Designing and Implementing a Grid Application for Cumulative Agrichemical Residue Tracking using Third-Party Data Sources and Software Components

Elizabeth Post

Lincoln Ventures Ltd, P O Box 133, Lincoln, 7640, Canterbury, New Zealand
postea@lvl.co.nz

It is increasingly important to design software for reuse so future changes can be implemented as easily and cheaply as possible. It is also beneficial to use existing resources where they are available. This paper describes an agricultural application for cumulative agrichemical residue tracking designed and implemented to use grid technologies to access external third-party databases and software components. The application incorporates pre-existing software, and accesses meteorological, spatial, agrichemical and operational data, most of which are external databases owned and managed by third parties, and where the application must also be able to handle alternative data sources in each domain area.

Keywords: Software engineering, grid computing, design, interoperability
ACM Classification: D.2.12, D.2.11, D.2.10

1. INTRODUCTION

In recent decades there have been significant developments in both hardware, such as high-speed networks, and software, such as web services and grid computing (Foster and Kesselman, 1998; 2003; Berman, Fox and Hey, 2003). It is important to recognise the benefits of these new developments, and to design software to make use of them. Currently there are not many suitable databases and software components already grid-enabled and available for use. However, things are moving so rapidly that it is sensible to design future applications to take advantage of these technologies and incorporate external data and components as these become available over the grid.

This paper describes the design and implementation of a scientific application that can utilise external software components and external data, if these are available, by using grid technologies, with no changes to the application itself, and only minimal changes to its "interface" components. (In the context of this paper the term "interface" refers to the components between the application and external resources, not the use of interfaces to implement inheritance. Where the term interface is used in connection with a user interface, this is clearly stated.) CART (Cumulative Agrichemical Residue Tracking) (Zabkiewicz and Praat, 2004) is an agricultural application to monitor the cumulative effect of spray deposition resulting from farmers spraying their orchards or crops with agrichemicals such as pesticides. Over time these agrichemicals will dissipate, partially by chemical degradation, and partially by dispersion such as being washed away by water or by evaporation. Although various software components used in CART, such as for modelling spray drift, chemical dissipation and chemical degradation, have been in existence for some time on a small scale for

Manuscript received: 13 December 2006
Communicating Editor: Professor Bill Appelbe

individual farms and fields, we believe that this is the first time these components have been combined to create an application to monitor the cumulative effects over time of agrichemical residues from spray operations performed by a large number of different users in a region.

This application is used in primarily two ways. Firstly it is a multi-user distributed application, where farmers or contractors use a web user interface through an internet browser to enter data about the spray operations they perform. The application allows them to experiment with the proposed spray operation with the current weather conditions, to establish whether any spray from this operation will drift onto neighbouring fields. Once they confirm the spray operation the data for this operation is saved in a central database. Thus, all spray operations over time for participating farmers in a specific region will be saved in a central spray diary database. This provides bodies such as regional councils or grower cooperatives with accumulated historical data over the region. Such bodies can then use the application in its second mode, as a multi-user windows application over an intranet, to study the cumulative effects of spray deposits from multiple operations over time, calculated by totalling what is left from each deposit after degradation and dissipation over time, including all deposits over the period of interest. Users in this second category may want to monitor the cumulative deposits of particular chemicals, or perhaps run speculative scenarios to simulate seasons of spray operations for particular types of crops (such as apples or avocados) to estimate potential cumulative effects if for instance land use of some farms was changed from one crop to another, or to residential use, or perhaps farming became more intensive. There are many potential ways this application could be used for such research and monitoring. Also, there are potential economic benefits for farmers who can show records of their spray program and thus sell their products at premium prices because they adhered to a particular management strategy.

For the sake of convenience in the remaining discussion we will refer to the individual farmer or contractor users as "spray operator" users, and the users monitoring overall spray use, such as regional councils or sprayer cooperatives, as "regional" users.

When designing such an application it is important to recognise that there are a number of datasets it needs to access and that some of these may well be owned and managed by external parties. In some cases it may be possible to obtain copies of such data, but in others not. For instance, CART needs to use the following datasets as input:

- Spatial GIS (Geographical Information System) data including such information as the location and boundary data of the farms and their fields, and the land use of these areas, waterways etc
- Agrichemical data describing the chemical composition of spray products and including information such as toxicity, application rate, degradation rate etc
- Meteorological data, which includes both the current conditions at the time of each spray event, such as wind speed and direction, temperature, relative humidity etc, but also historical data over time, including rainfall, for estimating dissipation of deposited chemicals by evaporation and dispersion by water.

CART then also produces various datasets as output, including:

- Archive of all spray events in the region over time
- Records of deposits of chemicals from all spray events. (These could be recalculated from the spray event archive, but for performance reasons it is more efficient to store this information after it is initially calculated.)

The output data from CART will be stored centrally on a system owned by such an organisation as a regional council or growers' cooperative. However, all the input data may well need to be obtained from external sources, and the application needs to be designed to allow this.

In the past if one needed such external data it was normal to perhaps purchase a copy of a database and load it locally. However, this means that one has to be aware that it is essential to obtain regular updates to ensure the local database is up to date. For data that changes infrequently this is a reasonable solution. However, for frequently changing data this can become a problem. The development of grid technologies now make it possible to access such an external database automatically over the internet. This means that the responsibility of ensuring the data is up to date belongs to the external source owning the data, not the user, and the user can be assured that the data is always up to date. There are still various ways in which this data can be accessed using grid technologies, and we will discuss these alternatives later in this paper.

Another important consideration for this application is its incorporation of existing software for various components such as the drift model that estimates how spray will drift according to various factors, including the wind speed and direction and composition of the spray material. Where suitable software components are available for use it is desirable to be able to incorporate them in another application rather than having to duplicate effort and waste resources by rewriting them (Post, 2002). Such components, if available could be dynamically linked libraries (DLLs) or executables that could be “wrapped” in code for use by the application. In many ways this is preferable for performance reasons. However, in some cases it may not be easy to obtain access to such a component. With the advent of grid computing and web services it will become increasingly common to find a web service that already exists that will perform a specific task that an application needs. In such cases it is necessary that the application can incorporate use of such external services.

The rest of this paper describes the issues considered when designing and implementing CART for potential future use incorporating external data and software components via “the grid”. It then explains what further work will need to be done to complete the work of grid-enabling CART. Finally it summarises the benefits of designing applications on such a way that they are easily extended to take advantage of grid technologies.

2. DESIGN CONSIDERATIONS

This section looks at the issues considered when designing CART so that it could make use of external data and software components via “the grid”.

In designing CART it was important to recognise that although it was possible to identify the generic types of data that would be used (e.g. GIS, agrichemical, meteorological) it was impossible to predict exactly what format each database would be in when CART is used by a number of different regional users. For instance, not only might different users of CART be using databases in completely different formats from databases used by other CART users (especially the GIS’s, where for instance some regional users may use ESRI GIS’s, and other Manifold, or any of a large variety of GIS’s), but also sometimes CART may need to dynamically choose from alternative data sources, particularly with meteorological data. Even within the same country different meteorological data is needed for farms in different regions, and this may be obtained from different suppliers. Also CART is designed for potential international use, and thus will need to be able to access and use meteorological data from a wide variety of sources, most likely in many different formats.

In the cases of both the databases and the software it is important to recognise that usually they will be owned and managed by external independent entities and the CART developers have no control over the format or content of the databases or how they are accessed, or the functional interface to the software component. The developers of these external resources have every right to change these as they will without consulting actual or potential users, although most responsible developers will do their best to use good initial design to minimize impact of future changes. Although the CART developers could write CART to use their own local version of the various databases and import the external data into them this is not always practical. For one thing that requires the CART administrators at each institution using CART to constantly ensure they have up to date data. For another, some datasets, such as GIS data, are just too large, and there may be proprietary and other limitations prohibiting their export to other GIS's. Thus it is preferable to design CART to be able to use external data and software components exactly as they are provided by their developers, without any transformations being necessary, and minimizing any impact on CART if there should be any changes in the external resources.

To achieve this we borrowed the same principle of abstraction that is used when creating grid middleware, of providing an interface between a resource (the external database or software component) and a consumer of that resource (the CART application). This meant the CART application could be written in such a way that it was totally independent of the interactions with the external resources, and thus robust enough to cope with change in external resources with minimal, possibly no, impact on the actual CART application.

Thus there is an interface between the CART application and each external data source, as illustrated in Figure 1. For these external data sources CART only needs to access the data in read-only mode. Changes to any data sources are managed external to CART. This is primarily because in some cases it will be impossible for CART to have write-access to databases owned by others. In any cases CART will not be needing to make changes to meteorological or agrichemical data. The only potential data changes CART may want to use would relate to GIS data, such as land use or property boundaries, and it is quite reasonable to require that such changes would be administered by the GIS maintenance team.

Similarly there is an interface between the CART application and each external software component as in Figure 2. In this case there will most likely be some data input to the software component and the result data returned.

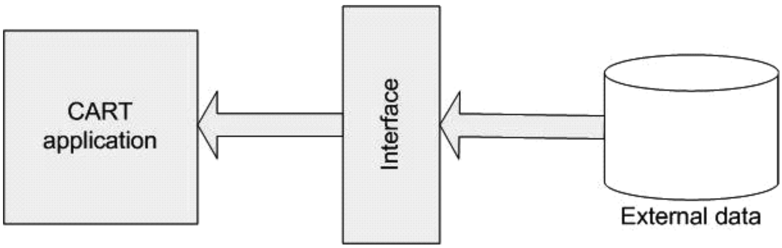

Figure 1

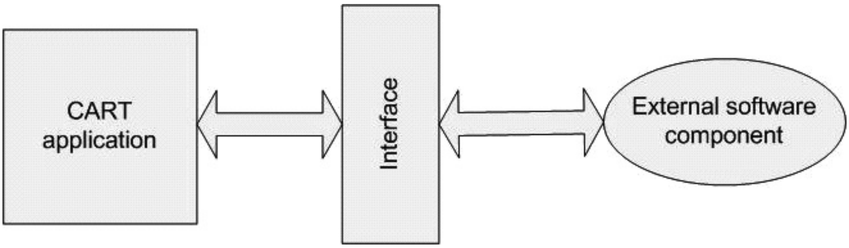

Figure 2

Thus any changes to external resources may require moderation or extension of an interface between that external resource and the CART application, or even creation of a new interface, but unless there have been radical change no changes should be necessary to the CART application itself.

The rest of this section discusses some of the issues that had to be considered when designing CART:

- to use external data, and
- to use external software components.

We will consider each of these separately.

2.1. Designing Data Interfaces

This section describes some of the issues that needed to be considered with regard to each type of data, further illustrating why using an interface was an extensible solution.

2.1.1 Spatial (GIS) Data

Spatial data such as used by CART is commonly needed for many reasons, and is usually set up and managed by organisations such as government or regional bodies. It will include much of the geographical information and will be kept reasonably up to date. However, various information may well change over time, such as ownership of properties, boundaries of properties and fields, and land use of areas. Thus a user of GIS data either needs direct access to an up to date GIS or needs to obtain revised data from time to time. GIS's often contain huge amounts of data and it is not often practical for other users other than the owner to obtain copies. Also, it is expensive to maintain such data sets, and some or all of the information may well be proprietary and not available for external users. Thus grid technologies provide a mechanism where external users may obtain controlled access to up to date data, as specified by the GIS owner. In the case of CART most likely the primary user will be a regional body or growers' cooperative that will already have their own GIS data. In this case CART will probably run on the same intranet and have direct access to their own GIS. However, it is possible that for instance a primary CART user, such as a growers' cooperative, may make a business arrangement with a GIS provider, such as a regional council, to have controlled access to relevant GIS data. In this case CART may need to use grid technologies to control access to this GIS data over the internet.

2.1.2 Agrichemical Data

In the case of the agrichemical data, most of the products used by farmers contain chemicals which are regulated, perhaps because of their toxicity, and because of their potential cumulative effects. Each country tends to have a regulatory body which manages registration of such agrichemical products, and issues registration numbers for each product. When each product is registered certain information about it, such as its chemical composition, toxicity etc, is provided and entered in the database. Much of this information could be obtained from the product suppliers themselves. However, since they have to register the products it makes sense to obtain the information from a central regulator's database, rather than from the various databases in different formats from different suppliers. For one thing it would be difficult to keep up with all new and discontinued products from all suppliers, including new suppliers. However, if all the information is recorded centrally in a regulator's database this would be just one database that needs to be accessed. It is also likely that such a regulator's database will continue to store information about discontinued products, which could be useful for CART in studying historical spray events using now discontinued products. While it is possible for us to create our own agrichemical database for use by CART it would be extremely difficult to ensure this was always up to date. Thus it makes much more sense for CART to be able to directly access such a central national regulator's database. We have also established that there are many other potential users of such a database. Unfortunately at this stage in New Zealand no such database exists in a form where CART could use it over the internet as a web service. However, CART has been designed so that if and when such a database does become available for grid access, it can access it without alternation to CART.

2.1.3 Meteorological Data

With meteorological data this is the area where the most progress has been made with regard to making such data available over the internet. There are many sources of meteorological data, and some are already available over the internet, such as by web services (Laurenson, Kiura and Ninomiya, 2002). However, with regard to CART we need to recognise that there are different ways in which such weather data may be used. Initially, when recording a spray event, a farmer enters the current data for the field being sprayed at that time. While it is possible that there may be a nearby weather station, perhaps owned by a national weather bureau, which is making its data available by a web service, it is unlikely that in most cases there will be one close enough. Thus the farmer is most likely to obtain the weather data either from an on-farm weather station, or by personal observation. CART is designed to allow data to be obtained directly from either a local on-farm weather station, or via a web service from a nearby weather station, or by direct manual entry by the spray operator.

The second way in which CART uses meteorological data is when calculating degradation and dissipation over time. In this case CART needs access to historical weather data over a period of time. For these calculations CART could either use recorded data from an on-farm weather station, or data from a source such as a national weather bureau from a nearby weather station, which could be obtained by a web service.

2.2 Designing Interfaces to Software Components

CART has been designed to take advantage of existing scientific knowledge concerning spray drift and chemical degradation and dissipation and it is not the primary intention of CART to develop and validate new models. Therefore, wherever possible, we prefer to incorporate existing implementations of models as software components in CART, rather than re-implementing them. Also the scientists planning to use CART for research wish to be able to compare model results with field results, and perhaps evaluate several alternate models to determine which gives the best results. Currently CART includes only one incorporated model, a component that models spray deposit and drift, AgDisp (AGDISP, 2003). However, there are plans to add several more components regarding the effect of shelter belts on spray drift, chemical degradation and chemical dissipation, as well as more drift models.

To enable incorporation of models that are already available as executable code "wrappers" are written to control access to these components. Such wrappers provide a standard interface to CART, and to incorporate alternate models for the same component it is only necessary to provide a wrapper for the new component that has the same interface to CART. This design also allows incorporation of original models that we may code ourselves.

We also use a modular design that allows CART to interface to a larger granularity component that may include several smaller granularity sub-components. For instance, an important part of

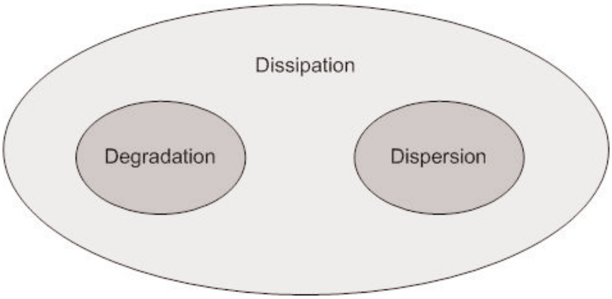

Figure 3

CART is calculating how chemicals dissipate over time. However, there are several factors involved in dissipation, including that chemicals degrade over time, but also that they evaporate or are dispersed by water, such as rainfall. This design is illustrated in Figure 3.

One of the advantages of this design is that it allows CART to interface to a number of alternate dissipation modules without "knowing" what is inside them. The module output is the same (total current concentration of chemicals) regardless of how many separate components are used to calculate it. Thus, in early stages of development the dissipation module may be a very simple implementation without any sub-components at all, but later a more complex dissipation module may be developed with a number of sub-components, and there may also be alternate versions of those sub-components. Also, these sub-components could be local, "wrapped" executables, or remote resources accessed via web services. However, CART can interact equally well with either dissipation module because each will have the same interface, and it is actually irrelevant to CART whether or not the module has any sub-components and what these are.

As alternate external software components and data sources become available it will be possible for each user to configure CART as he or she wishes, for instance to experiment with using alternate models for calculating drift.

2.3 Making Appropriate Choices when to use Grid Computing

Just because technologies exist that allow us to use remote external resources across the internet it is not always appropriate that this should be done. It is important to carefully evaluate each situation and decide when it is appropriate to use resources via the grid and when not. Grid computing is well-suited to loosely coupled non-interactive applications that run for a long time. Using grid resources in an interactive application can result in performance delays. Thus for instance it is preferable to be able to use the component locally rather than across the grid. Even then it is still possible to design an application to use a web service to automatically detect if the software has changed and if necessary download a new version for local use. However, in some cases local use may not be possible for various reasons, such as specialized software needing particular resources that are not necessarily available elsewhere, or because of licensing issues. In such cases it will be necessary to use the resource as a web service.

Similarly it will not always be appropriate for performance reasons to directly query a database over the internet unless there is no other way to get the data. For instance, in most cases a CART implementation will be hosted by a regional body that already has GIS data. Thus CART will have direct access to a local GIS. (CART's design allowing it to interface with many different GIS formats is still necessary as different users may well use different GISs.)

On the other hand the agrichemical database would ideally be hosted as a national resource and it would not usually be appropriate to have a local copy. In this case CART will likely use grid computing technologies to query this database. Only a small amount of data is needed for each spray event. If performance for an interactive application is too poor when accessing this database over the grid then it may be possible to arrange to download a copy of this database for local use. However, grid computing technologies could still be used to automatically detect if the database is changed, and if so to automatically download a new copy.

The meteorological data is yet another case. It may be appropriate to access the current weather conditions for a spray event from a web service from an appropriate weather station. However, when computing the degradation and dissipation of chemicals over time it is necessary to use a long series of historical data. In most cases it would be poor design for an application to directly access a long series of historical data one item at a time over the internet. Thus for this case CART would automatically download the appropriate data series and then use it locally. However, this could all be done using grid and web services. There is already a considerable amount of meteorological data available in such a form. One example that provides such data through a web service is MetBroker (Laurenson *et al*, 2002).

Thus when designing an application that may use grid resources it is essential to carefully evaluate each situation and determine the most appropriate way to implement it.

2.4 Overall Design of CART

Figure 4 shows the overall design of CART that resulted from the considerations described above. The interface between the user interfaces and CART also allows for future internationalisation.

In all cases if further alternate software or data resources are added all that should be necessary is to write new interfaces between CART and these resources.

3. IMPLEMENTATION

This section briefly describes how CART has been implemented using the design principles identified above.

3.1 Choice of C#.NET

The main CART application has been implemented using C#.NET, even though some external software components may be in other languages and may even only be available as executable code or dynamically linked library (DLL). C#.NET (Microsoft Visual C# Development Center) was chosen because it is a generic portable language and will run on other operating systems besides Microsoft in the event that for regional users with large amounts of data and a large number of users we may need to run the main CART application on a larger server running Linux in order to achieve the performance required for an interactive application. C# is also a clean, well-designed and implemented object-oriented language which also has a number of other useful features, such as the use of "interfaces" which make it easier to implement our interfaces between the application and external resources. (Interfaces in this programming context are a useful programming technique, not exactly the same as the interfaces we have used in CART, but which have been implemented using this technique).

3.2 Use of Model-View-Controller Design Pattern

To ensure that it will in future be possible to port CART to a different operating system and avoid possible potential issues with the user interface we have separated all user interfaces from the

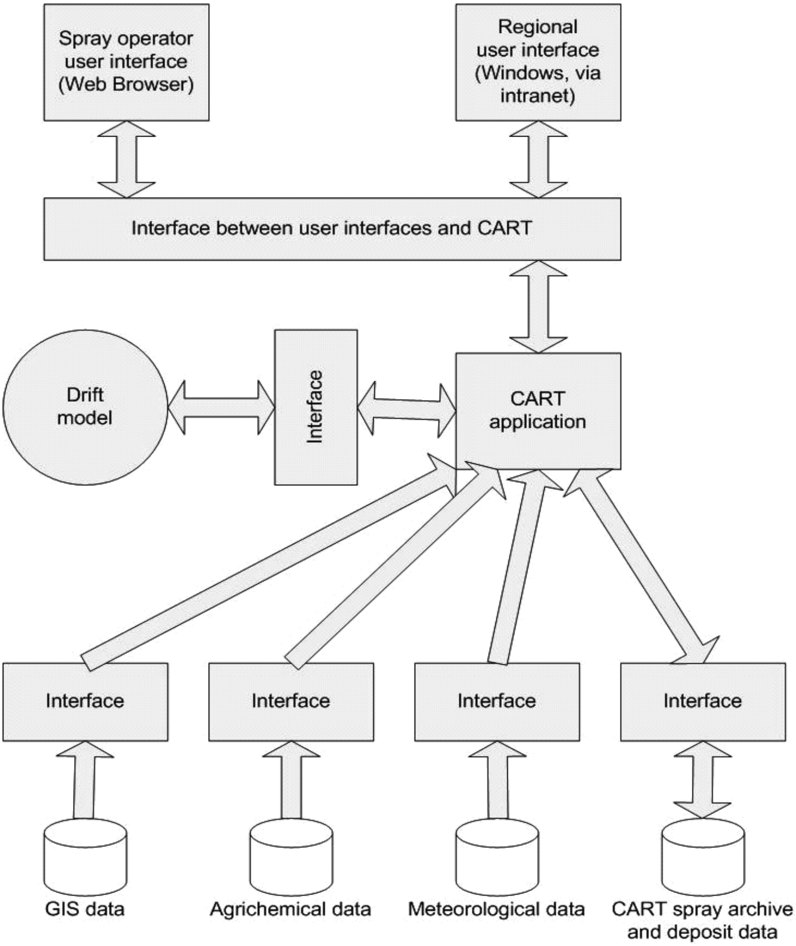

Figure 4

application by using the Model-View-Controller design pattern, as described in Gamma, Helm, Johnson and Vlissides (1995). This design also allows for future internationalization of CART by providing user interfaces in different languages, without having to change CART. It also allows for further user interfaces to be developed, which may use some functionality common with other user interfaces.

The current Web spray operator user interface has been implemented using ASP.NET, although the Model-View-Controller design of CART would allow for other Web technologies to be used if desired. The current Windows regional user interface was implemented as a C#.NET Windows Project.

3.3 Implementation of CART Interfaces

Each interface to a data source and to a software component has been implemented by writing one (or possibly more) C# dynamically linked library (DLL). In most cases all users will have the DLL's for all the available software components and will be able to select which to use. For interfaces to databases a DLL implementation makes it very easy to substitute different DLLs providing interfaces to different data sources for different users. All other DLLs that comprise CART will be the same for all users.

We have made use of an existing library that will read (and write) GIS shapefiles, ShapeLib (Shapefile C Library, 2004), which is wrapped in a C# wrapper. Most GIS's either use shapefiles directly for storing spatial data or can export to shapefiles, so this library can be used to interface with most GISs.

The interfaces to the current local databases have been implemented using ADO.NET. However, as we move to directly accessing external databases via the grid these interfaces will interact with the external databases by using web services, but the CART application side of the interfaces will remain the same.

3.4 Privacy and Security

Currently all users are identified when logging in to CART to control access regarding which users may read and write which data. As CART begins to use external data sources and software components with web services, access to these resources will automatically be negotiated by CART according to the terms in which CART is allowed to use these resources.

4. CONCLUSION

By using interfaces between the application and external data as described above it is possible for CART to interface to any number of different alternate data sources, including local data. For initial development of CART local data has been used but the design will allow for accessing grid-enabled resources as and when these become available.

This project demonstrates that it is possible to design and implement a scientific application so that it can be executed by using alternate data sources, where the data may be in different formats and either locally available or obtained over the internet with web services. Also this project illustrates how different users may use some data from different sources, but may share other central databases with other users. In addition this project shows how it would be possible to use different implementations of component models over the internet as web services if these were available.

When CART was designed and first implemented the web services to provide the necessary data and component models were not yet available. However, we believe that in the future it will be increasingly important to design and implement scientific applications so they can take advantage of grid computing technologies to incorporate both software components running elsewhere, and also various alternate data sources in different formats, without the application needing to be rewritten to adapt to such changes.

Conversely, it also becomes important for providers of both potential software components and external databases to make proper use of grid computing technologies to enable access to such software and databases by future external users.

Over the years there has been considerable wastage of resources as scientists duplicate what has been done elsewhere, such as writing software, but not well enough to be reusable by others (Post, 2003). Similarly, frequently data is not properly organised and is not available to external users, who then need to recreate similar databases at further cost. It is to be hoped that increasing use of grid computing technologies will encourage researchers to ensure that both software is developed professionally and databases designed and implemented professionally, so that these resources are reusable by external users and can be made available to others via web services.

5. FUTURE WORK

The prototype implementation of this application was completed by mid-2005. At this stage the application has been completed as described above, with interfaces to all data sources, and wrappers round scientific models used within the application, a spray operator Web user interface and a regional user Windows interface. It is now being demonstrated to potential users to get feedback. The application can continue to be used as it is at this stage, using local data and incorporated component models. However, as appropriate grid-enabled data and software resources become available over the internet this application can easily be extended to use these external resources. Some areas of work that may be undertaken in this regard are as follows.

- Incorporation of further software components. Some of these will be external components and will be incorporated by "wrapping" them with an interface CART can interact with. However, as suitable components become available over internet it is likely that although the interface to CART will stay the same, the "wrapper" will incorporate grid technologies to access these components with web services.
- Allow CART to directly access meteorological data over internet. There are already numerous resources that provide meteorological data, such as MetBroker (Laurenson *et al*, 2002) and those provided by such organisations as the National Institute for Water and Atmospheric Research (http://www.niwa.cri.nz) in New Zealand. We will initially identify one or two of these resources and write interfaces for CART to either access them directly as needed, or to automatically download an appropriate dataset as needed.
- We have identified that there are a large number of potential users of a grid-enabled national agrichemical database. Thus we are beginning negotiations with the regulatory authority to establish such a resource in New Zealand that provides a central electronically accessible resource incorporating all relevant data so that it can be accessed both by human users via web pages and also interactively by software with web services. Once such a resource has been established we will modify the relevant interface in CART to directly access this data via a web service, rather than locally, so we can be sure of accessing up to date data.
- If CART begins to be used internationally we will have to provide a new interface to access the national agrichemical database for each country rather than the New Zealand agrichemical database.
- As CART begins to be used by various organisations we will write new interfaces for CART to interact with their GISs. In most cases these will be local but CART has been designed to allow remote access as well if this becomes necessary. However, the large amounts of data in a GIS may also mean that some form of caching needs to be implemented to alleviate poor performance as far as possible.
- International use of CART will also require international versions of the user interfaces. The design of CART also allows for any number of interfaces so that different users could be accessing the same installation of CART using different languages.
- Other software to record spray operations in spray diaries already exists. It may be necessary to write an interface that allows spray diaries from other sources to be imported into CART.

ACKNOWLEDGEMENTS

This project is funded by the New Zealand Foundation for Research, Science and Technology. We also thank the USDA Forest Service for providing us with the AgDisp (AgDisp, 2003) DLL and allowing us to use it in this project. Thank you to the others involved in this project: John-Paul Praat, Jerzy A. Zabkiewicz, John Maber, Phil Dewar, Wayne Schou, Darryl Gillgren and Matthew Laurenson.

REFERENCES

AGDISP (2003): USDA Forest Service Spray Modeling Software, version 8.13, Harold W. Thistle, USDA Forest Service, 180 Canfield Street, Morgantown, WV 26505.

BERMAN, F., FOX, G. and HEY, A. (2003): *Grid Computing: Making the Global Infrastructure a Reality*, John Wiley and Sons Ltd.

FOSTER, I and KESSELMAN, C. (1998): *The Grid: Blueprint for a New Computing Infrastructure*, First edition, Morgan-Kaufmann.

FOSTER, I and KESSELMAN, C. (2003) *The Grid: Blueprint for a New Computing Infrastructure*, Second edition, Morgan-Kaufmann.

GAMMA, E., HELM, R., JOHNSON, R. and VLISSIDES, J. (1995): *Design Patterns*, Boston, MA: Addison-Wesley.

LAURENSON, M., KIURA, T. and NINOMIYA, S. (2002): Providing agricultural models with mediated access to heterogeneous weather databases, *Applied Engineering Agric*. 18: 617–625, and MetBroker web page at http://www.agmodel.net/, Accessed 24 November 2006.

MICROSOFT VISUAL C# DEVELOPMENT CENTER, (2006): http://msdn.microsoft.com/vcsharp/, Accessed 24 November 2006.

POST, E. (2002): Adventures with portability (2002): *Third LCI International Conference on Linux Clusters: The HPC Revolution 2002*, Florida, USA.

POST, E. (2003): Designing and implementing computer simulation models for portability and reuse, *MODSIM 2003*, Townsville, Australia, http://www.mssanz.org.au/modsim03/Media/Articles/Vol%204%20Articles/1751-1756.pdf, Accessed 24 November 2006.

SHAPEFILE C LIBRARY (2004): Frank Warmerdam. .NET wrapper provided by David Gancarz. (dgancarz@cfl.rr.com or david.gancarz@cityoforlando.net). http://lists.maptools.org/pipermail/shapelib/2004-January/000002.html. Accessed 24 November 2006.

ZABKIEWICZ, J. A. and PRAAT, J-P. (2004): CART – Cumulative Agrichemical Residue Tracking, *International Conference on Environmental Friendly Spray Application Techniques*, Warsaw, Poland.

BIOGRAPHICAL NOTES

Elizabeth Post

Elizabeth Post is a senior computer scientist at Lincoln Ventures Ltd, where she does both research and consultancy. She collaborates with scientists to deliver complex computing solutions to advance their research. In particular she is most interested in using advanced computing technologies such as grid computing and high performance computing to enable scientists and clients to solve problems they have not been able to do using smaller scale computing. She was formerly a senior lecturer in applied computing at Lincoln University, New Zealand. Before that she was a lecturer and a research officer at the University of Cape Town, South Africa.