparations for the vintage. Harvest Date estimate are also stored as readings for each block. Initial values can be entered using Figure 114 as "Guestimates" based on previous vintages whereas later estimates can be from analyses of graphs of indicative parameters such as pH, T.A., Baume etc. It is important to note that harvest dates can also have subjective and/or pragmatic influences such as wine style requiring early or late harvest, or the availability of harvest staff or equipment. Harvest dates are not the same as ripeness dates.

13.5.1.5 Harvest Date Parameters Graph

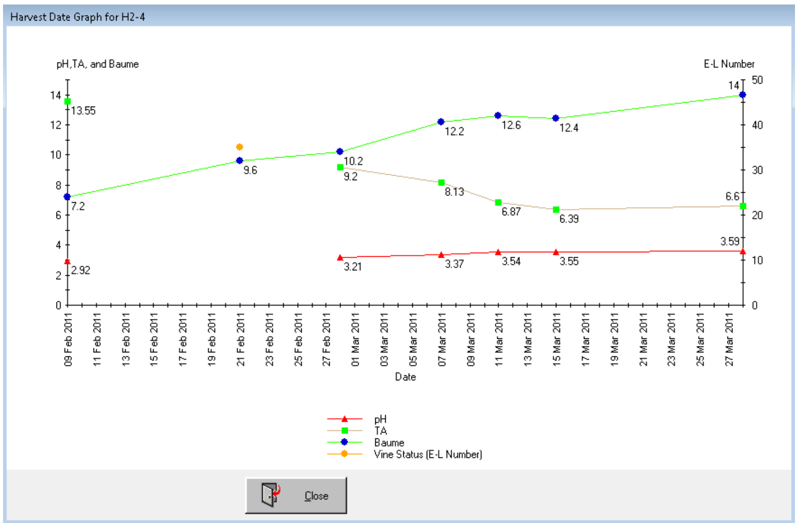
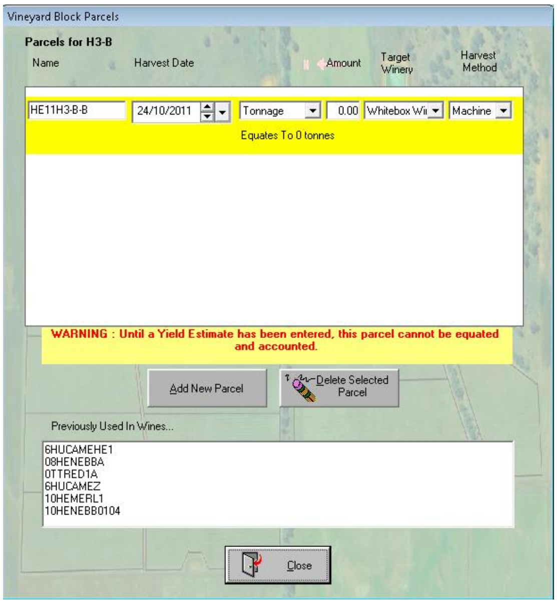


Figure 115. Harvest Date Parameters Graph

The graph shown here of a Harvest Date Parameters graph shows how such parameters might be used to indicate when a block of fruit is likely to be ready for harvest. It is created by pressing the “Harvest Date Estimate Graph” button on the details dialog.

13.5.1.6 Block Parcels

The definition of parcels to be harvested from a block is an issue of logistics. It involves the culmination of harvest date estimates and wine production requirements to create such a unit, and is therefore liable to change and cause changes elsewhere in the vintage plan, vintage logistics, and the overall production plan. As a result, this dialog has several relationships and issues to consider when it is used to create, change or delete a parcel definition.



Vineyard Block Parcels

Parcels for H3-B

Name	Harvest Date		Amount	Target Winery	Harvest Method
HE11H3-B-B	24/10/2011	Tonnage	0.00	Whitebox Wi	Machine

Equates To 0 tonnes

WARNING : Until a Yield Estimate has been entered, this parcel cannot be equated and accounted.

Previously Used In Wines...

6HUCAMEHE1
08HENEBBA
0TTRED1A
6HUCAMEZ
10HEMERL1
10HENEBB0104

Figure 116 . Vineyard Block Parcels Dialog

As can be seen by the illustration (Figure 116), the user is able to define as many parcels as required for the selected block. Parcels can be created, edited and deleted using the command buttons and parcel list. Each parcel can have different harvest dates, masses, target wineries, and harvest methods. Each parcel in the list is a separate instance of user defined control directly linked to a parcel in the model. So when a field is changed, this information is fed into the control and updates the object as well as performs some business rule checks to ensure appropriate messages are generated.

It should be noted that the Harvest Date are used to seed any new parcel definitions added to the list but do not affect nor are they affected by different harvest dates specified for individual parcels.

On the other hand, the yield estimates do need to be reconciled with the parcel expected yields, so a message is generated when the nett parcel yields do not match the overall expected yield.

Another feature to note in this dialog is the parcel amount type which dictates the meaning of the amount specified.

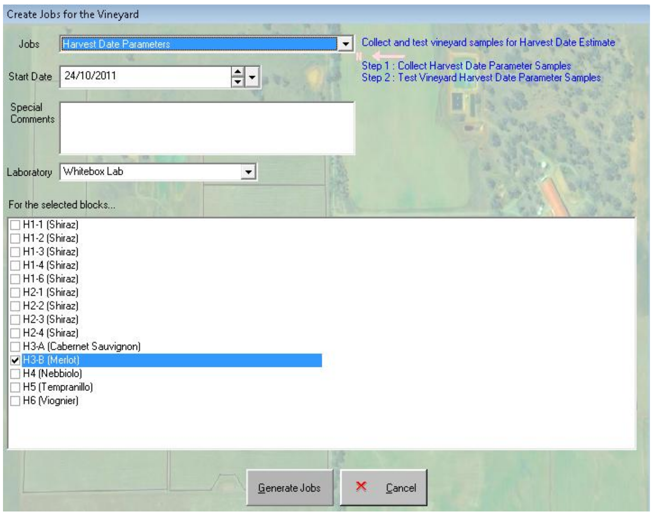
Tonnage The amount specified is in absolute tonnes of fruit

Proportion The amount is a proportion of the nett expected yield

Remainder The amount is ignored and the parcel is defined to have the appropriate amount to satisfy the nett yield.

Obviously only one parcel can be defined as type "Remainder" so a message is generated if more than one is defined. When a parcel is defined, the system resolves the amounts that each parcel equates to in tonnes and displays a message underneath each.

13.5.2 VINEYARD JOBS DIALOG



Create Jobs for the Vineyard

Jobs: Harvest Date Parameters

Collect and test vineyard samples for Harvest Date Estimate

Step 1 : Collect Harvest Date Parameter Samples
Step 2 : Test Vineyard Harvest Date Parameter Samples

Start Date: 24/10/2011

Special Comments:

Laboratory: Whitebox Lab

For the selected blocks...

- □ H1-1 (Shiraz)
- □ H1-2 (Shiraz)
- □ H1-3 (Shiraz)
- □ H1-4 (Shiraz)
- □ H1-6 (Shiraz)
- □ H2-1 (Shiraz)
- □ H2-2 (Shiraz)
- □ H2-3 (Shiraz)
- □ H2-4 (Shiraz)
- □ H3-A (Cabernet Sauvignon)
- ☑ H3-B (Merlot)
- □ H4 (Nebbiolo)
- □ H5 (Tempranillo)
- □ H6 (Viognier)

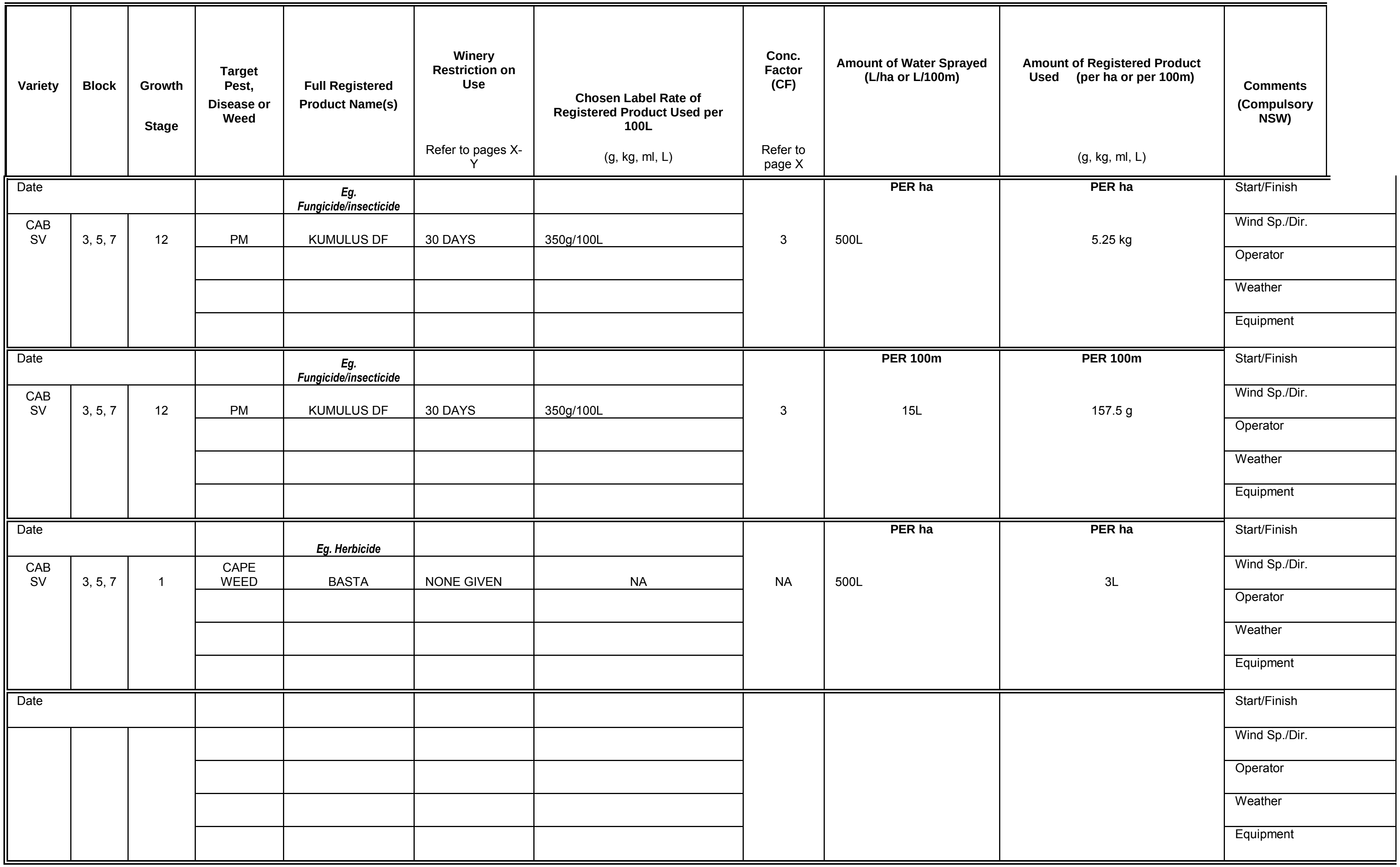
Figure 117. Vineyard Jobs Dialog

This dialog (Figure 117) is used for all of the jobs created in the vineyard that do not require more parameters than the start date, and optionally the block and a nominated laboratory. The actual criteria is that the job type is for a vineyard, and the GenerallyAvailable flag is true. It works in the same way that the Vessel jobs dialog works for the winery. (*13.3.4.5 Generate Job for Vessel*)

13.5.3 IRRIGATION, FERTILISATION, SPRAYING

The remaining options for creating jobs in the winery are now subject to a significant redesign. The original specifications for irrigation zones not matching vineyard blocks was found to be a very frustrating miscommunication because of semantics, but fortunately results in a simplification of the dialogs used to specify the details. Now, the use of irrigation can be performed on a block by block basis and therefore it fits into the same structures used for fertilisation and spraying. This allows a generic dialog to be created that will cover all three processes. The details for these worknotes are easily catered for within the worknote structure because of the broad and generic parameter mapping system.

The only other issue that needed resolution was the generation of a spray diary. This is simply a report that can be generated from the data from spraying jobs. Using a recommended template for a spray diary, supplied by the AWRI website, we can see the fields that will be necessary for the Spraying worknote. (Figure 118. Spray Diary Example - from AWBC website)

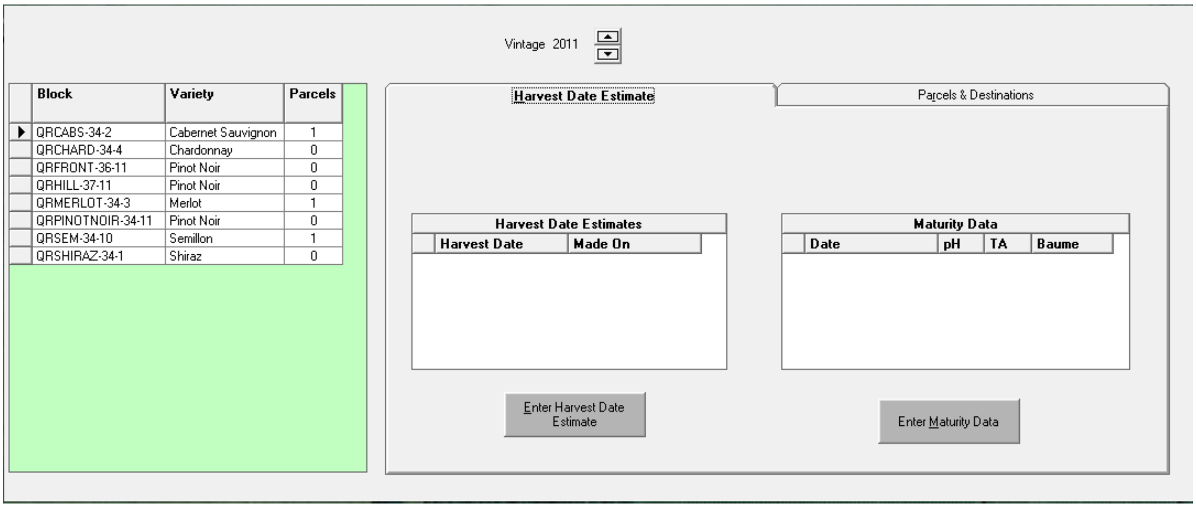


Variety	Block	Growth Stage	Target Pest, Disease or Weed	Full Registered Product Name(s)	Winery Restriction on Use Refer to pages X-Y	Chosen Label Rate of Registered Product Used per 100L (g, kg, ml, L)	Conc. Factor (CF) Refer to page X	Amount of Water Sprayed (L/ha or L/100m)	Amount of Registered Product Used (per ha or per 100m) (g, kg, ml, L)	Comments (Compulsory NSW)
Date				***Eg. Fungicide/insecticide***				**PER ha**	**PER ha**	Start/Finish
CAB SV	3, 5, 7	12	PM	KUMULUS DF	30 DAYS	350g/100L	3	500L	5.25 kg	Wind Sp./Dir.
										Operator
										Weather
										Equipment
Date				***Eg. Fungicide/insecticide***				**PER 100m**	**PER 100m**	Start/Finish
CAB SV	3, 5, 7	12	PM	KUMULUS DF	30 DAYS	350g/100L	3	15L	157.5 g	Wind Sp./Dir.
										Operator
										Weather
										Equipment
Date				***Eg. Herbicide***				**PER ha**	**PER ha**	Start/Finish
CAB SV	3, 5, 7	1	CAPE WEED	BASTA	NONE GIVEN	NA	NA	500L	3L	Wind Sp./Dir.
										Operator
										Weather
										Equipment
Date										Start/Finish
										Wind Sp./Dir.
										Operator
										Weather
										Equipment

Figure 118. Spray Diary Example - from AWBC website

13.6 The Unmanaged Vineyard

The Unmanaged Vineyard, like the Unmanaged Winery, needs only to allow for certain critical data points about the inputs and outputs of the facility so that the flow of relevant information and data integrity of the data model is maintained. In this case, we are only really concerned with parcel definitions. There is also a need for some shared information concerning harvest date parameters because the decision of when fruit is to be harvested may actually lay with the winemaker or customer rather than the manager of the vineyard. Note that the block yield information is not required because the nett yield from the block is of no concern to the enterprise.



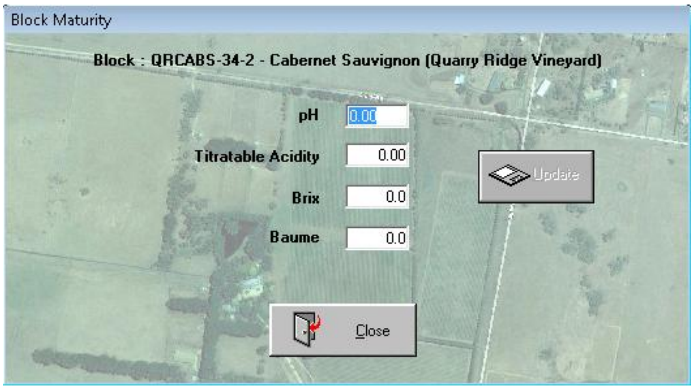
Block	Variety	Parcels
QRCABS-34-2	Cabernet Sauvignon	1
QRCHARD-34-4	Chardonnay	0
QRFRONT-36-11	Pinot Noir	0
QRHILL-37-11	Pinot Noir	0
QRMERLOT-34-3	Merlot	1
QRPINOTNOIR-34-11	Pinot Noir	0
QRSEM-34-10	Semillon	1
QRSHIRAZ-34-1	Shiraz	0

Figure 119. Unmanaged Vineyard Screen

The main screen for the unmanaged vineyard (Figure 119) is a summarised version of a managed vineyard.

There is no breakdown of the facility into blocks, but rather, they are listed to the left of the main screen. The user selects the appropriate block and then the two tabs are populated with data for that block.

The first tab allows Harvest date estimates in the same way that it is input for the managed winery, and in fact uses the same dialog. It also allows maturity data to be entered without the need for a laboratory facility, because the data is usually supplied as straight data from the grower, without concern for how they did the measurements. The entry screen (Figure 120) appears when the "Enter Maturity Data" button is pressed, and like the Harvest data, it's stored as a reading for the block.



Block Maturity

Block : QRCABS-34-2 - Cabernet Sauvignon (Quarry Ridge Vineyard)

pH 0.00

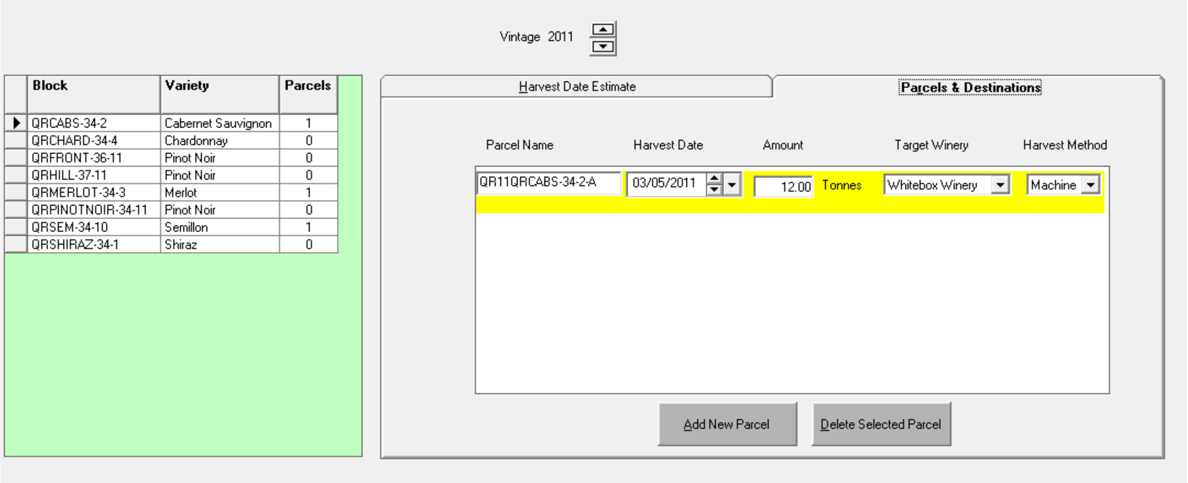
Titratable Acidity 0.00

Brix 0.0

Baume 0.0

Figure 120. Unmanaged Vineyard Block Maturity Input dialog

The Parcel definition tab (Figure 121) is a clone of the Parcels dialog for a managed vineyard (*13.5.1.6 Block Parcels*), however, because there is no nett yield data to compare to, no messages are generated when the amount is changed for any parcel. Similarly, the amount can only be expressed in tonnes, not proportions or remainder. This is in keeping with the typical nature of the relationship between fruit producers and winemaking enterprises where a tonnage is ordered, not a proportion of the overall expected yield.



Vintage 2011

Block	Variety	Parcels
QRCABS-34-2	Cabernet Sauvignon	1
QRCHARD-34-4	Chardonnay	0
QRFRONT-36-11	Pinot Noir	0
QRHILL-37-11	Pinot Noir	0
QRMERLOT-34-3	Merlot	1
QRPINOTNOIR-34-11	Pinot Noir	0
QRSEM-34-10	Semillon	1
QRSHIRAZ-34-1	Shiraz	0

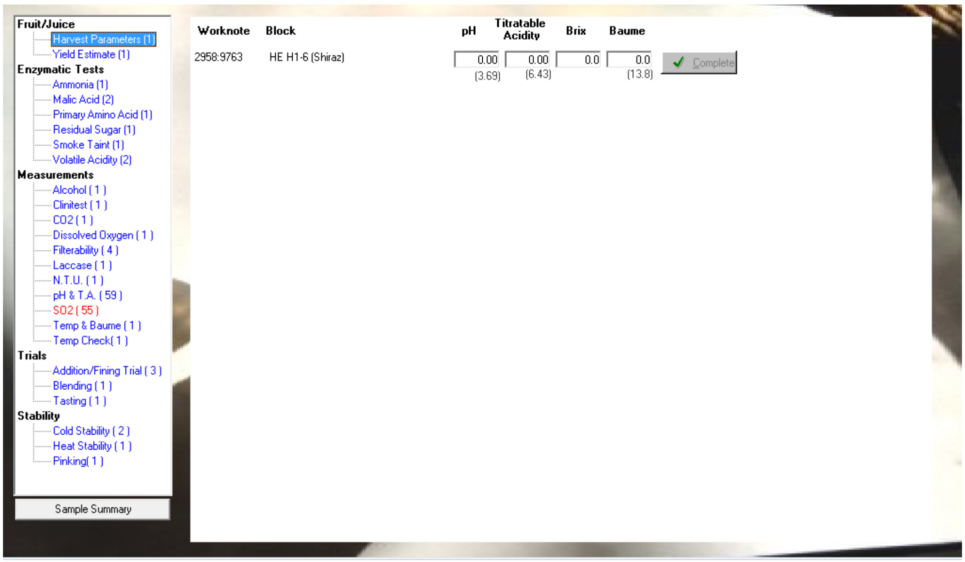
Parcel Name	Harvest Date	Amount		Target Winery	Harvest Method
QR11QRCABS-34-2-A	03/05/2011	12.00	Tonnes	Whitebox Winery	Machine

Figure 121. Unmanaged Vineyard - Parcels & Destinations Tab

13.7 Laboratories

The Laboratory Facility only requires a single node in the main tree menu because the function only has a single screen. This screen, however, is quite complex because it basically behaves as a magic lab notebook that allows the user to enter the measurements and it does the calculations necessary to derive the final outcome. In the case of an unmanaged laboratory, the only option available for each test type will be to enter the result directly; an option also available in the managed vineyard but not by default.

In keeping with a consistent interactive theme, the main laboratory screen (Figure 122) consists of a menu tree control allowing the user to select the analysis type which causes a list of the required tests to appear. Each type of test has a specific data entry control that includes a “Submit” or “Complete” button that becomes enabled when valid data is entered. The title of the button depends on whether the test type has been flagged to require ratification before any of the readings that might be generated are accepted as legitimate. This was discussed in *13.3.5 Reading Ratification*. If the flag is set and the reading does require ratification then any readings generated are simply flagged as requiring ratification and the other screens pick them up that way.



Worknote	Block	pH	Titratable Acidity	Brix	Baume
2958:9763	HE H1-6 (Shiraz)	0.00 (3.69)	0.00 (6.43)	0.0	0.0 (13.8)

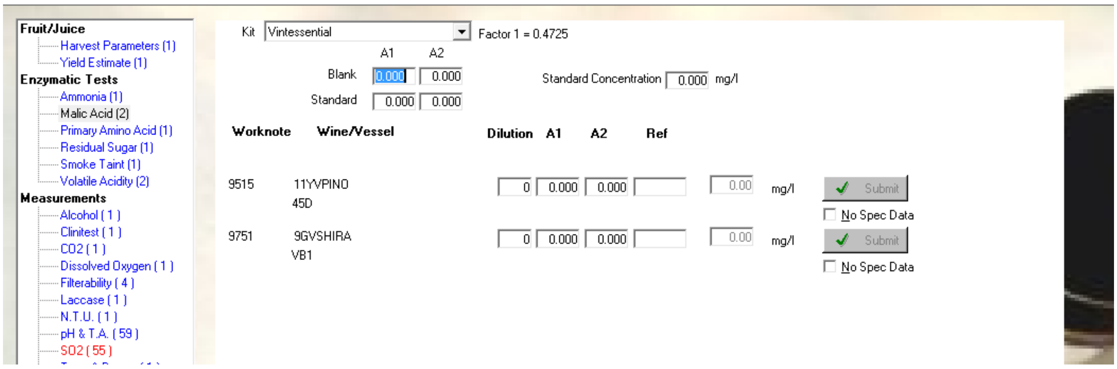
Figure 122. Laboratory Screen

The nodes are divided into categories as specified in the wish list chapter (see *9.1.3 Laboratories*)

Nodes that have corresponding analysis requests appear in blue with the number of tests in brackets, whereas nodes without any tests are in black. If one of the tests has a high priority then the label is in red, and the test will appear with a red background.

Obviously each test type has a different data entry control, but they generally come in four main types which are described below. Any and all changes entered into any of the controls are automatically recorded and cause the appropriate recalculations on the screen. This is particularly important given the fact that if the user selects another node, then returns to one of these screens, the information previously entered is not lost.

13.7.1 Enzymatic Tests



Kit Vintessential Factor 1 = 0.4725

	A1	A2
Blank	0.000	0.000
Standard	0.000	0.000

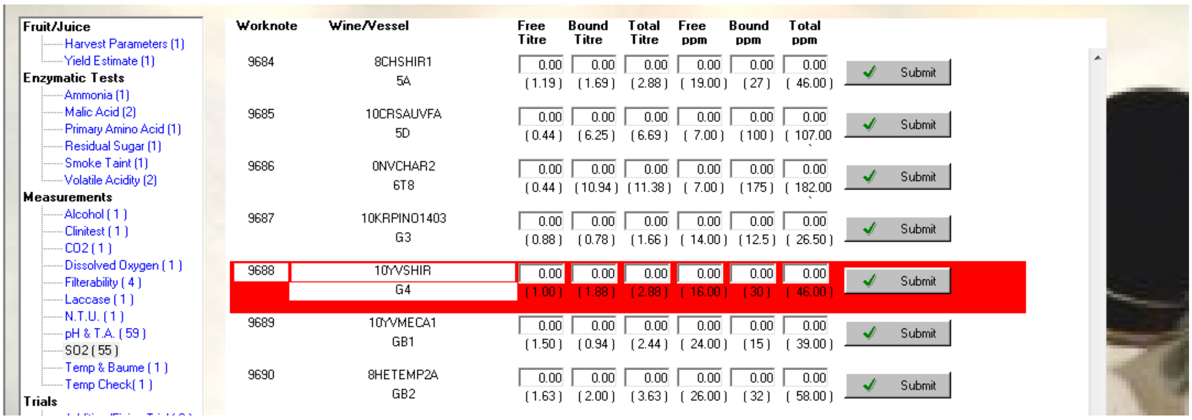
Standard Concentration 0.000 mg/l

Worknote	Wine/Vessel	Dilution	A1	A2	Ref				
9515	11YVPINO 45D	0	0.000	0.000		0.00	mg/l	✓ Submit	□ No Spec Data
9751	9GVSHIRA VB1	0	0.000	0.000		0.00	mg/l	✓ Submit	□ No Spec Data

Figure 123. Enzymatic Analysis Screen example

In the example shown (Figure 123) we see two tests requested. Enzymatic tests like these are done in batches because they require a blank and a standard to be analysed as well as the sample, in order to get a reading. This is why the header for the screen includes fields for the Blank and Standard tests. The sequence of additions of various chemicals in the analysis may lead to two or three spectrophotometric absorbance readings being taken. Once all the parameters have been entered, the results are displayed, along with an indication of the percentage recovery of the standard, which gives an indication of the accuracy of the test. The "No Spec Data" option allows the user to enter the result directly, in the event that the result is known but the actual analysis parameters are not. This can be useful for entering historical data. There are various kits available from different manufacturers that all perform the same chemical function, but may use different amounts or concentrations, and therefore have different calculations. These factors are associated with the kits and retrieved for calculations when the kit is selected from the drop-down list at the top of the screen.

13.7.2 Calculated Tests



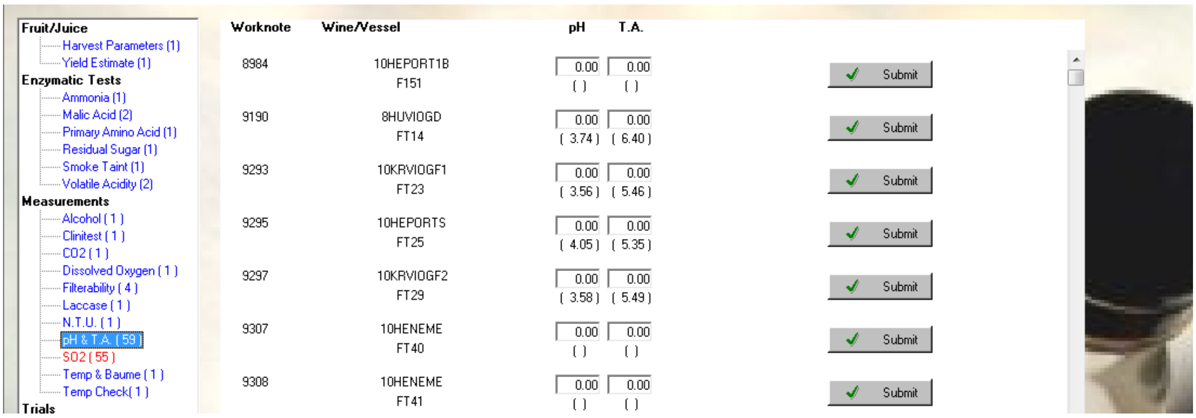
Worknote	Wine/Vessel	Free Titre	Bound Titre	Total Titre	Free ppm	Bound ppm	Total ppm	
9684	8CHSHIR1 5A	0.00 (1.19)	0.00 (1.69)	0.00 (2.88)	0.00 (19.00)	0.00 (27)	0.00 (46.00)	✓ Submit
9685	10CRSAUVFA 5D	0.00 (0.44)	0.00 (6.25)	0.00 (6.69)	0.00 (7.00)	0.00 (100)	0.00 (107.00	✓ Submit
9686	0NVCHAR2 6T8	0.00 (0.44)	0.00 (10.94)	0.00 (11.38)	0.00 (7.00)	0.00 (175)	0.00 (182.00	✓ Submit
9687	10KRPINO1403 G3	0.00 (0.88)	0.00 (0.78)	0.00 (1.66)	0.00 (14.00)	0.00 (12.5)	0.00 (26.50)	✓ Submit
9688	10YVSHIR G4	0.00 (1.00)	0.00 (1.88)	0.00 (2.88)	0.00 (16.00)	0.00 (30)	0.00 (46.00)	✓ Submit
9689	10YVMECA1 GB1	0.00 (1.50)	0.00 (0.94)	0.00 (2.44)	0.00 (24.00)	0.00 (15)	0.00 (39.00)	✓ Submit
9690	8HETEMP2A GB2	0.00 (1.63)	0.00 (2.00)	0.00 (3.63)	0.00 (26.00)	0.00 (32)	0.00 (58.00)	✓ Submit

Figure 124. Calculated Lab Test example

In this example (Figure 124) we see how the control allows the entry of a measurement, in this case a titration volume, which has a formula applied to it to generate the final test result. The results themselves can be entered directly. In this example, entering the Free SO_2 reading causes the Bound and Total readings to be revised, and the titres to be reverse calculated. Note, too that one of the tests has a red background indicating that the test has a high priority.

13.7.3 Direct Entry Tests

These tests are the simplest in that they just require the final reading to be entered without any calculations. In the example shown, pH and T.A. readings can be entered. This example (Figure 125) also shows previous results displayed directly under each field, if they are available. This can act as a secondary check and highlight any strange or rapid shifts that might be brought to the notice of the winemaker.

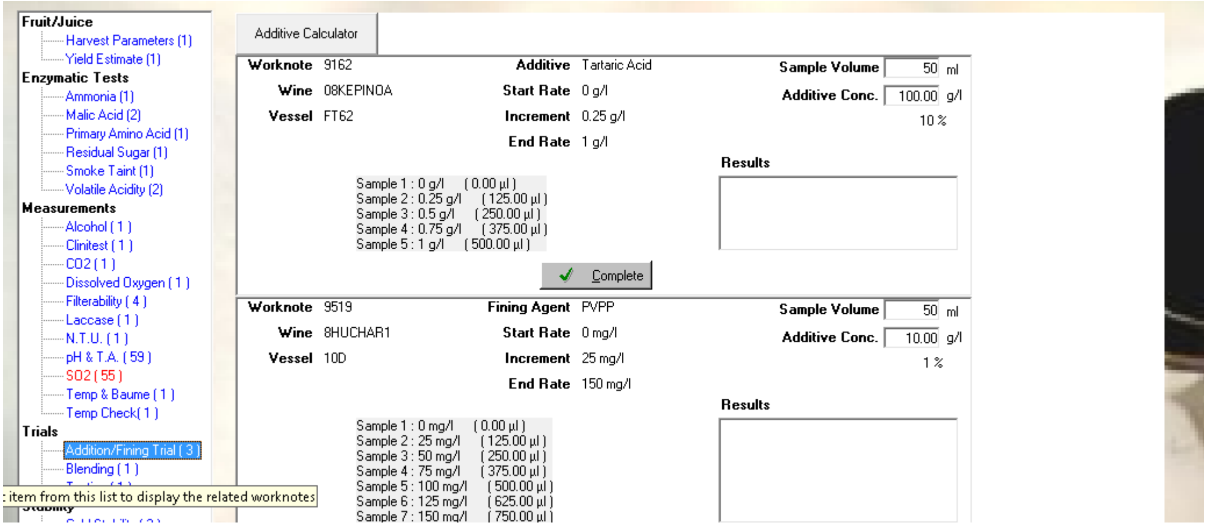


Worknote	Wine/Vessel	pH	T.A.	
8984	10HEPORT1B F151	0.00 ( )	0.00 ( )	✓ Submit
9190	8HUVIOGD FT14	0.00 (3.74)	0.00 (6.40)	✓ Submit
9293	10KRVIOGF1 FT23	0.00 (3.56)	0.00 (5.46)	✓ Submit
9295	10HEPORTS FT25	0.00 (4.05)	0.00 (5.35)	✓ Submit
9297	10KRVIOGF2 FT29	0.00 (3.58)	0.00 (5.49)	✓ Submit
9307	10HENEME FT40	0.00 ( )	0.00 ( )	✓ Submit
9308	10HENEME FT41	0.00 ( )	0.00 ( )	✓ Submit

Figure 125. Direct Entry Lab Test examples

13.7.4 Trials & the Additive Calculator

Trials such as blending, tasting or addition trials do not necessarily generate an objective or empirical reading. However, as discussed earlier in this document, such results in the form of comments or rationalisation of further actions are perfectly valid forms of data from a winemaking perspective. In fact, the more descriptive the prose, the easier the causal link is to establish.



Worknote 9162 Additive Tartaric Acid Sample Volume 50 ml
Wine 08KEPINOA Start Rate 0 g/l Additive Conc. 100.00 g/l
Vessel FT62 Increment 0.25 g/l 10 %
End Rate 1 g/l

Results

Sample 1 : 0 g/l (0.00 µl)
Sample 2 : 0.25 g/l (125.00 µl)
Sample 3 : 0.5 g/l (250.00 µl)
Sample 4 : 0.75 g/l (375.00 µl)
Sample 5 : 1 g/l (500.00 µl)

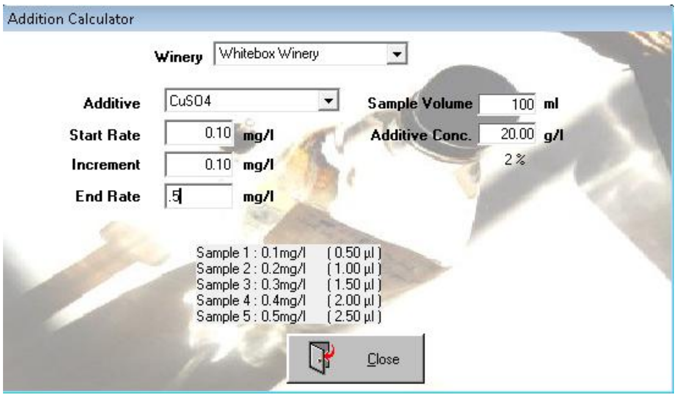
Worknote 9519 Fining Agent PVPP Sample Volume 50 ml
Wine 8HUCHAR1 Start Rate 0 mg/l Additive Conc. 10.00 g/l
Vessel 10D Increment 25 mg/l 1 %
End Rate 150 mg/l

Results

Sample 1 : 0 mg/l (0.00 µl)
Sample 2 : 25 mg/l (125.00 µl)
Sample 3 : 50 mg/l (250.00 µl)
Sample 4 : 75 mg/l (375.00 µl)
Sample 5 : 100 mg/l (500.00 µl)
Sample 6 : 125 mg/l (625.00 µl)
Sample 7 : 150 mg/l (750.00 µl)

Figure 126. Trials Test example

The Addition Fining trial shown as an example (Figure 126), has been selected because it also highlights some useful functions that have been included in the control to aid the testing process. In this case, the user can enter the proposed sample size and concentration of the additive and it will calculate the additions to each sample in the series of the trial. The same function is available as an "Additive Calculator" dialog (Figure 127) that can be invoked from the command button displayed at the top of the screen.



Winery Whitebox Winery

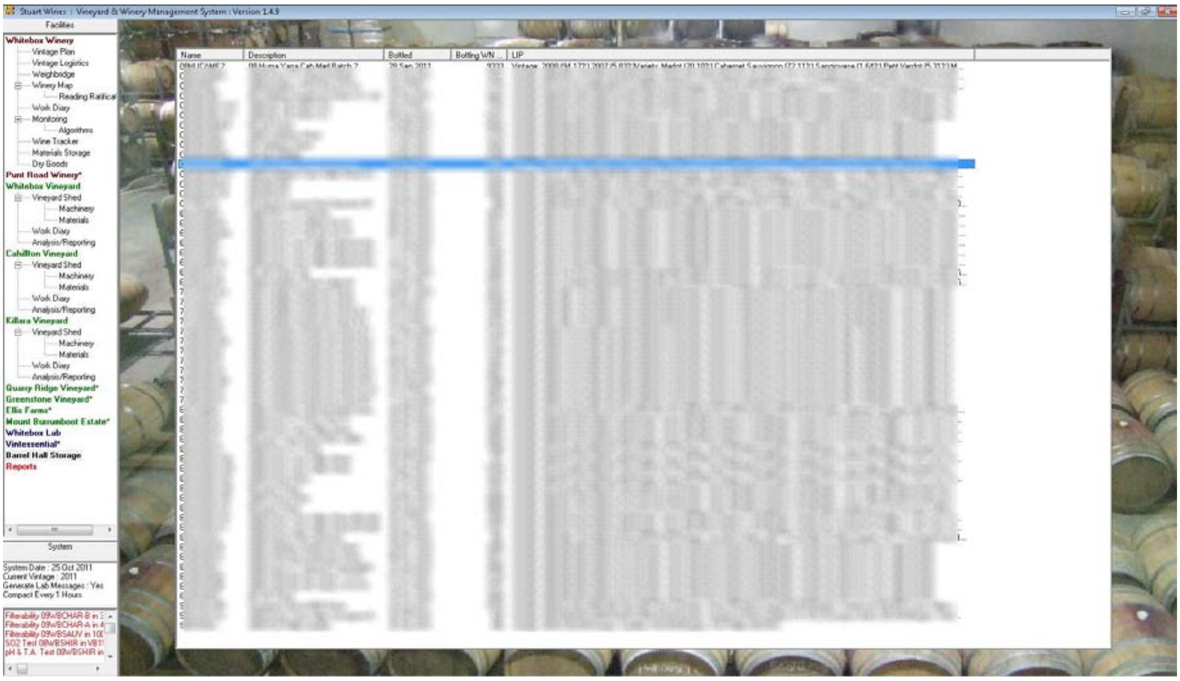
Additive CuSO4 Sample Volume 100 ml
Start Rate 0.10 mg/l Additive Conc. 20.00 g/l
Increment 0.10 mg/l 2 %
End Rate .5 mg/l

Sample 1 : 0.1mg/l (0.50 µl)
Sample 2 : 0.2mg/l (1.00 µl)
Sample 3 : 0.3mg/l (1.50 µl)
Sample 4 : 0.4mg/l (2.00 µl)
Sample 5 : 0.5mg/l (2.50 µl)

Figure 127. Addition Calculator Dialog

13.8 Storage Facilities

Originally, the design for Storage Facilities window (Figure 128) was quite ambitious and complex; the idea was to provide a complete logistical control screen that would locate and track batches of wines around and between storage facilities. However, this was found to be a major distraction from the essential aspects of the project because it was not really a wine production issue within the scope of the project. If the system was to be used commercially then there is no doubt this kind of functionality would be important, but at this point, the function of the screen is relatively simple.



System Date : 25 Oct 2011
Current Vintage : 2011
Generate Lab Messages : Yes
Compact Every 1 Hours

Filterability 09WBCHAR-B in
Filterability 09WBCHAR-A in 4
Filterability 09WBSAUV in 10
SO2 Test 08WBSHIR in VB1
pH & T.A. Test 09WBSHIR in

Figure 128. Storage Screen

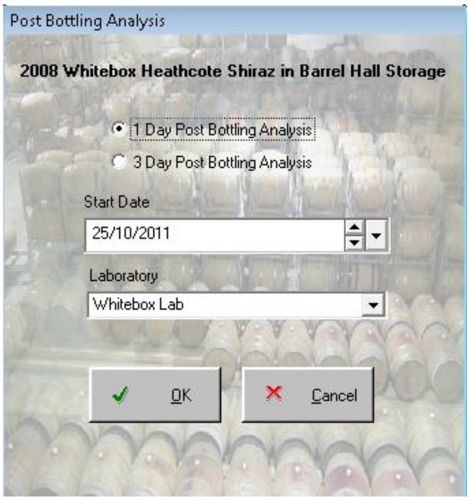
It simply consists of a list of bottled wines stored in the facility. (Please note the data has been blurred due to its commercial sensitivity). The list includes various details that enable the user to identify specific aspects of the wine and the bottling process; double-clicking on an item in the list loads the bottling job report into the Job viewer dialog. Right-clicking on an item displays a popup menu with three options...

1 day post bottling analysis...	Invokes the Post Bottling Analysis creation dialog with the 1 day test preselected.
3 day post bottling analysis...	Invokes the Post Bottling Analysis creation dialog with the 3 day test preselected.
Move Wine...	Display the “Move Wine” dialog

Table 29. Bottled Wine Menu

13.8.1 Post Bottling Analysis

This dialog (Figure 129) is used for both the One and Three day analyses jobs. These jobs are similar to the pre bottling jobs (see *13.3.4.24.2 Pre Coarse Filtering / Bottling*) except that they focus on parameters that are of concern after the process. (Table 30)



Post Bottling Analysis

2008 Whitebox Heathcote Shiraz in Barrel Hall Storage

● 1 Day Post Bottling Analysis

○ 3 Day Post Bottling Analysis

Start Date: 25/10/2011

Laboratory: Whitebox Lab

Figure 129. Post Bottling Analysis job creation dialog

The aim is to ...

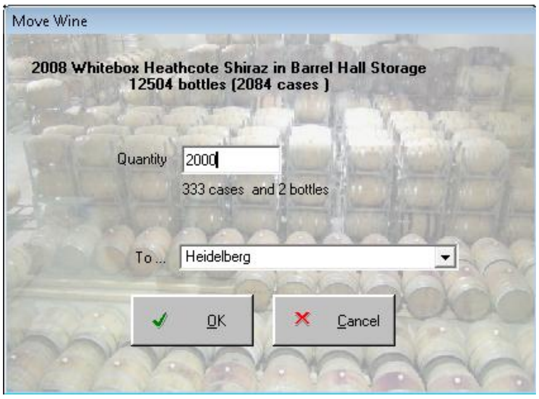
1. See whether or not any elevated levels of parameters are likely to be long term concerns by monitoring any drop-off over the short term.
2. Ensure that any additions immediately prior to the bottling process have taken effect, such as grape concentrate, SO_2 etc.

	1 Day Post	3 Day Post
Free and Total SO_2	✓	✓
pH & TA	✓	✓
Residual Sugar	✓	
Volatile Acidity	✓	
Malic Acid	✓	
N.T.U (Turbidity)	✓	✓
Dissolved Oxygen	✓	✓
CO_2	✓	

Table 30. Post Bottling Jobs and Worknotes

The job and worknotes are applied to the wine objects, not the bottled wine objects, which causes the jobs to appear in the work diary of the winery. This alleviates the need to have a work diary function in the storage facility.

13.8.2 Move Wine


Move Wine

2008 Whitebox Heathcote Shiraz in Barrel Hall Storage
12504 bottles (2084 cases)

Quantity 2000

333 cases and 2 bottles

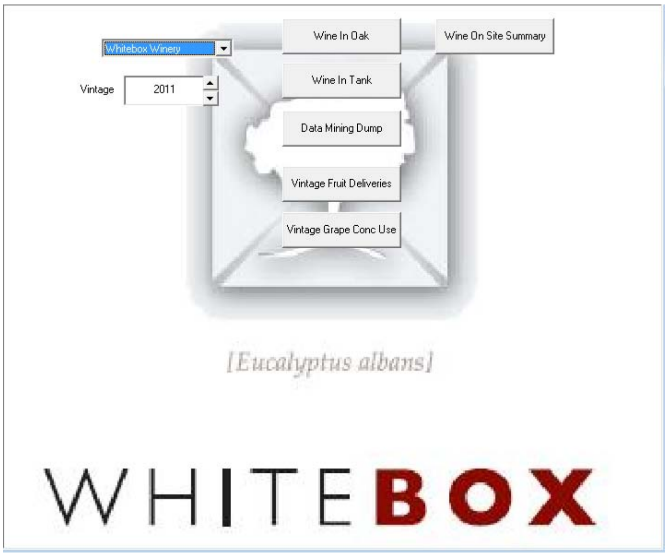
To ... Heidelberg

Figure 130. The Move Wine dialog

This dialog (Figure 130) does not generate a job. There is no worknote type to move a wine from one facility to another. The dialog simply causes the bottled wine object to have a new location assigned to it by setting the StorageID property of the object to the id of the selected facility.

13.9 Reports

The Reports screen is really a generic and central location for reports to be initiated. There are no permission limitations restricting the number of users accessing this screen at the same time, and so it is permanently in Read-Only state. At the moment, there are only a select few reports available as they tend to come on line as required. The facility selected from the drop-down list acts as a filter for the list of available reports that appear in the form of command buttons. The vintage spin box is used as a parameter for any reports that require a vintage to be nominated.


Whitebox Winery

Vintage 2011

Figure 131. Reports Screen

This screen (Figure 131) will obviously develop and become more sophisticated as the need for more reports come through. Of those already defined...

Wine In Oak	Provides a list of all the current wines that have some or all of their contents in oak, and the volume of that wine in oak.
Wine In Tank	Provides a list of all the current wines that have some or all of their contents in vessels other than oak, and the volume of that wine.
Wine In Site Summary	This is a general report of all the current wines in the selected winery. It includes all the LIP information of reach wine.
Data Mining Dump	This is a detailed dump of all the wines into a spreadsheet that can be manipulated using the facilities of Excel. It is in lieu of a proper data mining function that requires more time to be developed than can be allowed for this project.
Vintage Fruit Deliveries	A summary by harvest date by block of all the fruit parcels delivered to the selected winery.
Vintage Grape Conc. Use	Displays a list of all the Grape Concentrate addition jobs over the selected vintage.

Table 31. Reports Screen Options

14. A POPULATED INSTANCE OF THE MODEL

This chapter is concerned with describing a fully populated database and the nature of the data it contains. In this instance (Table 32), it is the database used to pilot and test the application at Stuart Wines main winery facility near Heathcote in Central Victoria. It covers two complete vintages of activities and details of wines going back four years. The enterprise, as described in Case study 1 of Chapter 7, is at the very top end of scale of enterprise that this system was designed for.

MDB size	8.3 Mb
Nº Jobs	2671
Nº worknotes	8452
Nº wines	1655 (194 current)
Nº Wine Readings	27,436

Table 32. Database parameters

Audit Log information dumping into the database was turned off because it was causing the size of the database to expand by about 1Mb per week.

14.1 JOBS AND WORKNOTES

There are 73 different Job types so far designed for the system and 76 different worknote types.. Naturally, some worknote types are used in several Job types. Details for these jobs are not important and are user defined to a certain extent; however the worknotes do have some important internal features that need to be described. In particular the way the parameters and results fields are used in the worknote objects.

Worknote	Parameters	Results
Collect Yield Estimate Samples	None	None
Test Vineyard Yield Estimate Samples	None	R1 : Average Bunch Weight R2 : Bunches Per Vine R3 : Yield Estimate
Collect Harvest Date Parameter Samples	None	None
Test Vineyard Harvest Date Parameter Samples	None	R1 : pH R2 : TA R3 : Baume
Barrel Prune	None	None
Drop Fruit	None	None
Post Vintage Prune	None	None

Worknote	Parameters	Results
Petiole Test	None	None
Soil Test	None	None
Take Climate Readings from Vineyard		R1: Rainfall R2 : Maximum Temperature R3 : Minimum Temperature
Ammonia Test	P1 : Kit ID P2 : Blank A1 P3 : Blank A2 P4 : Standard A1 P5 : Standard A2 P6 : Standard Concentration P7 : Dilution P8 : A1 P9 : A2 P10 : Reference	R1 : Ammonia concentration result
Malic Acid Test	P1 : Kit ID P2 : Blank A1 P3 : Blank A2 P4 : Standard A1 P5 : Standard A2 P6 : Standard Concentration P7 : Dilution P8 : A1 P9 : A2 P10 : Reference	R1 : Malic Acid concentration result
Primary Amino Acid Test	P1 : Kit ID P2 : Blank A1 P3 : Blank A2 P4 : Standard A1 P5 : Standard A2 P6 : Standard Concentration P7 : Dilution P8 : A1 P9 : A2 P10 : Reference	R1 : Primary Amino Acid concentration result
Residual Sugar Test	P1 : Kit ID P2 : Blank A1 P3 : Blank A2 P4 : Blank A3 P5 : Standard A1 P6 : Standard A2 P7 : Standard A3 P8 : Standard Concentration P9 : Dilution P10 : A1 P11 : A2 P12 : A3 P13 : Reference	R1 : Fructose concentration result R2 : Glucose concentration result R3 : Residual Sugar concentration result

14. A populated instance of the model

Worknote	Parameters	Results
Volatile Acidity Test	P1 : Kit ID P2 : Blank A1 P3 : Blank A2 P4 : Blank A3 P5 : Standard A1 P6 : Standard A2 P7 : Standard A3 P8 : Standard Concentration P9 : Dilution P10 : A1 P11 : A2 P12 : A3 P13 : Reference	R1 : Volatile Acidity concentration result
Alcohol Test		R1 : Alcohol percentage result
Clinitest Test		R1 : Clinitest result
CO_2 Test		R1 : CO_2 concentration result
Dissolved Oxygen Test		R1 : D.O. result
N.T.U. Test		R1 : N.T.U result
pH & T.A. Test		R1 : pH R2 : T.A.
SO_2 Test		R1 : Free SO2 ppm R2 : Total SO2 ppm
Temp & Baume Test		R1 : Temperature R2 : Corrected Baumé
Addition/Fining Trial	P1 : Additive ID P2 : Start Rate P3 : Increment P4 : End Rate P5 : Sample Volume P6 : Additive Concentration	R1 : Results field
Blending Trial		
Tasting Trial		R1 : Results field
Process Fruit	All parameters and results stored in Job Fruit Receival objects	
Add SO2	P1 : Additive ID P2 : Amount	R1 : Final Additive ID R2 : Final Amount R3 : Batch ID
Add Acid	P1 : Additive ID P2 : Amount	R1 : Final Additive ID R2 : Final Amount R3 : Batch ID
Add Yeast	P1 : Yeast ID P2 : Amount P3 : Nutrient ID P4 : Nutrient Amount	R1 : Final Yeast ID R2 : Final Amount R3 : Batch ID R4 : Final Nutrient ID R5 : Final Nutrient Amount

Worknote	Parameters	Results
		R6 : Nutrient Batch ID
Transfer/Rack Wine	P1 : Source Vessel ID P2 : Volume to transfer P3 : Transfer Method P4 : Target Vessel P5 : Filter ID P6 : Leave Diffusers In P7 : Clean Empty Vessels P8 : Transfer Chips P9 : Transfer MOX	R1 : Source Vessel ID R2 : Volume transferred R3 : Transfer Method Used R4 : Target Vessel R5 : Filter ID used R6 : Volume remaining R7 : Remainder was lees
Press Wine	P1 : Press ID P2 : Regime ID P3 : Target Vessels string	R1 : Target Wine 1 R2 : Target Wine 2 R3 : Target Wine 3 R4 : Wine 1 Amount R5 : Wine 2 Amount R6 : Wine 3 Amount
Clean Tank		
Sanitise Tank		
Rinse Tank		
Clean Equipment		
Sanitise Equipment		
Rinse Equipment		
Add Chips	P1 : Chips Type ID P2 : Amount in grams	R1 : Chips Type ID R2 : Amount in grams R3 : Chips Batch ID
MOX Setting	P1 : Add, change or remove P2 : Rate	
Diffuser Setting	P1 : In? Yes or No	
Remove Chips		
Heat/Cool Wine	P1 : Heat (0) or Cool (1) P2 : Method ID P3 : Target Temperature	R1 : Heat (0) or Cool (1) R2 : Method ID R3 : Temperature Achieved
Bottle Wine	P1 : Bottling Plant ID P2 : Storage Facility ID P3 : Description P4 : Labels ID P5 : Bottle ID P6 : Box ID P7 : Divider ID P8 : Pad ID P9 : Closure ID P10 : Capsule ID P11 : Pallet ID P12 : Sterile Filter P13 : Volume P14 : Vessel ID	R1 : Bottling Plant ID R2 : Storage Facility ID R3 : Final Bottle Count R4 : Labels Batch ID R5 : Bottle Batch ID R6 : Box Batch ID R7 : Divider Batch ID R8 : Pad Batch ID R9 : Closure Batch ID R10 : Capsule Batch ID R11 : Pallet Batch ID
Transfer Skins from Press	P1 : Skin ID	R1 : Skin ID

Worknote	Parameters	Results
	P2 : Vessel ID	R2 : Vessel ID
Transfer Skins from Vessel	P1 : Skin ID P2 : Vessel ID P3 : All (0) or Some(1)	R1 : Skin ID R2 : Vessel ID R3 : All (0) or Some(1)
Scum Run	P1 : Vessel ID P2 : Ullage P3 : Ullage Check P4: Gas Blanket P5 : Gas Blanket Check P6 : Cooling P7 : Temperature P8 : Scum	R1 : Vessel ID R2 : Ullage R3 : Ullage Check R4: Gas Blanket R5 : Gas Blanket Check R6 : Cooling R7 : Temperature
Add Additive	P1 : Additive ID P2 : Amount	R1 : Additive ID R2 : Amount R3 : Batch ID
Add Fining	P1 : Additive ID P2 : Amount	R1 : Additive ID R2 : Amount R3 : Batch ID
Add Malo Bugs	P1 : Additive ID P2 : Amount	R1 : Additive ID R2 : Amount R3 : Batch ID
Temperature Check	P1 : Time	R1 : Temperature
Battonage	P1 : Time	
Drain & Return	P1 : Time	
Dump Wine		
Load Wine for Removal		
Plunge	P1 : Time	
Pump Over	P1 : Time	
Sparge		
Valve to Valve	P1 : Time	
Cold Stability Test		R1 : Stable True or False
Heat Stability Test		R1 : Stable True or False
Cap Management	P1 : Time	
Filterability		R1 : Time 1 R2 : Time 2
Add Nutrient	P1 : Additive ID P2 : Amount	R1 : Additive ID R2 : Amount R3 : Batch ID
Smoke Taint		R1 : Guaiacol R2 : 4 Methyl Guaiacol
Reset Wine Volume	P1 : Original Volume	R1 : Final Volume
Laccase Test		R1 : Result
BSC Clinitest Test		R1 : Clinitest result
BSC Residual Sugar Test	P1 : Kit ID P2 : Blank A1	R1 : Fructose concentration result

Worknote	Parameters	Results
	P3 : Blank A2 P4 : Blank A3 P5 : Standard A1 P6 : Standard A2 P7 : Standard A3 P8 : Standard Concentration P9 : Dilution P10 : A1 P11 : A2 P12 : A3 P13 : Reference	R2 : Glucose concentration result R3 : Residual Sugar concentration result
Barrel Topping	P1 : Consolidate? Yes or No P2 : Topping Wine ID	R1 : Final Volume R2 : Topping Wine Used R3 : Topping Wine Remaining
BSC pH TA Test		R1 : pH R2 : T.A.
BSC Malic Acid Test	P1 : Kit ID P2 : Blank A1 P3 : Blank A2 P4 : Standard A1 P5 : Standard A2 P6 : Standard Concentration P7 : Dilution P8 : A1 P9 : A2 P10 : Reference	R1 : Malic Acid concentration result
General Winery Worknote		
Add Grape Concentrate	P1 : Additive ID P2 : Amount	R1 : Additive ID R2 : Amount R3 : Batch ID
Pinking Test		R1 : Result

Table 33. Worknote Details

It should be noted that the results parameters only take effect when the worknote is completed, and readings for wines or blocks take precedence over worknote results for reporting purposes; the readings are only created when the worknote is completed.

14.2 Adding new items

The database and model have been designed so that any new technology or material that needs to be added to the system is immediately configurable without needing to recompile the application. However, preparations for vintage exposed the application to changes because of new processes being devised and new materials and techniques introduced that required either new menu options or dialogs. The database and model required no modifications because of these additions, just the interface.

15. TESTING THE SYSTEM

The process of testing the system was based on standard commercial testing regimes, although in practice it was performed with less formality and bureaucracy than a real commercial testing process. In essence testing can be regarded as serving three separate functions...

1. To remove bugs and errors in the processing of input data (Unit/Function Testing)
2. To ensure that the system is robust for use in a practical sense. (System /User Testing)
3. To ensure that the system fulfils the expectations of the requirements. (Product Verification)

Each of these phases has a different means of investigation and verification. The point of having a testing regime in the context of this project is not as crucial as a commercial system because the point of the project is not to produce a commercial system, but rather investigate the possibilities, capabilities, and efficacy of a data model for the wine industry, and so the testing need only satisfy this criterion. In this chapter, we briefly discuss each phase of testing, and then list and discuss all the major issues and enhancements that came about as a result of this process.

15.1 UNIT AND FUNCTIONAL TESTING

A unit is defined as a component or subsystem of the system that contains a distinct function or purpose. For the programming environment used for this project, units can be defined as individual forms, class objects, module functions and procedures, user controls, and designer objects. Unit testing is used to make sure that the unit is internally free of errors by ensuring that all possible inputs and outputs can be handled properly. Each unit has its own functional specification declared at the top of the code to ensure that any changes or bug corrections do not alter the overall intention of the unit.

After a few rounds of user testing during the initial installation phase, most of the programming bugs were isolated and fixed. Many bugs were caused by poor user input checking and insufficient data validation. Each time a unit was added or modified, the lessons from previous bug fixes were commonly implemented to ensure that such problems were not repeated.

Up to the date of submission of this thesis, there have been 72 reports of system errors occurring that might be regarded as failures in unit testing. Of these, 18 were found within days of the first release of the system, due to configuration problems and the fact that it was the first time multiple users were accessing the database at the same time. Of the remaining reports, most were caused by subtle alterations to the inputs and outputs of the units as a result of changes from system and user testing. The solution to these problems generally came about by more thorough error checking and greater control of user input through interface control design.

15.2 System and User Testing

The user interface is regarded as a separate layer in the architecture of the system, and is subject to unique pressures because it must deal with user inputs, in the form of keyboard and mouse inputs. This leads to a whole new set of issues for testing the system because the designer cannot always envisage how a user is going to interact with it. This was particularly obvious for some users who do not typically interact with computers as often as became necessary with the introduction of this system. For example, it was found that users were double-clicking on command buttons which caused some processes to be executed twice and simultaneously, leading to system crashes. Often, it was possible to change user's behaviour by generating messages, beeps, or disabling or hiding screen features to reinforce the error. However, at a primary level it is necessary to ensure that inputs are tested and verified and inappropriate values prevented from being generated and introduced into the system.

15.3 Product Verification

Product Verification is the hardest but most important phase of testing the system. The criteria for acceptance or verification are less obvious because they are not as clearly defined as the previous phases. At the start of this chapter, we referred to the purpose for carrying out testing was to verify that the system satisfied the requirements of the project. But the requirements, as stated in chapter 9, were simply the manifestation of idealistic and optimistic statements of what might be achieved according to chapter 2. In essence, success or failure can only be gauged in the eye of the beholder, because the criteria are subjective. However, taking a leaf from the winemakers' book, subjectivity is not an excuse to ignore or separate this information from empirical evidence. It just requires a more pragmatic and internally consistent basis to evaluate it.

The evidence we sought came in the form of user feedback. This information not only highlighted problems, but also provided criteria for the solutions, if not the solution itself. The feedback generally fell into 4 main categories

1. Achieving requirements and specifications
2. Performance with respect to multi-user access and overall speed and responsiveness
3. Maintaining data integrity
4. Enhancements

15.3.1 Requirements Met?

From the outset, the requirements of the system were the accumulated expectations of several key individuals. Naturally, these expectations came with their own assumptions and perspectives. It was particularly interesting to observe users when they encountered functions that were not part of their own expectations, but soon embraced them. For some, this gave them a different perspective and understanding of what the system could do and how it should work. For others, it was only through repeated use that the benefits became apparent.

The shift in expectations meant that the requirements changed and grew. The controlled and deliberate development process shifted to an iterative process of improvement as users became more familiar with the overall philosophy of the system. In retrospect, this was perhaps a little unfair for some users because the project evolved from a system in which they were masters that dictated the direction of development, to observers being confronted with an amalgamation of many ideas.

The implication of changing to an iterative development is that it becomes a continuous process. It is easy to see that each of the functions specified in Chapter 9 have been accommodated. The question of whether these have satisfied the original statements in Chapters 1 and 2 is now moot because the data model has become the impetus for driving the on-going development. Perhaps this is an indication of the success of the design of the model as fulfilling the requirements?

15.3.2 Performance

The performance of a system is usually based on how well it responds to user inputs and how quickly it completes tasks. Of course, these factors are controlled by the speed and efficiency of each of the constituent parts of the system, and the volume of data it must accommodate. Although the architecture of the system allows for system components to be distributed on either the client or the server computers, it was simpler to implement all the processing on the client computer and just use the server to store the database and the executable. As the data accumulated over the 2010 and 2011 vintages, it was noticeable how the performance of the system deteriorated. This provided a useful service by highlighting the inefficient code and data structures in the model as it became increasingly populated.

In general there were three major issues that became apparent...

1. It was discovered that sorting functions are more efficiently performed by the database and SQL techniques rather than sorting within the model. This meant that when new records were added to a collection, it was sometimes necessary to refresh the whole collection to maintain the sort order.

2. Some features of the data model rely on structures being built by the model rather than direct reference to data from the database. Like the sorting issue stated previously, this was found to be faster to implement from within the data refreshing process by retrieving the relevant data in a complex SQL command string rather than coding a sorting algorithm to rebuild the list in memory.
3. The server and client computers at the test site were about 7 years old and of relatively low specifications which caused significant performance delays in comparison to running the system on a newer computer with more RAM.

15.3.3 Data Integrity

By far the greatest threat to the success of the project came from allowing ambiguous or contextually erroneous data to be entered into the system. If the data loses context or order then it becomes misleading and can lead to misinterpretation. The key to the model's ability to keep track of events and actions is to ensure that the data faithfully records all the important features of these actions in such a way that the story can be effectively retold and information can be derived and interpreted from these points.

It was particularly interesting to attempt to rebuild the history of some wines in the wine tracker, when the sequence of events was long and complex. This became more obvious when, during the early stages of the data model testing, when some of the complex jobs had not been developed, the rationale and causal trail of action event and reaction were not always clear because the worknotes were not grouped into jobs that clearly indicated the intention of the actions being taken. The discrete actions were recorded but the overall plan was not clear because the jobs structure was not yet being exploited properly.

15.4 Enhancements and Key Issues

Testing the data model found and generated many bug fixes. In addition, as the users became more familiar with the system, the more readily they came up with new ideas, which led to several major enhancements. It also raised several issues that did not have clear-cut resolutions. What follows here is a list of these issues, some with resolutions and some requiring further consideration.

15.4.1 Planning – Overkill?

The vintage planning sections of the application were perhaps the most difficult concepts for many users to grasp. This was not because the task was unfamiliar, but rather because this was a visual manifestation of a process that was usually carried out in their heads. It could be argued that having an elaborate set of functions is an over elaboration of a simple process, particularly for small scale enterprises. For example, not allowing fruit to be received unless it was set up in the plan with vindefs and vessel allocations was confronting to some winemakers when they were using the system because it seemed to be contradictory to one of the original principles of designing a system; it should not alter the winemaker's natural method of working. However, another principle must also be considered here, and that is that the user needs to take the role of data collection and recording seriously so that the system can help do its job in a timely and appropriate way. Allowing random parcels of fruit into the winery without consideration for the vintage plan is not appropriate. The need to be flexible means there is also a need to be organised. A plan is a means of understanding the future; being ready for contingencies and emergencies, and being prepared to take advantage of circumstances.

15.4.2 Worknote and Job Sequencing

The design of the jobs and worknotes structures within the data model was based purely on ethnographic evidence of its relevance as observed and reinforced verbally by many winemakers. It reflects the reality that individual tasks are logically grouped together when an overall job is defined in the context of the winemaking process. However, the implementation of this in the application was flawed because it was based on an ideal situation that did not allow for mistakes, or circumstances that required a different strategy. The best way to explain this is by example.

In this case we have three separate jobs, each with an individual worknote, but they could equally have been three worknotes under one job.

1. Transfer 500 litres of Wine A in Vessel X to Empty Vessel Y, called Wine B.
2. Add 20 mg/l of SO_2 to remaining Wine A
3. Transfer 500 litres of Wine C in Vessel Z to Vessel X

Clearly, the order these tasks are completed have a fundamental impact on the end result; and this is why the completion sequence in the system was originally strictly enforced. However, several key events occurred that forced this strict regime to be loosened. The first was when the winemaker, who had taken complete control over data entry of the system, took several days off. Upon returning to the winery, dozens of jobs required completion in the system. At the time, worknotes were being date and time stamped automatically on completion with the current date and time. This was therefore replaced with the Completion Dialogs (section *13.2.5)* to allow the completion dates to be back dated for all those worknotes that had been completed earlier. The winemaker went through all the jobs, completing them and specifying the completion date, but soon found that it was not possible to set the date properly because jobs that had been completed earlier, but with later actual completion dates , caused the system to restrict the completion date time stamp to after these times. In other words, the worknotes had been completed in the system in a different order to how they had been completed in reality.

It was therefore necessary to loosen the rules even more, to allow the dates to be back dated to earlier than the last completed task's time stamp; with the proviso that this was to be used diligently and carefully so that the time stamps did not cause conflicts. This rule was then ignored completely because the user did not understand the consequences, and assumed that the system would be smart enough to work out what was meant.

The consequences can be understood by using the example. Task 2 was actually completed after task 1, but the worknote was completed and time stamped at an earlier date. When wine is transferred from one tank to another, two new wines are created; the remaining wine in vessel X gets a new ID and the new wine in Vessel Y is new too. However, the SO_2 addition was applied to the new wine in Vessel X, but the date it was added appears as before the wine was actually created because the date time stamp was input incorrectly.

Unfortunately, the system cannot be made smart enough to assume it understands and can interpret the meaning behind a completion time stamp. Such ambiguities can occur because...

1. The completion sequence could change
2. The order of notification of completion of events could differ from the actual order
3. Errors could be made in recording the sequence

This unfortunate situation required a great deal of remedial manipulation of data to resolve. The important lesson here was that the programmer must assume the role of protecting the system and the data from the ignorance or even malevolence of the user. The resolution was to partially roll back the latter changes that allowed back dating completion time beyond previously completed actions. Some exceptions were identified such as additions and some non-transfer actions where no volume change or new wines were created, in which case they could be backdated beyond the latest date of any similar actions.

15.4.3 Wine Names

The data model does not use the name of a wine as its identifier. This leaves the wine name free to be used by the user in any way they feel appropriate. This was discussed earlier in section *9.1.2.6 Naming conventions for wines*. However, this led to further confusion for some users because, if the original name for the wine was automatically generated, why not the subsequent wines? The model has algorithms to generate and derive new names and incremented names but how these are used became an issue. The main problem was that opinions as to how this should operate ranged from indifference to a demand for the complete encoding of LIP data into the name. The ability to configure the data model to suit all the possibilities became as confusing as the issue.

As far as the internal workings of the data model are concerned, wine names are not used as identifiers of the wine. In fact, the name of the wine is a parameter like any other. The initial generation of a wine name based on the ferment details was universally accepted as reasonable and standard practise, although smaller enterprises often did not even require this because they often only had one small parcel of any particular variety of fruit anyway.

The issue comes about later in the process when some winemakers change the nature of the names of the wines to reflect the batches they belong to, or perhaps even start indicating the end use of the wine by attaching a code for the brand label. The system does attempt to increment names when a new wine is created; when a wine is transferred or filtered. However, this is only a simple and wholly unsophisticated algorithm that is simply trying to keep wine names unique; a hangover from the original specification of the system that assumed greater control over the names of wines.

15.4.4 Barrel Sets and Wine Splitting

The concept of barrel sets was originally regarded as a means of simplifying the handling of wines in multiple vessels. It is difficult to find the origin of the idea because when it was first suggested, most winemakers already understood the concept and had informally adopted it as a standard management mechanism, even if it was not formally named or documented. This made the design of the concept easier because the intricacies had already been worked out and it was believed that any issues had already been thought through.

However, this was not the case because there was a loophole in the logic that allowed winemakers to switch from treating a wine as a homogenous single entity, but secretly maintaining the option of splitting the wine into separate components as they perform and behave differently. This is misleading for two reasons,

1. Topping a wine in a barrel set does not mean each vessel got the same amount of the topping wine, so each has a different but unknown LIP. Only the overall LIP is known.
2. The history of the wine from a tracking perspective is false and misleading because although the wine has been a separate object in reality, the data model has not treated it as such because it has made assumptions that come with the advantages of treating the wine as a single object in a barrel set.

If some of the barrels are cut out and treated as a separate wine, the system cannot reverse engineer a new history for this wine. The granularity of data never existed. The problem is that the only way to do this in the system is to transfer out the appropriate wine to another vessel, then transfer this wine into the individual vessels once they have been removed from the original barrel set. The new wine's history begins at the moment the wine is transferred out, but in actual fact it has existed as a separate entity for perhaps weeks or months before hand.

This situation arose on several occasions. The fortunate aspect on each occasion was that it was possible to separate out components, because the wine had always been topped by consolidation from a small VC, and so the LIP never altered. Unfortunately, this loop-hole still exists in the system. There seems no solution if the user is prepared to accept the lack of tracking integrity, and perform the split. In most cases, the difference in LIP from vessel to vessel is insignificant, and the loss of history is ignored.

A minor addendum to this issue is the term "Barrel Set". It was argued that the term should be replaced by the term "Vessel Set" because any mobile vessels in a winery can be included into a barrel set, by definition. The case study site had a large number of different types of smaller and mobile vessels, including, barriques, hogs-heads, flextanks, pallecons, and variable capacity tanks from 1000 to 200 litres. The construct of a barrel set was based on the mobility of the vessels, not the material from which they were made. In fact, many wineries have large static, open oak vessels used to perform open ferments.

15.4.5 Wine Volumes

A significant factor in keeping the data model relevant is to keep accurate and precise data. When we are dealing with vessels with capacities varying from hundreds to tens of thousands of litres, the issue of accuracy and precision becomes significant. It was found that the measurement of the volume of wines was made using the most pragmatic method available at the time. If wine was transferred to barriques, then the barriques were counted to calculate the volume. When wine is transferred to a tank, then a flow-meter might be used, or an estimate made based on a dip or ullage measurement at either the source or target tank. This can often be misleading. There is an assumption that all vessels of a particular type have the same volume, so that a 200 litre ullage in each of two vessels of the same type means that they have precisely the same volume. It is easy to see how the nominal volume of a wine can vary based on whichever method is used to measure it. Often these changes are attributed to losses due to evaporation, or taste-offs when wine is transferred, but it may well be just a measurement issue, rendering the rationalisation and accounting of losses moot.

15.4.6 The Data Model Language

It is clear from testing and the nature of some of the objects in the data model that the Visual Basic 6.0 class and object structures and collections are inadequate to fashion an efficient model. Apart from performance issues, there were two other major short-comings.

15.4.6.1 Model Refreshing

This problem was alluded to in the previous section of this chapter (section *15.3.2 Performance*). It was often convenient to have the data in the model pre-sorted so that some searching criteria could be assumed. This was particularly important for collections like wine readings where they were assumed to be in date and time order, so the latest reading would always be at the end of the collection. However, searching and manipulating these structures, particularly when different sort orders were required, was grossly inefficient in this environment. The problem is not with the design of the model but rather with the so-called "Object Oriented" environment. The solution, as stated earlier, was to use SQL features to achieve the desired results, but a proper solution would be to have or build better standards for the objects themselves. It is almost certainly true that having a thinner client and moving the data model to the server as a server component service would alleviate this, but not all enterprises will have a server, and a more functional modelling language is a better solution than a patch-up.

15.4.6.2 Back References

It can be noted that in some objects, there is a property that refers to the ID of a parent entity that would otherwise be easy to determine by referencing it within the model. For example, the VinDef object contains the property WineryID that keeps the ID of the winery that the vindef belongs to. In a normal object oriented model, this should be a simple matter of referring back to the parent or container to determine the ID, however this is not an option in Visual Basic 6.0, so the ID has been copied into the vindef object, to save the need for searches to determine the parent object ID.

15.4.7 Use of assumed and suspect

It was found that readings flagged as suspect or assumed, were misleading in that they were date-time stamped when they were set as assumed or suspect, and the date-time stamp was not the date of the reading. This was particularly important for the data dump and scoring system which was giving misleading results in that readings were appearing newer than they really were, because they were the date of the flag setting, not the reading. The reason for this was that the time stamp was there for use by the data model, not for reporting to the user. The data model would always understand the meaning of the time stamp, but users are not so disciplined. The monitoring score system and data dump was modified to use the reading date from un-assumed and non-suspect readings only.

15.4.8 Some Rejected Ideas

15.4.8.1 Enzyme Kit Corrections

Calculations for the enzymatic tests include determining the so-called “recovery” rate of the analysis by comparing the concentration of a known standard against the analysed result for that standard. A recovery rate of 90% or better is regarded as adequate, however 95% or better was recommended by the manufacturers. It was proposed by some laboratory staff that the recovery rate could be used to generate a correction factor...

$$\frac{\text{standard concentration}}{\text{recovery fraction}}$$

to recalculate the results. This effectively adjusted all the data to assume a 100% recovery of the standard.

This correction was dubious in the opinion of the author, as it seemed to assume an understanding and acceptance of a linear relationship between the cause of the error, whatever that may be, and the analysis results. The manufacturer confirmed that this correction was inappropriate and so it was excluded from the calculations algorithm.

15.4.8.2 Wine Homogenisation

Amongst the suggestions for new jobs and worknote types, there was a request for a “Homogenise” wine function so that two wines could be made equivalent to each other, with the same LIP, as if the wines had been blended in a single tank, then split back into the two source vessels.

The problem here was that the process to create the two wines was described as a double valve to valve transfer from one tank to the other and back. It was proposed that this would homogenise the two wines if left to run for several hours. However this was rejected because the process cannot be said to actually homogenise the two wines.

16. Conclusions and Further Research Possibilities

Proper judgement of the data model, as a structure built from object-oriented concepts, designed to capture and manage the activities in a winemaking enterprise, needs to be undertaken from different perspectives.

- As a stand-alone framework for the data, the model works well and manages the inputs and outputs appropriately.
- As a platform for developing an advanced user interface, it also worked well up to a point, but often required enhancements when the expectations from the application became more advanced.
- From a winemaking and grape-growing point of view, the model needs to show more about how it can be exploited.
- From a programming perspective, it needs to be thoroughly re-worked to properly exploit object oriented design principles.

This is not to say that the model has been a failure, because evidence and feedback from the testing site suggest that the system will continue to be used as an on-going means of managing the winery and vineyard. The fact is that the system, with all the grand and optimistic ideals that accompanied it at the beginning, still fell short on some points.

The assumption that the system would be intuitive and provide a seamless transition from previous systems was naive and overly-simple. Any move to modify a system and impose a new regime, particularly if that original regime was built out of years of experience, will always have an adverse initial impact that may last longer than anticipated. Indeed, any attempt to formalise or manage this process tended to cause the nature of the process to change. The impact of the data model on the process was always minimised as much as possible, and the intention was always to enhance the process not hinder it. The problem is that not everyone starts from the same position or perspective, and so the adaption was more difficult and more stressful for some than others.

Once the model was in action, and particularly after the back-entry of data from previous vintages, the system provided a simple and effective means of performing the main tasks for which it was designed. This included reporting previous activities, the current status of the enterprise, and planning and designing activities for the future. The Monitoring screen became the instigator for most of the pre-emptive activities in the winery. The Winery Map screen provided the main browsing function, and the wine tracker was soon adopted as a first port of call for investigating the history and provenance of a wine. Over time, the interface was improved based on user feedback. The jobs and worknotes became more complex, and the reports themselves were an interesting exercise in graphic design. In particular, there were requests for more sophisticated reports and changes to the de-facto data mining function. It was interesting to note that each change made and implemented in the system was accompanied by a corresponding increase in confidence in the system by the users. This was most noticeable when navigational short-cuts were implemented so that contextual information became more easily accessible. The more situations encountered, the better the navigational pathways could be made. The increased functionality and perceived responsiveness to their requests meant that even the institutionalised users were able to start thinking more freely and use more of their intuitive skills.

The added formality that the system imparted into the enterprise also had an effect on the discipline and professionalism of the winery. Cellar-hands were required to be more disciplined with reporting the results and outcomes of their work. Job print-outs were returned with all the relevant results and information so that the system could be updated immediately. This allowed senior managers to see the status of activities as they were completed.

Clearly, the data model requires the positive participation of its users. If an operator chooses to use the system just for record keeping, without taking advantage of some of the models advanced planning and operations features, then it might be said that the model is cumbersome. This is because the user needs to have a degree of enthusiasm for keeping the model up to date and to have what might be described as a healthy respect for the importance of diligent management. For example, if yield estimates are too erroneous then this has a large impact on the planning and logistics function of a winery, and thus the effectiveness of the plans diminish as the seeding for operational functions become less precise. This is true, regardless of whether the model is being used or not. In a sense, it is reassuring (and indicative of the relevance of the system) to note that difficulties arise when plans and estimates diverge significantly from reality. When there is substantial excess or deficit of fruit from a particular block or defined parcel, the user is forced to go back and modify the plan to accommodate the discrepancy. At first, this was perceived as a deficiency in the model, or at least the interface, because it might be seen as step backwards to resolve the issue or a lack of flexibility in the system. But upon reflection, this is quite serendipitous. After all, even without the model in use, the operator needs to rely on reasonable accuracy of the estimates in order to ensure that winery can cope logistically.

It is true that a better designed user interface would be able to implement the model more satisfactorily. The intention of this thesis was not to build a complete system, but rather the underlying data model. If a situation occurs that the interface cannot present a simple solution or provide a means of resolving such situations, it is not the model that is at fault, but rather the implementation.

Therefore, the model is not fool-proof in the sense that it is only effective when it is provided reasonable information. It copes well within the normal bounds of operations of an enterprise. It even provides advanced functionality when disasters occur, or unexpected situations arise, but it is not a panacea to incompetent management. Nor does is aid or abet fraudulent activities. Deliberate manipulation of volumes, fruit provenance, additions, and other L.I.P. constructs are not easily disguised. The following sections describe some of the major areas for further research.

16.1 A finger tip interface

Given that the original specifications for the GUI were to simply support and demonstrate the functionality of the data model, the prospect of properly developing the user interface to its greater potential is an exciting possibility for future development. In particular, the interface can incorporate newer technologies such as touch-screen functions introduced lately in new mobile phone technology. Similar features should soon be available commonly on laptops and desktop computers,

These new interactive technologies could be used to represent and instigate many functions that currently use the common menu system, or command buttons. Many suggestions were made during the testing phase, but time was restricted and these suggestions would not have any impact of the data model. They all involved either greater cross-linking navigation between features in the system or better use of visual cues to reinforce text information.

Another field that might benefit from rework is the input of subjective data, which was basically put to one side in most of the application as there were no appropriate scales developed by the users in time for publication. Visual and fingertip interactions are more accessible and usable, and intuitive than option lists or slider controls that require deft mouse or keyboard skills.

16.2 Configuration Application

The proposed configuration application was not completed due to time constraints. The application that was built allowed the modification of users and of some code-set tables but still requires work to update all the remaining tables, and a few other features required by the users.

16.3 Location Objects

The winery facility type needs to be modified so that it is possible to define more areas or locations within the facility, apart from a Tank Farm and a Barrel Hall. This requirement became evident late in the project when it was discovered that it made better sense to define reefers (refrigerated containers) as areas like the barrel hall. This allows readings for vessels and wines to include a location reading so that the period of time spent in a reefer can be calculated for the purposes of tracking wines and cold stability processing.

16.4 Blend Samples

It was found very late in the project that it would be very useful to use the system to manage the sample blends that are part of the final blending process for some wines in the facility. Such wines needed to have analyses performed and needed to be treated as distinct objects. It is proposed that a separate node and object should be defined, to be known as the blend library, to contain these blend objects and allow them to be managed, analysed, and used to generate blending jobs and worknotes.

16.5 Jobs created on the fly

Another aspect of the system that came to light only after it had been used for some time was the possibility of making jobs and worknotes dynamically rather than from a set menu. The idea was proposed that it should be possible to add and remove any worknotes types from some jobs. This then expanded to allow for any worknote type to be added to any job, and perhaps even the ability to create free-form jobs. This concept would remove the need for the extensive menu required on the interface and it is also plausible that the fixed format jobs could still be useful by acting as templates to apply to a new general structure rather than a pre-fabricated but fixed structure. This is perhaps the most interesting possibility for further investigation because it takes the model and application beyond the original scope of the project which was to ensure that winemakers do not have to change the way they think about their jobs. This scenario offers the possibility of utilising advanced user-interface concepts and technology to develop a free-hand virtual approach to winemaking. Perhaps even more importantly, it allows for increased significance and prominence of the cause and effect in the psyche of the model’s users because the user would either need to generate a note to make that intention clear, or the system would need to be modified to build a job description based on the worknote types selected.

16.6 More Advanced Monitoring Algorithms

As the users became more familiar with the Monitoring function of the application, it was found that the simple algorithms of applying points for readings levels and times was inadequate. Some wines scored quite low scores but needed immediate actions because of factors such as ullage levels, perceived taints. Other wines had higher scores only because they had been left longer without a reading being made. The issue was complicated by the fact that assumed and suspect readings were excluded from calculations, and new wines resulting from transfers or toppings are given a clean slate for readings which leads to artificially high scores.

One major idea was to restrict the maximum score to 10 or 100; giving the score a better comparative quality and act as a guide to prioritising scoring values. Needless to say this issue requires a great deal more consideration.

16.7 Connection to other systems

Although a generic solution to this issue is not possible because external systems will always differ in the degree and manner that they interact with external systems, it is conceivable that a great deal of benefit to the value if the data and convenience to the users can be achieved by linking other systems.

At the test site there were two main possibilities. The first was the stainless steel vessel temperature control system called “Vinwizard” (7.1.1.10) which uses probes to measure the temperature of the vessel, and solenoid controlled brine lines to cooling jackets and heating panels to control the temperature. Currently, the settings in Vinwizard need to be duplicated in the data model system because there is no means of accessing the Vinwizard database. The second possibility is the spectrophotometer which it is believed can send the absorbance readings out through a communications port at the back of the unit.

16.8 Data Mining and Reporting Functions

The data dump report currently exists as a simple proof of concept for the data model rather than a useful and definitive tool for the system. The issue is that not all the data from the model is displayed in the spreadsheet because it requires a multi-dimensional grid to accommodate the data.

Currently, the report operates in two dimensions (wines and readings) as shown in the diagram below...

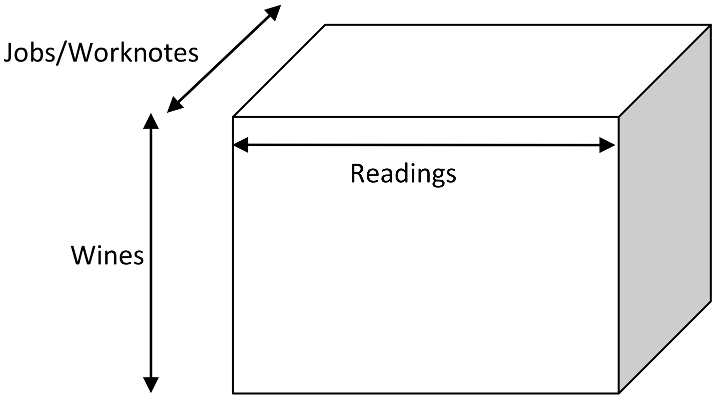


Figure 132. The data mining data grid

can view the data from different perspectives and perform more meaningful data mining analyses.

16.9 A BETTER STANDARD OF CODING

In the final analysis, the most significant area in which the model needs to improve is the language and standard of code in which it is written. Visual Basic 6.0 is not a genuine object-oriented environment and so the whole basis of the data model is compromised from the beginning. It is known that Microsoft's .NET architecture is a better prospect for writing the application, and can be easily translated from the current code by using the Visual Basic .NET architecture and language.

17. BIBLIOGRAPHY

1. Boulton. R.B., Singleton. V.L., Bisson, L.F., and Kunkee. R.E. (1999). *Principles and Practices of Winemaking***.** Kluwer Academic/ Plenum Publications, New York.

2. Burrell, J., Brooke, T., Beckwith, R. (2003), "Extending ubiquitous computing to vineyards", *Conference on Human Factors and Computing Systems, CHI '03 Extended Abstracts on Human Factors in Computer Systems*, ACM Press, New York, NY, pp.822-3.

3. Caillard, A. (2008) "The Rewards of Patience". Allen & Unwin. Crows Nest, NSW.

4. Gleazer, E.A., (1990) "Wineries get with the program – computers are the newest tools in the wine. " Wines & Vines, March, 1990,

5. Iland,P., Bruer,N., Edwards, G., Weeks, S., and Wilkes, E., (2004) *Chemical Analysis of Grapes and Wine: Techniques and Concepts.* Patrick Iland Wine Promotions Pty Ltd, Campbelltown, South Australia.

6. Iland, P., Bruer, N., Ewart, A., Markides, A. and Sitters, J. (2004) *Monitoring the winemaking process from grapes to wine: techniques and concepts*. Patrick Iland Wine Promotions Pty Ltd, Campbelltown, South Australia.

7. Iland, P., Grbin, P., Grinbergs, M., Schmidtke, L., Soden, A. (2007) *Microbiological Analysis of Grapes and Wine: Techniques and Concepts.* Patrick Iland Wine Promotions Pty Ltd, Campbelltown, South Australia.

8. Michalewicz, Z., Schmidt, M., Michalewicz, M., Chiriac, C. (2007) "Process Improvement for Profitability using Adaptive Business Intelligence Software". 13th Australian Wine Industry Technical Conference, Workshop 06 and 52 .

9. Rankine B. C. (2004) Making Good Wine: A Manual of Winemaking Practice for Australia and New Zealand. MacMillan, Sydney.

10. Wollan , D., Kelly, M., and Baxter, N. (2002) "Winery Management Systems – Some Thoughts from a Developer" The Australian & New Zealand Grapegrower & Winemaker. July 2002.

11. Shanmuganathan, S., Sallis, P., and Narayanan, A. (2009) Unsupervised artificial neural nets for modelling the effects of climate change on New Zealand grape wines. B. Anderssen et al. (eds), 2009. 18th IMACS World Congress - MODSIM09 International Congress 13-17 July 2009, Cairns, Australia. ISBN: 978-0-9758400-7-8. pp. 803-809. Australian Wine Industry State of the Environment 2003 http://www.wfa.org.au/PDF/Environment2003.pdf

12. Wilson. J.A. "Winery Management System", unpublished

13. Personal communications and interviews with...

- Peter Wilson, Chief Winemaker, Stuart Wines.
- David Barber, Whitebox Vineyard Manager, Stuart Wines
- Bailey Carrodus, Owner/Winemaker, Yarra Yering Wines.
- Kirsty Waller, Viticultural Manager, Barossa Valley Estate Wines
- James Halliday, Wine Critic.
- Peter Gago, Chief Winemaker, Penfolds
- Daryl Rathjen, Grapegrower, Colbinabbin, Victoria
- Tim Hampton, Manager, Yarra Yering Winery and Vineyard
- Stuart Bourne, Chief Winemaker, Barossa Valley Estate
- Louis Decastella, Winemaker, Louis Decastella Wines
- Peter Peyton, Consultant Viticulturist, Yarra Valley
- Stephen Phillips, Winemaker, Stuart Wines

18. Appendix A : Student Survey

This brief survey is part of a PhD thesis on the management of grape growing and winemaking enterprises. It is intended to provide anecdotal evidence on the background, experience, and motivation of people within the winemaking industry.

You will note that you are not required to provide your name.

No personal information will be kept, nor will individuals be identified in the results.

If you are not comfortable answering any of the questions, please cross it out.

The survey data will be collected and then these sheets will be thoughtfully and humanely destroyed.

Thank you,

James Wilson

Personal Details

Year	1st Year	2nd Year	3rd Year	Grad. Dip.
Course	Oenology	Viticulture		
Gender	Female	Male		
Age	□			

Other Qualifications

Please specify any other tertiary qualifications you possess.

Wine Industry Work Experience

Please provide a brief list of any roles you have performed in the wine industry.

Other Work Experience

Please provide a brief list of any other industries you have previously worked in.

Do you have any family members working or studying within the wine industry?

Please indicate the extent to which the following aspects of the course or external factors have influenced or motivated you to study in this field.

(Tick one box for each row.)

	Not at all	Slightly	Significantly	Strongly	Primary Influence / motivation
Family involvement	□	□	□	□	□
Lifestyle	□	□	□	□	□
Money, Financial reward	□	□	□	□	□
Love of Wine	□	□	□	□	□
Marketing	□	□	□	□	□
Accounting	□	□	□	□	□
Science (chemistry, biology etc)	□	□	□	□	□
Business Management	□	□	□	□	□

Any other factors that have influenced your choice.(Please specify)

Information Technology Experience

Please provide an indication of your experience with computers, in the I.T. industry or with software programs. (Please, no game consoles)

Aspirations for the future

Please indicate what roles you aspire to in the wine industry. Do you aspire to managing a winery or vineyard? Do you want to become an owner of a business? Do you have a role or position organised after you complete your course?

END

Thankyou.

19. Appendix B : Database Tables

t_AdditivesStored	
ID	Long Integer
AdditiveID	Long Integer
Batch	Text
Cost	Currency
Complete	Yes/No

t_BottledWines	
ID	Long Integer
StorageID	Long Integer
Description	Text
WineID	Long Integer
LabelID	Long Integer
ClosureID	Long Integer
CapsuleID	Long Integer
BottleType	Long Integer
BoxType	Long Integer
DividerType	Long Integer
PadType	Long Integer
WNID	Long Integer
BottleCount	Long Integer
Top	Long Integer
Left	Long Integer
t_CS_Additives	
ID	Long Integer
FacilityID	Long Integer
Name	Text
AdditiveType	Long Integer
ByVolume	Yes/No
Rate	Long Integer
RateFactor	Double
Trialable	Yes/No
AffectspH	Yes/No
AffectsTA	Yes/No
AffectsSO2	Yes/No
AffectsSugars	Yes/No
AffectsVolume	Yes/No
Sulphites	Yes/No
Egg	Yes/No
Milk	Yes/No
Nuts	Yes/No
Casein	Yes/No
Default	Yes/No

t_CS_DryGoods_BottleTypes	
ID	Long Integer
Description	Text
Style	Text
Colour	Text
Volume	Long Integer
Punt	Text
Make	Text

t_CS_DryGoods_BoxTypes	
ID	Long Integer
Description	Text
Capacity	Long Integer
BoxesPerPallet	Long Integer

t_CS_DryGoods_Capsules	
ID	Long Integer
Description	Text

t_CS_DryGoods_Closures	
ID	Long Integer
Description	Text
Capsule	Yes/No

t_CS_DryGoods_DividerTypes	
ID	Long Integer
Description	Text

t_CS_DryGoods_Labels	
ID	Long Integer
FrontImage	Text
BackImage	Text
Description	Text

t_CS_DryGoods_PadTypes	
ID	Long Integer
Description	Text

t_CS_DryGoods_PalletTypes	
ID	Long Integer
Description	Text
t_CS_ELVineStatus	
ELNumber	Long Integer
Description	Text
Image	Text

t_CS_EnzymeKits	
ID	Long Integer
TestType	Long Integer
LabID	Long Integer
Brand	Text
Factor1	Double
Factor2	Double

t_CS_FacilityTypes	
ID	Long Integer
Description	Text

t_CS_HeatCoolMethods	
ID	Long Integer
Description	Text
FacilityID	Long Integer
Heat	Yes/No
Cool	Yes/No

t_CS_JobTypes	
ID	Long Integer
JobName	Text
FacilityType	Long Integer
Enabled	Yes/No
JobDescription	Text
TargetSpecific	Yes/No
GenerallyAvailable	Yes/No
CompleteInSequence	Yes/No
NoDialog	Yes/No
StandingOrder	Yes/No

t_CS_JobWorknotes	
Job	Long Integer
Worknote	Long Integer
Sequence	Long Integer
CanBeCancelled	Yes/No
NeedsRatification	Yes/No

t_CS_PressingRegimes	
ID	Long Integer
PressID	Long Integer
Name	Text
Description	Text

t_CS_Regions	
ID	Long Integer
RegionDescription	Text
Code	Text
State	Text
Country	Text

t_CS_Varieties	
ID	Long Integer
VarietyDescription	Text
Colour	Text
Code	Text

t_CS_VesselCategories	
ID	Long Integer
Category	Text
OnSkinsFermentor	Yes/No
DefaultCapManagementWNID	Long Integer

t_CS_VesselTypes	
ID	Long Integer
Description	Text
Code	Text
CategoryID	Long Integer
CapacityTonnes	Double
CapacityLitres	Long Integer
Icon	Text
UllageAllowance	Long Integer

t_CS_WorknoteTypes	
ID	Long Integer
FacilityType	Long Integer
TargetObjectType	Long Integer
WorknoteDescription	Text
Instructions	Memo
RequiresResults	Yes/No
Editable	Yes/No
Addition	Yes/No
Transfer	Yes/No
RequiresParameters	Long Integer
RequiresTime	Yes/No

t_DBVersion	
Major	Long Integer
Minor	Long Integer
Revision	Long Integer

t_Diary	
ID	Long Integer
User	Text
FacilityID	Long Integer
DateCreated	Date/Time
ForDate	Date/Time
DiaryNote	Text
Start	Long Integer
End	Long Integer

t_DryGoodsStored	
ID	Long Integer
WineryID	Long Integer
DryGoodType	Long Integer
DryGoodID	Long Integer
Batch	Text
Cost	Currency
Complete	Yes/No

t_Equipment	
ID	Long Integer
Type	Long Integer
Description	Text
FacilityID	Long Integer
Icon	Text
State	Long Integer

t_Facilities	
ID	Long Integer
FacilityName	Text
Code	Text
RegionID	Long Integer
FacilityType	Long Integer
Managed	Yes/No
MainView	Yes/No
Sequence	Long Integer
Image	Text
SubtleImage	Text
LockedBy	Text

t_GUI_MenuImages	
ID	Long Integer
Facility	Long Integer
MenuType	Text
Image	Text

t_GUI_MenuLabels	
ID	Long Integer
FacilityID	Long Integer
NodeType	Text
Label	Text
LinksToNode	Text
top	Long Integer
left	Long Integer

t_Job_FruitReception	
ID	Long Integer
WineryID	Long Integer
ParcelID	Long Integer
JobID	Long Integer
ExpectedTonnes	Double
ExpectedYield	Double
Reception	Long Integer
HarvestMethod	Long Integer
PMSPerBin	Long Integer
OtherBinAdditive	Long Integer
OtherBinAdditiveRate	Long Integer
BinCount	Long Integer
Chill	Yes/No
Rest	Yes/No
Process	Long Integer
AnalyseSO2	Yes/No
AnalysepHTA	Yes/No
AnalyseTempBaume	Yes/No
AnalyseMalic	Yes/No
AnalysePAA	Yes/No
AddSO2	Yes/No
AddSO2When	Long Integer
AddH2T	Yes/No
AddH2TWhen	Long Integer
Yeast	Long Integer
YeastRate	Double
YeastWhen	Long Integer
Nutrient	Long Integer
NutrientRate	Double
Press	Long Integer
PressRegime	Long Integer
PressSplit	Yes/No
AsWhite	Yes/No

t_Job_FruitReception_Allocations	
ID	Long Integer
JFRID	Long Integer
WineID	Long Integer
VesselID	Long Integer
Tonnes	Double
Drain	Yes/No
Analyse	Yes/No
Additions	Yes/No
Yeast	Yes/No

t_Jobs	
ID	Long Integer
JobType	Long Integer
FacType	Long Integer
Heading	Text
StandingOrder	Long Integer
Vintage	Long Integer
DateTimeCreated	Date/Time
Status	Long Integer
SpecialComments	Text
TargetSpecific	Yes/No
Steps	Long Integer

t_MonitoringAlgorithms	
ID	Long Integer
AlgorithmName	Text
WineryID	Long Integer

t_MonitoringAlgorithmTests	
ID	Long Integer
AlgorithmID	Long Integer
Parameter	Text
TestType	Long Integer
TestIncrement	Double
TestThreshold	Double
TestPoints	Double

t_Owners	
ID	Long Integer
Description	Text
Address	Text
ABN	Text
Contact	Text

t_Plan_Parcels	
ID	Long Integer
Vintage	Long Integer
Description	Text
VineyardID	Long Integer
BlockID	Long Integer
WineryID	Long Integer
AmountType	Integer
Amount	Double
ExpHarvestDate	Date/Time
HarvestType	Long Integer

t_Plan_ParcelToVinDef	
ID	Long Integer
ParcelID	Long Integer
VinDefID	Long Integer

t_Plan_VinDefToWineBatch	
ID	Long Integer
VinDefID	Long Integer
Proportion	Double
WineBatchID	Long Integer

t_Plan_VinDefVesselTypeAllocations	
ID	Long Integer
VinDefID	Long Integer
VesselType	Long Integer
Amount	Double
Multiple	Long Integer
Priority	Long Integer

t_Plan_VinificationDefs	
ID	Long Integer
VinDefName	Text
Vintage	Long Integer
WineryID	Long Integer
FermentOnSkins	Yes/No
InitialProcessing	Text
Description	Text
Yield	Double
PressRegime	Long Integer
Split	Yes/No
Yeast	Long Integer
FermentationLength	Long Integer

t_Plan_VintageLogisticsVesselAllocations	
ID	Long Integer
VinDef	Long Integer
Vessel	Long Integer
StartDate	Date/Time
FermLength	Long Integer
VDVAID	Long Integer
Multiple	Long Integer
Amount	Double

t_Plan_WineBatches	
ID	Long Integer
BatchName	Text
Vintage	Long Integer
WineryID	Long Integer
Description	Text
Vessels	Text
MaturationLength	Long Integer
OakPercentage	Double
OakNewPercentage	Double

t_ScoringAlgorithms	
ID	Long Integer
AlgorithmName	Text

t_Skins	
ID	Long Integer
WineryID	Long Integer
PressWN	Long Integer
Description	Text
PressID	Long Integer
VesselID	Long Integer
WineID	Long Integer
Status	Long Integer

t_StandingOrders	
ID	Long Integer
JobType	Long Integer
Active	Yes/No
SpecialComments	Text
LabID	Long Integer
WineryID	Long Integer
VineyardID	Long Integer
StorageID	Long Integer
Target	Text
Basis	Long Integer
Hours	Long Integer
Days	Long Integer
Months	Long Integer
MonthDay	Long Integer
StartDate	Date/Time
ExpiryDate	Date/Time

t_TankMap	
ID	Long Integer
Name	Text
TypeID	Long Integer
WineryID	Long Integer
TankFarm	Yes/No
XPos	Long Integer
YPos	Long Integer
State	Long Integer
AutomatedHeating	Yes/No
AutomatedCooling	Yes/No
ManualHeating	Yes/No
ManualCooling	Yes/No
HeatingOn	Double
CoolingOn	Double
MOX	Yes/No
GasBlanket	Yes/No
SortOrder	Long Integer
BarrelSet	Long Integer
BarrelSetNotes	Text
Maker	Text
YearMade	Long Integer
Toast	Text
Material	Text
Unused	Yes/No
UllageInBarrelSet	Yes/No

t_TrackerDump	
ID	Long Integer
Userid	Text
ObjectID	Long Integer
ObjectType	Long Integer
TargetObject	Long Integer
Layer	Long Integer
Priority	Long Integer
WNID	Long Integer
WNTarget	Long Integer
WNCompletedDate	Date/Time
wnheading	Text

t_Users	
ID	Long Integer
UserID	Text
Name	Text
LoggedOn	Yes/No
Editor	Yes/No
Admin	Yes/No
SuperAdmin	Yes/No
Active	Yes/No
KillMe	Yes/No
AskBeforeExiting	Yes/No

t_Vessel_Readings	
ID	Long Integer
VesselID	Long Integer
ReadingType	Long Integer
ReadingValue	Text
ReadingDateTime	Date/Time
Vintage	Long Integer
WNID	Long Integer

t_VineyardBlockReadings	
ID	Long Integer
BlockID	Long Integer
ReadingDate	Date/Time
Vintage	Long Integer
WorknoteID	Long Integer
ReadingType	Long Integer
ReadingValue	Text

t_VineyardBlocks	
ID	Long Integer
VineyardID	Long Integer
BlockDescription	Text
VarietyID	Long Integer
Clone	Text
Area	Double
PlantedYear	Text
Rows	Long Integer
MenuX	Long Integer
MenuY	Long Integer
Vines	Long Integer

t_VineyardReadings	
ID	Long Integer
VineyardID	Long Integer
ReadingDate	Date/Time
WorknoteID	Long Integer
ReadingType	Long Integer
ReadingValue	Text

t_VWMS_AuditLog	
ID	Long Integer
Msg	Text
User	Text
MsgDateTime	Text

t_VWMS_SystemConstants	
SplashImage	Text
EnterpriseLogo	Text
EnterpriseName	Text
BarrelDepreciation	Long Integer
LogToFile	Yes/No
LogToDB	Yes/No
LabMessages	Yes/No
LastCompacted	Date/Time
CompactEveryHours	Long Integer
DeathDate	Date/Time

t_VWMS_SystemMessages	
ID	Long Integer
Generated	Date/Time
Msg	Text
Cleared	Yes/No

t_WineGroups	
ID	Long Integer
GroupName	Text
Winery	Long Integer
Category	Long Integer

t_WineryDeliveredParcels	
ID	Long Integer
WineryID	Long Integer
Vintage	Long Integer
PlanParcel	Long Integer
HarvestDate	Date/Time
Tonnes	Double
Docket	Text
ProcessingJobID	Long Integer
HarvestType	Long Integer
BlockID	Long Integer

t_Wines	
ID	Long Integer
Winery	Long Integer
OnSkins	Yes/No
Lees	Yes/No
MonitoringAlgorithm	Long Integer
WineGroup	Long Integer
RegionalClaim	Long Integer
VarietalClaim	Long Integer
VintageClaim	Long Integer

t_Wines_BSC	
ID	Long Integer
WineID	Long Integer
VesselID	Long Integer

t_Wines_BSC_Readings	
ID	Long Integer
WBSCID	Long Integer
ReadingType	Long Integer
ReadingValue	Text
ReadingDateTime	Date/Time
Vintage	Long Integer
WNID	Long Integer
Assumed	Yes/No
Suspect	Yes/No
AwaitingRatification	Yes/No

t_Wines_Chip	
ID	Long Integer
WineID	Long Integer
Chips	Long Integer
KGS	Long Integer
InWNID	Long Integer
OutWNID	Long Integer

t_Wines_LIP	
ID	Long Integer
WineID	Long Integer
Vintage	Long Integer
BlockID	Long Integer
Volume	Double
WNID	Long Integer

t_Wines_Readings	
ID	Long Integer
WineID	Long Integer
ReadingType	Long Integer
ReadingValue	Text
ReadingDateTime	Date/Time
Vintage	Long Integer
WNID	Long Integer
Assumed	Yes/No
Suspect	Yes/No
AwaitingRatification	Yes/No

t_Wines_SourceWines	
ID	Long Integer
Wine	Long Integer
SourceWine	Long Integer
WNID	Long Integer
Volume	Long Integer

t_Worknotes	
ID	Long Integer
JobID	Long Integer
WorknoteType	Long Integer
FacilityID	Long Integer
TargetID	Long Integer
Heading	Text
Vintage	Long Integer
DateTimeCreated	Date/Time
Status	Long Integer
Sequence	Long Integer
RequiredStartDate	Date/Time
CompletedDate	Date/Time
RequiresResults	Yes/No
P1-P15	Text
R1-R12	Text

20. Appendix C : System Constants

20.1 Heat Cool

HC_HEAT = 0
HC_COOL = 1

20.2 Facility Types

FT_WINERY = 1
FT_VINEYARD = 2
FT_STORAGE = 3
FT_LABORATORY = 4
FT_BOTTLING = 5

20.3 Object Types

OBJECTTYPE_WINE = 1
OBJECTTYPE_FRUITPARCEL = 2
OBJECTTYPE_BLOCK = 3
OBJECTTYPE_VESSEL = 4
OBJECTTYPE_EQUIPMENT = 5
OBJECTTYPE_WINE_BSC = 6

20.4 Menu Node Types

NODETYPE_WINERY = "WI"
NODETYPE_WINERYVINTAGEPLAN = "WP"
NODETYPE_WINERYVINTAGELOGISTICS = "WL"
NODETYPE_WINERYMAP = "WM"
NODETYPE_WINERYMAP_RATIFICATION = "WR"
NODETYPE_WINERYWEIGHBRIDGE = "WG"
NODETYPE_WINERYWORKDIARY = "WD"
NODETYPE_WINERYMONITORING = "WS"
NODETYPE_WINERYMONITORINGALGORITHMS = "WA"
NODETYPE_WINERYWINETRACKER = "WW"
NODETYPE_WINERYMATERIALS = "WT"
NODETYPE_WINERYDRYGOODS = "WY"
NODETYPE_REPORTS = "RP"
NODETYPE_VINEYARD = "VI"
NODETYPE_VINEYARDSHED = "VS"
NODETYPE_VINEYARDWORKDIARY = "VW"
NODETYPE_VINEYARDANALYSISREPORT = "VA"
NODETYPE_VINEYARDMACHINERY = "VM"
NODETYPE_VINEYARDMATERIALS = "VT"
NODETYPE_VINEYARDSTAFF = "VF"
NODETYPE_LABORATORY = "LB"
NODETYPE_STORAGE = "ST"

20.5 Worknote Status

WORKNOTESTATUS_CANCELLED = 999
WORKNOTESTATUS_WAITING = 20
WORKNOTESTATUS_NEEDSPARM = 30
WORKNOTESTATUS_READY = 40
WORKNOTESTATUS_ACTIVE = 50
'WORKNOTESTATUS_NEEDSRATIFICATION = 55
WORKNOTESTATUS_COMPLETE = 60

20.6 Vineyard Reading Types

VINEYARDREADING_RAINFALL = 1001
VINEYARDREADING_MAXTEMP = 1002
VINEYARDREADING_MINTEMP = 1003

20.7 Vineyard Block Reading Types

READING_BLOCK_STATUS = 100
READING_BLOCK_BPV = 1301
READING_BLOCK_ABW = 1302
READING_BLOCK_YIELDESTIMATE = 1303
READING_BLOCK_HARVESTDATEESTIMATE = 1501
READING_BLOCK_PH = 1502
READING_BLOCK_TA = 1503
READING_BLOCK_BAUME = 1504

20.8 Worknote Types

WNTYPE_GETYIELDFRUIT = 12
WNTYPE_TESTYIELDFRUIT = 13
WNTYPE_GETHARVESTDATEFRUIT = 14
WNTYPE_TESTHARVESTPARAMS = 15
WNTYPE_BARRELPRUNE = 16
WNTYPE_DROPFRUIT = 17
WNTYPE_POSTVINTAGEPRUNE = 18
WNTYPE_PETIOLETEST = 19
WNTYPE_SOILTEST = 20
WNTYPE_CLIMATEREADINGS = 21
WNTYPE_LAB_AMMONIATEST = 22
WNTYPE_LAB_MALICACID = 23
WNTYPE_LAB_PRIMARYAMINOACID = 24
WNTYPE_LAB_RESIDUALSUGAR = 25
WNTYPE_LAB_VOLATILEACIDITY = 26
WNTYPE_LAB_ALCOHOL = 27
WNTYPE_LAB_CLINITEST = 28
WNTYPE_LAB_CO2 = 29
WNTYPE_LAB_DISSOLVEDOXYGEN = 30
WNTYPE_LAB_NTU = 31
WNTYPE_LAB_PHTA = 32
WNTYPE_LAB_SO2 = 33

WNTYPE_LAB_TEMPBAUME = 34
WNTYPE_LAB_ADDITIONFININGTRIAL = 35
WNTYPE_LAB_BLENDINGTRIAL = 36
WNTYPE_LAB_TASTINGTRIAL = 38
WNTYPE_PROCESSFRUIT = 39
WNTYPE_ADDSO2 = 40
WNTYPE_ADDACID = 41
WNTYPE_ADDYEAST = 42
WNTYPE_TRANSFERRACKFILTER = 43
WNTYPE_PRESS = 44
WNTYPE_TANKCLEAN = 45
WNTYPE_TANKSANITISE = 46
WNTYPE_TANKRINSE = 47
WNTYPE_EQUIPCLEAN = 48
WNTYPE_EQUIPSANITISE = 49
WNTYPE_EQUIPRINSE = 50
WNTYPE_CHIPSADD = 51
WNTYPE_MOX = 52
WNTYPE_DIFFUSER = 53
WNTYPE_CHIPSREMOVE = 54
WNTYPE_HEATCOOLWINE = 56
WNTYPE_BOTTLEWINE = 57
WNTYPE_TRANSFERSKINSFROMPRESS = 58
WNTYPE_TRANSFERSKINSFROMVESSEL = 59
WNTYPE_SCUMRUN = 60
WNTYPE_ADDOTHER = 61
WNTYPE_ADDFINING = 62
WNTYPE_ADDMLF = 63
WNTYPE_LAB_TEMPERATURE = 64
WNTYPE_BATTONAGE = 65
WNTYPE_DRAINRETURN = 66
WNTYPE_DUMPWINE = 67
WNTYPE_LOADWINEFORREMOVAL = 68
WNTYPE_PLUNGE = 69
WNTYPE_PUMPOVER = 70
WNTYPE_SPARGE = 71
WNTYPE_VALVETOVALVE = 72
WNTYPE_LAB_COLDSTABLE = 73
WNTYPE_LAB_HEATSTABLE = 74
WNTYPE_CAPMANAGEMENT = 75
WNTYPE_LAB_FILTERABILITY = 76
WNTYPE_LAB_SENSORY = 77
WNTYPE_ADDNUTRIENT = 78
WNTYPE_LAB_SMOKETAINT = 79
WNTYPE_RESETVOLUME = 80
WNTYPE_LAB_LACCASE = 81
WNTYPE_LAB_CLINITEST_BSC = 82
WNTYPE_LAB_RESIDUALSUGAR_BSC = 83
WNTYPE_TOPPING = 84
WNTYPE_LAB_PHTA_BSC = 85
WNTYPE_LAB_MALICACID_BSC = 86

WNTYPE_GENERALWORKNOTE = 87

20.9 VINIFICATION DEFINITION PROCESSING

VINDEF_CHILLFRUITMUST = 0
VINDEF_RESTFRUITMUST = 1
VINDEF_DESTEMONLY = 2
VINDEF_CRUSHDESTEM = 3
VINDEF_WHOLEBUNCHPRESS = 4
VINDEF_RUNANALYSES = 5
VINDEF_ADDITIONS = 6
VINDEF_ADDATMUSTPUMP = 7
VINDEF_ADDATPRESS = 8
VINDEF_PUMPTOFERMENTOR = 9
VINDEF_PUMPTOCHILLERTANK = 10
VINDEF_ADDATFERMENTOR = 11

20.10 JOB TYPES

JOBTYPE_YIELDESTIMATE = 8
JOBTYPE_HARVESTDATEPARAMETERS = 9
JOBTYPE_BARRELPRUNE = 10
JOBTYPE_DROPFRUIT = 11
JOBTYPE_POSTVINTAGEPRUNE = 12
JOBTYPE_PETIOLETEST = 13
JOBTYPE_SOILTEST = 14
JOBTYPE_CLIMATEREADINGS = 15
JOBTYPE_RECEIVEANDPROCESSFRUIT = 16
JOBTYPE_WINETRANSFERRACKFILTER = 17
JOBTYPE_WINEADDITION = 18
JOBTYPE_WINEPRESS = 19
JOBTYPE_AMMONIATEST = 20
JOBTYPE_MALICACIDTEST = 21
JOBTYPE_PRIMARYAMINOACIDTEST = 22
JOBTYPE_RESIDUALSUGARTEST = 23
JOBTYPE_VOLATILEACIDITYTEST = 24
JOBTYPE_ALCOHOLTEST = 25
JOBTYPE_CLINITESTTEST = 26
JOBTYPE_CO2TEST = 27
JOBTYPE_DISSOLVEDOXYGENTEST = 28
JOBTYPE_NTUTEST = 29
JOBTYPE_PHTATEST = 30
JOBTYPE_SO2TEST = 31
JOBTYPE_TEMPBAUMETEST = 32
JOBTYPE_ADDITIONFININGTRIAL = 33
JOBTYPE_BLENDINGTRIAL = 34
JOBTYPE_TASTINGTRIAL = 36
JOBTYPE_VESSELCLEAN = 38
JOBTYPE_VESSELSANITISE = 39
JOBTYPE_VESSELRINSE = 40
JOBTYPE_EQUIPCLEAN = 41

JOBTYPE_EQUIPSANITISE = 42
JOBTYPE_EQUIPRINSE = 43
JOBTYPE_MOX = 44
JOBTYPE_DIFFUSER = 46
JOBTYPE_CHIPSREMOVE = 47
JOBTYPE_TEMPCONTROL = 48
JOBTYPE_HEATCOOLWINE = 49
JOBTYPE_BOTTLEWINE = 50
JOBTYPE_TRANSFERSKINSFROMPRESS = 51
JOBTYPE_TRANSFERSKINSFROMVESSEL = 52
JOBTYPE_SCUMRUN = 53
JOBTYPE_TEMPERATURE = 54
JOBTYPE_MULTIPLEWINELAB = 59
JOBTYPE_BATTONAGE = 60
JOBTYPE_DRAINRETURN = 61
JOBTYPE_DUMPWINE = 62
JOBTYPE_LOADWINE = 63
JOBTYPE_PLUNGE = 64
JOBTYPE_PUMPOVER = 65
JOBTYPE_SPARGE = 66
JOBTYPE_VALVETOVALVE = 67
JOBTYPE_COLDSTABILITY = 68
JOBTYPE_HEATSTABILITY = 69
JOBTYPE_CAPMANAGEMENT = 70
JOBTYPE_1DAYPOSTBOTTLING = 72
JOBTYPE_3DAYPOSTBOTTLING = 73
JOBTYPE_PRECOARSEFILTRATIONANALYSES = 74
JOBTYPE_PREBOTTLINGANALYSES = 75
JOBTYPE_FILTERABILITY = 76
JOBTYPE_RACKTESTADDRETURN = 78
JOBTYPE_RACKRETURN = 79
JOBTYPE_SMOKETAINT = 80
JOBTYPE_RESETVOLUME = 81
JOBTYPE_LACCASE = 82
JOBTYPE_CLINITESTTEST_BSC = 83
JOBTYPE_RESIDUALSUGARTEST_BSC = 84
JOBTYPE_TOPPING = 85
JOBTYPE_PHTATEST_BSC = 86
JOBTYPE_MALICACIDTEST_BSC = 87
JOBTYPE_GENERALJOB = 88

20.11 Wine Status

WINESTATUS_PROPOSED = 10
WINESTATUS_MUST = 20
WINESTATUS_FERMENT = 30
WINESTATUS_WINE = 40
WINESTATUS_BOTTLED = 100
WINESTATUS_LOADEDREMOVED = 110
WINESTATUS_DUMPED = 120

WINESTATUS_TRANSFERREDAWAY = 150

20.12 Ferment Status

FERMENTSTATUS_EARLY = 1
FERMENTSTATUS_MID = 2
FERMENTSTATUS_LATE = 4

20.13 Wine Sub Status Flags (Bitmap)

WINESUBSTATUS_MLF = 1
WINESUBSTATUS_ONLEES = 2
WINESUBSTATUS_ONBENTONITE = 4
WINESUBSTATUS_FILTERABLE = 8
WINESUBSTATUS_FINED = 16
WINESUBSTATUS_COLDSTABLE = 32
WINESUBSTATUS_HEATSTABLE = 64
WINESUBSTATUS_BOTTLEREADY = 256
WINESUBSTATUS_BLENDINGWINE = 128
WINESUBSTATUS_COMPLETEWINE = 512

20.14 Wine Reading Types

WINEREADINGTYPE_VESSEL = 1000
WINEREADINGTYPE_NAME = 1001
WINEREADINGTYPE_STATUS = 1002
WINEREADINGTYPE_SUBSTATUSFLAGS = 1003
WINEREADINGTYPE_FSO2PPM = 1010
WINEREADINGTYPE_TS02PPM = 1011
WINEREADINGTYPE_PH = 1012
WINEREADINGTYPE_TA = 1013
WINEREADINGTYPE_TEMP = 1014
WINEREADINGTYPE_BAUME = 1015
WINEREADINGTYPE_COLOUR = 1016
WINEREADINGTYPE_AMMONIA = 1017
WINEREADINGTYPE_MALICACID = 1018
WINEREADINGTYPE_PRIMARYAMINOACID = 1019
WINEREADINGTYPE_RESIDUALSUGAR = 1020
WINEREADINGTYPE_VOLATILEACIDITY = 1021
WINEREADINGTYPE_ALCOHOL = 1022
WINEREADINGTYPE_CLINI = 1023
WINEREADINGTYPE_CO2 = 1024
WINEREADINGTYPE_DISSOLVEDOXYGEN = 1025
WINEREADINGTYPE_NTU = 1026
WINEREADINGTYPE_GASBLANKET = 1027
WINEREADINGTYPE_HEATING = 1029
WINEREADINGTYPE_COOLING = 1030
WINEREADINGTYPE_MOX = 1031
'WINEREADINGTYPE_AUTOTEMP = 1032
WINEREADINGTYPE_SKINS = 1033

WINEREADINGTYPE_SCUM = 1034
WINEREADINGTYPE_RESIDUALGLUC = 1035
WINEREADINGTYPE_RESIDUALFRUC = 1036
WINEREADINGTYPE_COLDSTABLE = 1037
WINEREADINGTYPE_HEATSTABLE = 1038
WINEREADINGTYPE_FILTERABILITY = 1039
WINEREADINGTYPE_COMMENT = 1040
WINEREADINGTYPE_GUAIACOL = 1041
WINEREADINGTYPE_4METHYLGUAIACOL = 1042
WINEREADINGTYPE_LACCASE = 1043

20.15 Vessel Reading Types

VESSELREADING_WINE = 2001
VESSELREADING_BARRELSET = 2002
VESSELREADING_WINERY = 2003
VESSELREADING_LOCATION = 2004

20.16 Vessel Statuses

VESSELSTATUS_DISABLED = 100
VESSELSTATUS_DIRTY = 200
VESSELSTATUS_RINSED = 210
VESSELSTATUS_CLEANED = 215
VESSELSTATUS_SANITISED = 220
VESSELSTATUS_PENDING = 300
VESSELSTATUS_SKINS = 305
VESSELSTATUS_WHITEJUICE = 310
VESSELSTATUS_REDJUICE = 315
VESSELSTATUS_REDMUST = 320
VESSELSTATUS_REDFERMENT = 330
VESSELSTATUS_WHITEFERMENT = 340
VESSELSTATUS_REDSTORAGE = 350
VESSELSTATUS_WHITESTORAGE = 360
VESSELSTATUS_REDULLAGE = 370
VESSELSTATUS_WHITEULLAGE = 380
VESSELSTATUS_DELETED = 999

20.17 Equipment Types

EQUIPMENT_PRESS = 1
EQUIPMENT_RECEIVAL = 2
EQUIPMENT_FILTER = 3

20.18 Wine Transfer Methods

TRANSFERMETHOD_TRANSFER = 0
TRANSFERMETHOD_RACK = 1
TRANSFERMETHOD_FILTER = 2

20.19 Skin Status

SKINSTATUS_INPRESS = 1
SKINSTATUS_INWINE = 2
SKINSTATUS_VESSEL = 3
SKINSTATUS_DUMPED = 4

20.20 Additive Types

ADDTYPE_YEAST = 1
ADDTYPE_CHIP = 2
ADDTYPE_ACID = 3
ADDTYPE_SO2 = 4
ADDTYPE_YEASTNUTRIENT = 5
ADDTYPE_OTHER = 6
ADDTYPE_FINING = 7
ADDTYPE_MLF = 8

20.21 Dry Goods Types

DRYGOODS_CAPSULE = 3
DRYGOODS_CLOSURE = 4
DRYGOODS_DIVIDER = 5
DRYGOODS_BOTTLE = 1
DRYGOODS_PAD = 7
DRYGOODS_BOX = 2
DRYGOODS_LABEL = 6
DRYGOODS_PALLET = 8

20.22 Wine Group Types

WINEGROUPCAT_GENERAL = 1
WINEGROUPCAT_REGION = 2
WINEGROUPCAT_VARIETY = 3
WINEGROUPCAT_VINTAGE = 4