1 Problems, Limitations and Improvements

1.1 Introduction

Chapter four discussed some of the options that are available to a developer implementing a metadata engine. Chapter five outlined the options chosen in developing the prototype metadata engine as well as the modifications that were undertaken on the Isite source code in order to implement the engine. The purpose of this chapter is to discuss the many limitations of the developed prototype. The chapter also discusses the possible improvements that could be incorporated into a commercial metadata engine. The limitations that will be discussed include:

- Limited Querying
- Limited Access
- Cryptic Search Criteria
- Hard Coding of Source File Locations
- Points and Lines Display Only
- Accuracy and Integration
- Charging
- Privacy
- Speed
- Metadata Divergence
- Firewalls
- Data Currency

1.2 Limited Querying

One of the major limitations with the developed prototype is that the metadata engine has a very limited querying ability. Once the results of the user's query on the ANZLIC compliant metadata records has been returned there is essentially no other query that can be accomplished. From the results of the users query on the metadata records the user is able to choose which of the datasets they wish to display as well as the central point of interest for each of those datasets. Other than to zoom in and out, pan and clear the data, the spatial data viewer cannot perform any other GIS functionality on the datasets.

More work needs to be undertaken on the spatial data viewer to give it the ability to parse more detailed user queries involving the datasets that are displayed by the data viewer. The two aspects that will have to be investigated in order to achieve this extra functionality are:

1) The addition of extra user interface functionality will have to be added to the spatial data viewer to allow the user to actually enter the parameters for the query that they wish to submit. This is simply a matter of adding a few extra buttons, menu items and forms that will allow for this information to be accepted by the spatial data viewer.

2) Since it is possible that more than one of the datasets being viewed is required to satisfy the users query, a method of parsing the query into subqueries will have to be established. Each of the subqueries would be able to be solved by one, and only one, of the datasets that the user is currently viewing. The subqueries, once derived from the original query, would be sent to the database that could solve them. The results of the subqueries would be returned from the remote servers, where the individual datasets resided, and merged into one. This merged result should be the answer to the users original query and would be displayed by the spatial data viewer. To achieve all this more access metadata will have to be returned to the central server from the remote servers. The extra access metadata

will have more information about the components of each of the datasets, hence allowing the queries to be parsed effectively.

1.3 Limited Access

Another of the limitations that the prototype has is that it only allows for one user to operate it at any one time. This limitation occurs since the access metadata that is returned as a result of a successful search, is written to a file called "Directory_Andrew.txt". The results for every single search overwrite this file every single time. Hence if one user conducts a query the results are written to the "Directory_Andrew.txt" file. If a second user then comes along and uses the metadata engine before the first user has finished, the results of the first users query in the "Directory_Andrew.txt" file will be overwritten with the results of the second users query.

This is a relatively straight forward problem to fix. Two things need to be done:

1) Adjust the Spatial Data Viewer code to accept, as an argument, the name of the file that the access metadata has been written to. Currently this information is hard coded into the Spatial Data Viewer and hence the system has no choice but to use the same file to store the access metadata every time it is used.

2) Add some extra code to the metadata engine that creates a file, with a random name, to store the access metadata in. The name of the file is then passed as an argument to the spatial data viewer. This should assure that two users should not be overwriting the results of each others queries.

1.4 Cryptic Search Criteria

One of the biggest limitations that the spatial data viewer has is the cryptic nature of the search criteria for each of the datasets. At the present time the search criteria that has to be entered by the user is the number that is used to identify a point or line in the dataset. The identification number that a point or line has bears no relationship to its

location in space. To use the metadata engine effectively the user must know an identification number for a point or line in the area that they are interested in. If they do not know such an identification number then it is almost impossible to use the metadata engine efficiently.

To increase the usability of the system the search criteria should be linked with a street address dataset. By doing this the user would be able to enter a street address for the search criteria, rather than the point or line identifier. The metadata engine would then look up the appropriate street address in the street address dataset, get its coordinates and then display all the data for a predefined distance around it. By using the street address approach it would also be possible to do away with needing search criteria for each of the datasets, a single search criteria could be used for all datasets.

1.5 Hard Coding of Source File Locations

A limitation that has resulted from a quick fix programming solution is that the name and location of the files "directory.txt" and "Directory_Andrew.txt" have been hard coded into the modified Isite code. This results in the application looking for them at the same hard coded location for every installation. This makes the installation of the metadata engine and remote Isite servers more cumbersome as the directory structure has to be set up to exact specifications with everything placed in the correctly named directory.

The best way to eliminate this limitation is to read the location and file names of the two aforementioned files from the "zserver.ini" file when it is read to initialise zserver. All that would be required for this to be achieved would be to add the parameters concerning the "directory.txt" and "Directory_Andrew.txt" to the "zserver.ini" file and then to modify the metadata engines source code so that it reads these parameters. This would have the effect that the installer would simply have to edit the parameters concerning the location and name of the "directory.txt" and "Directory_Andrew.txt" files when they edit the rest of the parameters in the "zserver.ini" file to enable zserver to operate.

1.6 Points and Lines Display Only

One of the biggest limitations with the spatial data viewer is that it only displays points and lines. At this stage the spatial data viewer has no provision for the display of polygons or attributes. This limited display functionality has obvious drawbacks when more complex spatial data than the simple points and lines in the two SQL datasets are to be viewed.

The solution to this limitation of the system is simply to add more functionality by undertaking some extra coding. The extra coding required to achieve the added functionality would be quite complex. Due to the time constraints and prototype nature of this research project it was not deemed necessary to undertake them. The prototype that was developed is simply aimed at being a proof of concept, rather than a commercial product that is able to do everything that a user could possibly need.

1.7 Accuracy and Integration

In any application that integrates, or merges, two or more datasets together, either into the one dataset or to display the two or more datasets concurrently, the inaccuracy of one or more of the datasets can cause many integration problems. Integration problems occur where common features in two or more datasets do not have the same coordinate values as each other. This has the effect that when the two datasets are displayed concurrently by a spatial data viewer they will not appear in the same location, when in fact they should.

There is only one real solution to this problem, which is to improve the individual accuracy of each of the datasets. Government organisations are constantly in the process of undertaking that exact task on the fundamental datasets that make up a typical SDI. As more accurate means of measurement and better total quality management (TQM) procedures are adopted by the majority of the survey profession the integration problem will start to diminish. This is one problem associated with the

prototype that can not be solved quickly, but it is one that is likely to fix itself over time.

1.8 Charging

Other than certain government organisations it is highly unlikely that companies are going to place their spatial data into a system such as a metadata engine if there is no profit to be made. At the current time there is no means in the prototype to actually charge users for the data that they use/view. The charging issue is one that has to be addressed as the quality and quantity of the spatial data that is likely to be put on the web is likely to be poor if there is no money to be made.

There are three main issues that will have to be addressed to solve the charging problem. These are:

1) At what point do people pay? Do users pay the minute they access the system? Or, do users pay the minute they access specific data from specific datasets? Do users pay to simply search the metadata records?

2) How do people pay? Will the metadata engine be credit card based so that anyone with a credit card can use the metadata engine by simply supplying their credit card number over the web? Or, will the metadata engine be based on an accounts system where the user will first have to get an account and a password, so they can be billed, to access the system?

3) How will the metadata engine divide the profits amongst the spatial data suppliers? Will the metadata engine split the profits evenly amongst the spatial data suppliers? Or, will the metadata engine split the profits according to whose data gets accessed the most?

The first point is an issue that has to be resolved by the developer, however if the user has to pay to early in the process it is likely that potential repeat customers will be scared off if they are constantly getting charged for data that is of no use to them. A possible solution is to have access to the metadata records and sample datasets free so

that the user can see what they are buying. Once the user accesses the full datasets they are then charged for the amount of data they view.

The second point is really a judgement call for the developer, however it is probably easiest to develop it using an accounting system approach. However it is likely to be used by more people if it is a credit card based system due to the ease of paying for the data. Having to obtain an account is likely to turn one off users away from using the system.

It is unlikely that the data suppliers would be happy to split the profits from the system evenly, which therefore counts out the easier of the two options in the third point. This means that the metadata data engine would have to have some code added to it that allows it to keep track of exactly which datasets have been accessed by each of the users and exactly how many times. Another point to consider here is the possibility that different data suppliers may charge different rates for accessing their datasets. This will have the effect that in the returned access metadata there will have to be a parameter that details the rate at which the customer is to be charged.

As the prototype is developed as a proof of concept, not as a commercial system, it was not necessary for all this added complexity to be incorporated into the prototype to allow for the charging of customers. Software developers that wish to implement a commercial system should have no trouble taking the above factors into account when developing their system.

1.9 Privacy

One of the big advantages that occurs when datasets are merged together is that their combined value is often greater than the sum of the values of the individual datasets. As mentioned earlier the reason for this is the fact that when two datasets are combined they not only show their own information, they also tend to show new information that results from the integration. Often this new information will be in the form of patterns and trends. The problem with this is that some people may not wish others to know this new information about them. Everyone wants some degree of

privacy and this type of system that is able to create, sometimes unforeseen, new information about people and places could make some people feel very uncomfortable.

There is no quick and easy fix for this problem. Despite the fact that much of the information that would be created from the process of integrating many datasets together would be of extreme value for planning improved infrastructure, services and business opportunities, it is also apparent that in the wrong hands the information could be used to exploit others. Legislation is the only answer to the problem of privacy, however if legislation is taken too far another problem, "censorship" arises which many people are equally, if not more, sensitive about.

1.10 Speed

One of the major disadvantages that occurs when retrieving spatial data from many datasets located across a network is that it is often quite slow due to the lack of network speed. If a database was set up on the users local system that contained all the spatial data that they were ever likely to need, it will be faster than a distributed system. There are two reasons why this approach is not as practical as a distributed approach; firstly the shear size that the users computer would have to be, and secondly the problem of data currency and updates.

There are several ways that the speed of the system can be improved. The first method for improving the speed of the system is to apply programming practices that improve the efficiency of the metadata engine as a whole. This reduces the time that the metadata engine actually takes to formulate the results for the users query. It will however not reduce the time that the results actually spend travelling across the network.

The second method is to actually improve the standard of the communication lines between each of the servers. This reduces the time that the results actually spend travelling across the network. This is something that the developers of such a system are unable to undertake as it is up to the government and telecommunications industry

to improve the overall standard of the telecommunications network. At the present time there is a project named VicOne that is being undertaken by the Victorian government that will link all government departments via a broadband network (AAPT 1998). Once this network is completed the speed of information retrieval from government organisations will be greatly enhanced, however the speed of information retrieval from private organisations that are not on the broadband network will be considerably slower.

1.11 Metadata Divergence

One of the points that should be mentioned now is that during the course of this research a new metadata standard has been created that deals with the return of access metadata to the metadata engine. The problem with creating your own standards is that it becomes increasingly difficult to find software that will be able to use data in your standard the further that you diverge from the accepted standards. This is something that Australia should take note of with their metadata guidelines. Even though there are a reasonable number of software developers in the Asia/Pacific that are capable of developing applications using standards develop in Australia, there are many times more in the USA and Europe. If standards in Australia diverge too far from those that are being used elsewhere in the world, then Australia will be limiting the range of software that that is available to it.

There is no real solution to this problem other than to keep up to date with what the rest of the world is doing in terms of the standards that they are setting for their data. If we keep abreast of the developments overseas it should be relatively straight forward to make sure that the standards that are in place locally do not diverge too far from those established throughout the rest of the world.

1.12 Firewalls

Firewalls are very inconvenient when developing a metadata engine prototype such as the one developed in this research project. Firewalls are used to prevent unauthorised

access to the internal networks of organisations from outsiders. This means that any dataset that is in the hands of a custodian that uses a firewall to protect their network, cannot be accessed by users outside the network. This really defeats the purpose of having such a metadata engine that has as one of its main objectives the effective and efficient dissemination of spatial data.

The best solution to this problem is to have a mirror copy of the dataset outside the firewall that can be accessed by users from outside the organisation. Daily, weekly, monthly, or even yearly the mirror dataset can be updated so that it reflects the master dataset behind the firewall. This of course means that outside users will be unable update the dataset unless there was some sort of mechanism to report changes to the mirror dataset to the master dataset where they could be entered. Obviously only certain types of outside users would have these types of privileges, whereas the rest of the community would only be able to view and query the dataset.

1.13 Data Currency

One of the problems that arises when a mirror site is introduced into the system to counter the problems of firewalls is the fact that the data within it may not be current. For many applications a few errors due to the data being slightly out of date are not a problem, however it is vital for the user to know exactly when the data that they are using was up to date. Organisations can make allowance for data that is out of date, assuming that they know that the data is out of date and by how much. The last thing an organisation needs is to make a decision on data that is out of date and yet they assumed it was accurate at the time they accessed it.

The ANZLIC metadata guidelines have a property in page 0 that is supposed to represent the currency of the dataset. This property could be used by the data custodian to hold the date that the mirror dataset was last updated. The best way to achieve this would be to automate this process using a program that updated the mirror dataset and at the same time modified the metadata record so as though it held the correct date for the time that the dataset was last updated.

1.14 Chapter Summary

This chapter outlined the limitations that the developed prototype has along with some of the improvements that could be made to eliminate or improve these limitations. The prototypes limitations could be classified into policy and coding limitations. Policy limitations could in general only be solved by government, whereas the coding limitations could be solved by a developer who had more time to spend developing a commercial system.

The policy limitations included:

1) Accuracy and integration. This limitation exists as very few, if any, datasets are completely accurate. This means that when the datasets are shown concurrently by the spatial data viewer they do not fit together properly.

2) Charging. The prototype has no ability to charge the users for the data that they access. The method of charging is both a choice for the designers and the data custodians.

3) Privacy. One of the big advantages of developing a metadata engine is that you can create new information by viewing datasets concurrently that have not been viewed together before. This however does create a problem when the wrong people create this information as they can find out information about people that those people may not wish them to know.

4) Speed. Network speed is a problem in a system that utilises a network to access datasets located across it. Broadband fibre optic cables that are in the process of being installed are a possible solution.

5) Metadata divergence. Metadata standards in Australia cannot diverge to far away from those that are adopted in the USA and Europe as software may be difficult to come by.

The coding limitations include:

1) Limited querying. The spatial data viewer is very limited in the GIS functionality that it is able to perform. Currently the viewer is only able to zoom in and out, pan, and fit the spatial data on the screen. Extra code needs to be added to the spatial data viewer to give it extra querying capability.

2) Limited access. Due to the fact that the access metadata is always written to the same file the metadata engine can only handle one user at a time. This limitation is able to be solved by generating random file names for each of the users.

3) Cryptic search criteria. The search criteria for each of the datasets is the ID for a particular line or point in the dataset. This ID has no relationship to its location in space and hence it is very difficult to search the datasets. The best solution to this limitation is to link a street address dataset to the spatial data viewer so that the user can search by street address.

4) Hard coding of source file locations. The names and locations of the files that contain the access metadata are hard coded into the source code. Solving this limitation is simply a matter of placing the files names and locations in the "zserver.ini" file.

5) Points and lines display only. The spatial data viewer has the ability to display points and lines only, it does not have the ability to display polygons and attribute data. Extra code needs to be added to the spatial data viewer to give it extra display capability.

6) Firewalls. Many organisations have the datasets located behind firewalls in order to protect them from outsiders. This means that a metadata engine cannot gain access to the dataset unless a mirror copy exist outside the firewall.

7) Data currency. A result of having a mirror copy of a dataset that is located behind a firewall is that the currency of the dataset that the users are accessing has to be known. The metadata record that corresponds to that dataset must be kept up to date to allow users to find out the currency of the dataset.

1.