

Catchment Planning Tool

Mugodo, J.^{1,5*}, **S. Linke**^{2,5}, **S. Arene**^{3,5}, **N. Marsh**^{4,5} and **P. Lea**^{1,5}

¹*Civil and Environmental Engineering, University of Melbourne, VIC 3010*

²*The Ecology Centre, University of Queensland, QLD 4072*

³*Australian Rivers Institute, Griffith University, QLD 4111*

⁴*CSIRO Land and Water, Indooroopilly, QLD 4068*

⁵*eWater CRC, ACT 2601, Australia*

*Email: jmugodo@unimelb.edu.au

Abstract: Catchment managers need to prioritize areas for conservation and restoration because they have limited resources and compete with other land uses (Margules and Pressey, 2000). The Catchment Planning Tool (CPT) arose from the growing need by natural resource managers to identify and prioritize areas for conservation and restoration activities at the catchment scale (Marsh et al., 2007). The problem faced by efforts to develop environmental software is how to create a program that addresses all the needs of different audiences. Generally, attempts are made to develop a stand-alone tool that has every conceivable functionality. However, a more pragmatic approach is to develop a generic program that can accommodate the various issues faced by different catchment managers. Our solution was to develop the CPT (Figure 1), a Geographic Information System (GIS) program that utilizes The Invisible Modelling Environment (TIME) code base (Rahman et al., 2003) to display and interrogate spatial data. A plug-in system was built into the design of the CPT to allow it to communicate with any models that provide some prerequisite identification details. In the CPT, a plug-in is a program that provides extra functionality to the CPT, or that can access the CPT's functionality via the CPT's plug-in application programming interface (API). The CPT was designed to be able to use a combination of plug-ins to assess in-stream biological and environmental attributes and to predict the effects of environmental and management changes on rivers at a catchment scale. This ability of the CPT to easily plug-in programs makes it easy to extend the functionality of the CPT, giving it the flexibility required to address the different challenges that face a wide range of catchment managers.

The basic CPT configuration includes the core CPT application, that displays GIS data and hosts plug-ins, a plug-in manager to organize other plug-ins, and a GIS Toolbox plug-in. The GIS Toolbox plug-in provides a collection of GIS functionality that can be used for both terrestrial and riverine applications. Catchment delineation from digital elevation models (DEM) and the routing of river properties (e.g. taxa composition and stream constituents) through a river system are examples of the functionality that is specific for river studies. TIME code implemented based on Prosser et al. (2001) was used to provide the catchment delineation functionality. The delineation of catchments enables the creation of a river network composed of nodes and links. By knowing the from-node and the to-node of a link, properties of streams can be routed up and down the network.

For specific environmental problems, further plug-ins can be added to the basic CPT configuration to extend its functionality. MARXAN is a software program that is used for optimizing reserve design (Ball and Possingham, 2000; Possingham et al., 2000). A MARXAN plug-in was developed for the CPT. This paper demonstrates some of the functionality available within the basic CPT configuration and uses the MARXAN plug-in as an example of how the basic CPT functionality can be extended. The example shows how resource managers could employ the CPT plug-in architecture to seamlessly combine various models (MARXAN in this instance) and GIS functionality under one computer program.

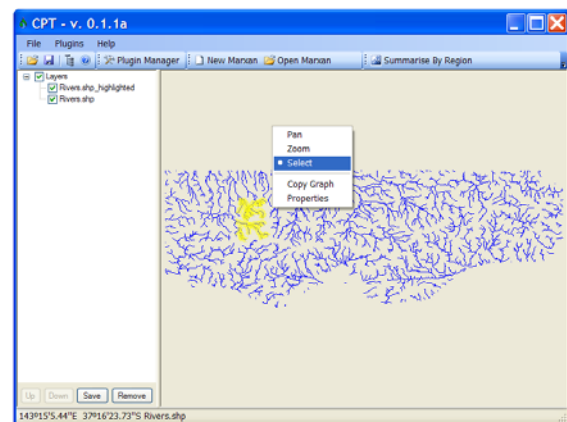


Figure 1. The Catchment Planning Tool (CPT) is a GIS interface for displaying and analyzing spatial data. Its functionality can be extended by adding plug-ins that are organized under the Plugins menu.

Keywords: Catchment asset prioritization, restoration, conservation, MARXAN, decision-making, GIS

1. INTRODUCTION

Catchment managers need to identify and prioritize areas for conservation and restoration because they have limited resources and compete with other land uses (Margules and Pressey, 2000). The Catchment Planning Tool (CPT) arose from the growing need by natural resource managers to prioritize areas for conservation and restoration activities at the catchment and local scales, and from the recognition that different natural resource managers have different issues to resolve (Marsh *et al.*, 2007). Indeed, a problem that plagues efforts to develop environmental software is how to create a program that addresses all the needs of different audiences. A sample of the wide range of challenges that various river managers require solutions for include:

1. identifying reaches and wetlands of high conservation value
2. determining priorities and 'best value' for protection and restoration activities at catchment scale and at more localized scales
3. making the best use of Environmental Water Allocations as part of overall strategies to restore flow regimes, within a broad water management context
4. predicting responses in aquatic ecosystems and water quality resulting from landuse changes and water management activities
5. assessing the cumulative impacts of changes to the catchment, including restoration activities such as reforestation, and potential changes in climatic factors
6. assessing the relative roles of major ecosystem disturbances such as drought, floods, fire
7. devising water quality strategies and plans that consider the source and behaviour (spatial and temporal) of contaminants.

Given the plethora of solutions required by managers, it is difficult to offer an initial software program that addresses all issues for everyone.

A pragmatic approach to this problem of competing demands is the plug-in concept that is used in computing to:

- enable third-party developers to create and add capabilities that extend an application
- add features in the future that were not included intentionally or unintentionally in the main program
- reduce the size of an application

A plug-in is a computer program that interacts with a host application (CPT, in this case) to provide additional, generally more specific, functions. In the CPT context, a plug-in is any program that provides extra functionality to the CPT or that can access the CPT's functionality via the CPT's plug-in application programming interface (API). A plug-in system allows the flexibility to use any program that provides some prerequisite identification details. The CPT was designed to be able to use a combination of plug-ins to assess in-stream biological and environmental attributes and to predict the effects of environmental and management changes on rivers at a catchment or local scale.

The CPT is part of the suite of software products for river management initiated by the eWater Cooperative Research Centre (CRC) (<http://www.ewatercrc.com.au/technologies/>, accessed May 2009). With the aim of making the CPT relevant to current catchment management efforts, partner organizations of the eWater Cooperative Research Centre (CRC) were asked to provide their catchment frameworks for improvement and incorporation into the CPT as plug-ins. Three examples of customized plug-ins from collaborations with the Department of Sustainability and Environment (DSE), CSIRO Land and Water, and the University of Queensland, respectively are:

1. a plug-in for the Aquatic Value Identification And Risk Assessment (AVIRA) database for evaluating the status of river planning units based on environmental, economic, social and risk metrics (Heron, 2009; Peters, 2009)
2. a GIS Toolbox plug-in that pools and builds on the GIS functionality in the TIME code base to provide a collection of commonly used spatial functions that can be employed in both terrestrial and riverine applications (Prosser *et al.*, 2001; Rahman *et al.*, 2003)
3. a plug-in to provide the CPT with functionality for prioritizing locations for conservation using a pre-existing tool called MARXAN (Ball and Possingham, 2000; Possingham *et al.*, 2000). MARXAN has been used internationally in a number of terrestrial (Rondinini *et al.*, 2006), marine (Fraschetti *et al.*, 2009), and fresh water (Hermoso *et al.*, in press) projects.

The ability of the CPT to easily plug-in any model based on a few identification details allows it to provide an interface between other eWater CRC products such as the Ecological Response Modelling (ERM) framework (Marsh *et al.*, 2007) and the Water and Contaminant Analysis and Simulation Tool (WaterCAST) (Jordan *et al.*, 2007). Such a software solution would enable managers to address challenges 3 to 7 in the above list of challenges. In addition, more specialized third party programs, addressing some of the seven challenges listed above could be developed independently and then brought together in the CPT as plug-ins to provide a holistic and well integrated catchment planning solution.

The aim of this project was to develop a versatile software product that can meet the various needs of catchment managers without the use of commercial GIS software. The MARXAN plug-in and the AVIRA plug-in both enable the CPT to address points 1 and 2 in the above list of challenges. However, currently, MARXAN is more widely used and AVIRA relies more on expert opinion in setting conservation and restoration priorities. Therefore, this paper focused on the MARXAN plug-in and the GIS Toolbox plug-in as examples of how the functionality of the core CPT of opening, displaying and saving GIS data can be extended through its plug-in architecture.

2. METHODS

2.1. The CPT plug-in architecture

The plug-in architecture plays the central role in the CPT. Figure 2 illustrates the CPT's plug-in structure and how the CPT and the plug-ins interact to get the services they require. The CPT is essentially a GIS enabled Graphical User Interface (GUI). It takes GIS input and displays it in the map area of the GUI. We cater for .arc raster files and ESRI shape files, a vector format with data stored in fields (ESRI, 2006), because these are common GIS data formats. The plug-in system was implemented so that the CPT could be extensible by dint of pooling its basic functionality with the specialized services of different plug-ins as show in Figure 2. The basic CPT configuration, represented by the thick solid boxes in Figure 2, includes the core CPT application and two plug-ins. The former displays GIS data and hosts plug-ins. The two plug-ins in the basic CPT configuration are a plug-in manager to organize other plug-ins, and a GIS Toolbox plug-in. The GIS Toolbox builds on and bundles together some of the GIS utilities within the TIME code base. It provides a collection of GIS functionality that can be used for both terrestrial and riverine applications. Catchment delineation from digital elevation models (DEM) and routing of river properties (e.g. taxa composition and stream constituents) through a river system are examples of the functionality that is included specifically for river based projects. TIME code implemented based on Prosser *et al.* (2001) was used to provide the catchment delineation functionality. The delineation of catchments enables the creation of a river network composed of nodes and links. By knowing the from-node and the to-node of a link, properties of streams can be routed up and down the network. Most importantly, Figure 2 serves as an example of how the functionality of the basic CPT configuration can be extended by a new plug-in, the MARXAN plug-in in this case (dotted box). This enables different managers to have a different set of more specialized plug-ins according to the challenges they need to tackle in their catchments.

The CPT was written in C#, one of the languages supported by the Microsoft .NET Framework (Schildt, 2006). The plug-in host is a key part of the CPT's plug-in system. The CPT uses the plug-in host to find and load plug-ins from a specified folder. The CPT can then call upon a combination of services from any of the loaded plug-ins. Individual plug-ins can use the plug-in host to access the services of other plug-ins in the same manner as the core CPT program. Behind the GUI of the CPT the Virtual Map List Manager (VLM) forms the backbone of the GIS data management (Figure 2). It is used to store loaded maps in a tree structure, it has methods to rearrange the stacking order of the stored maps, to remove maps, and to add or remove data fields from shape file maps. As part of the mandatory host services, the plug-in host uses a reference to a VMLM to provide plug-ins with access to the map data used by the CPT. This arrangement facilitates two-way communication between the CPT and the plug-ins. The plug-in architecture was designed to allow two-way communication in order to maximize the flexibility of the CPT and plug-ins assemblage.

Before an external program can work as a CPT plug-in, it has to provide information which is part of the plug-in interface mandatory services (Figure 2). This simple plug-in interface was put in place so that meeting its information requirements would not be onerous. The mandatory services of a plug-in are used to cite the authors of the plug-in and to enable the plug-in host to search for the plug-in by name. Another mandatory service is to determine whether a plug-in should be accessible or not at a given time (i.e. locked or unlocked). The plug-in interface has initialize and dispose routines that perform some user defined tasks when the plug-in is loaded and when the plug-in is unloaded, respectively. Lastly, the plug-in interface

provides access to the plug-in's tool bar and menu as part of its mandatory services. The additional services in the plug-in represent the extra functionality it introduces to the CPT. This new functionality may be called directly by the CPT, or it may be accessed by the user via the tool bar and menu items of the plug-in.

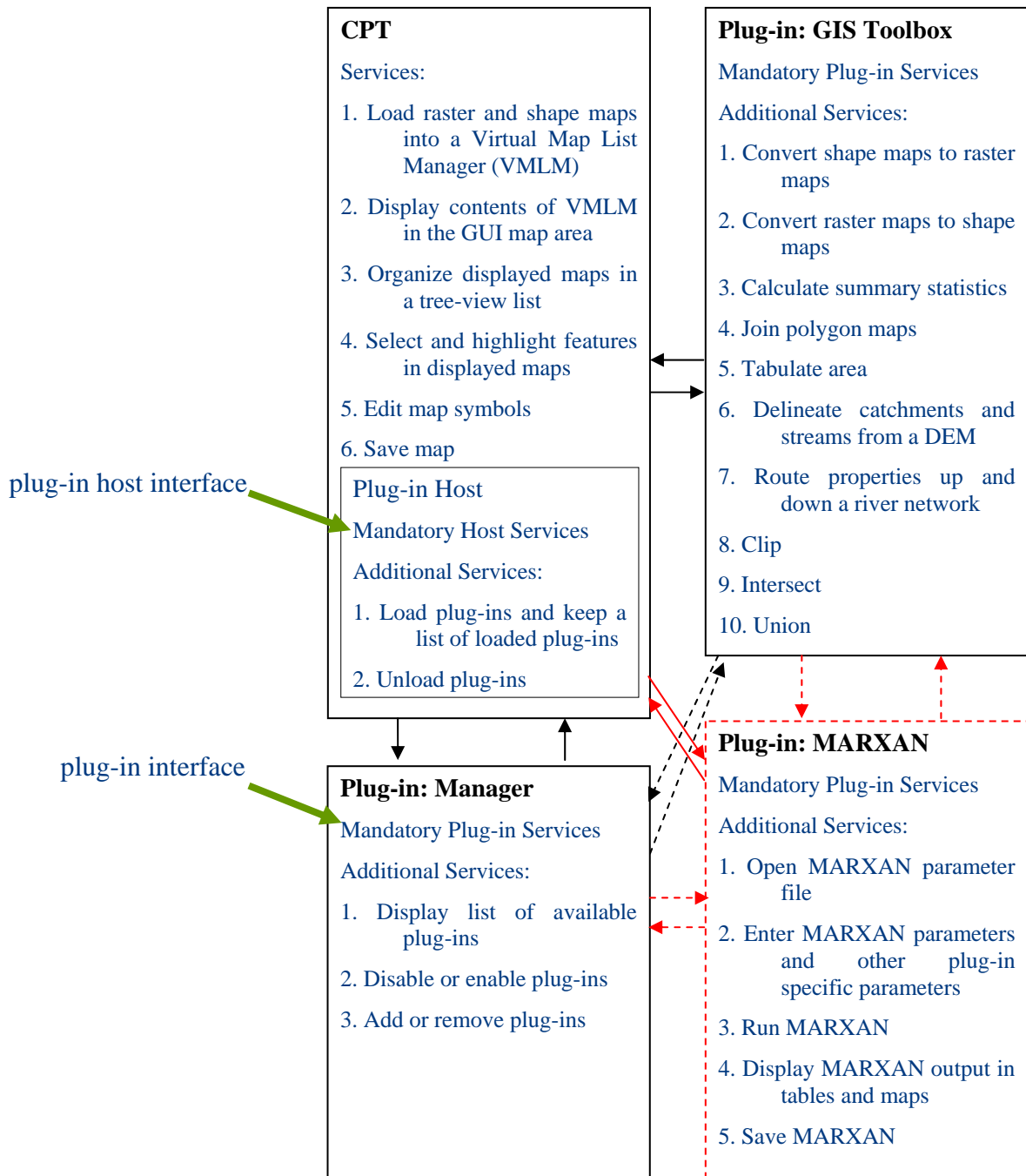


Figure 2. The interaction of the CPT and some plug-ins through the plug-in host interface and the plug-in interface. There is two-way communication between the core CPT and the plug-ins (thin solid arrows) and also two-way communication between the plug-ins (thin dotted arrows) making it easy for the different components to request services from other components.

2.2. Installing the basic CPT and extending it with specialized plug-ins

The CPT program will be available for download from <http://www.toolkit.net.au/tools/cpt> from July 2009. The CPT download will be in the form of a setup.exe file that allows easy installation of the three

components of the basic CPT configuration (thick solid boxes in Figure 2). Before the actual installation of the CPT, the *setup.exe* file checks that the computer has the appropriate versions of the Microsoft .Net framework software. If not, the user is directed to the Microsoft website where the file is downloaded. After the prerequisite Microsoft software is installed, the *setup.exe* will install the core CPT application in a user defined folder and the Manager plug-in and the GIS Toolbox plug-in are installed in a subfolder called plugins. Plug-ins such as the MARXAN plug-in that extend the basic configuration will be available as individual downloads. They can be downloaded directly to the plugins subfolder, or the CPT plug-in installer can be used to copy them there from an alternative download location. Once a plug-in is installed, the plug-in is displayed under the Plugins menu of the CPT's GUI where the user can directly access its functionality via the submenu items of the plug-in. The installation of a plug-in also displays its toolbar in the CPT GUI.

2.3. Using the CPT to prioritize conservation areas

We used invertebrate taxa distribution data in Victoria from Linke *et al.* (2008), a combination of the basic CPT configuration and a plug-in for MARXAN to demonstrate how plug-ins can allow the CPT to perform specialized tasks. The values of the MARXAN parameters used in the process were entered via the CPT (Figure 3). The main parameters that must be set are those for the data input text files (a file for taxa names, a file of planning unit unique identifiers and or names, and a file of the distribution of the taxa among the planning units). In this case, the data from Linke *et al.* (2008) were already in text format. When data are not available in text format the MARXAN plug-in has a wizard that can use the GIS Toolbox plug-in to perform the necessary pre-processing operations (from the list shown in Figure 2) to create the MARXAN input data files from GIS layers. The other option when data are not already in a text format suitable for MARXAN is for the MARXAN plug-in to extract the data from a database. Some of the other more advanced MARXAN parameters that are preset, but can be modified by the user, include the algorithm for selecting catchment planning units, the number of simulations and the number of iterations per simulation. We used simulated annealing (Ball and Possingham, 2000), 1,000 simulations and 10,000 iterations per simulation. Further details on MARXAN parameters are given in Ball and Possingham (2000).

The MARXAN plug-in was used to select catchment planning units for a near optimum reserve network based on their complementarity in terms of their overall invertebrate taxa representation. This meant that catchment planning units were not simply selected because they were individually rich in taxa. Instead, catchment planning units were included in the reserve design because they complemented other catchment planning units that had already been selected so that when pooled together, the catchment planning units met (or got close to) the pre-defined conservation targets of representing all the taxa, while reserving areas that have low cost tradeoffs and appropriate spatial configurations (Possingham *et al.*, 2000; Rondinini *et al.*, 2006). For example, in riverine reserves, configurations that have protected upstream subcatchments provide a higher level of protection because they do not leave down stream conservation sites exposed to upstream disturbances (Hermoso *et al.*, in press; Linke *et al.*, 2008). Only the simulation that resulted in the lowest cost out of the 1000 simulations was taken as the best solution (Ball and Possingham, 2000). All the simulations were useful, however, because they were used to calculate the irreplaceability of each planning unit which is determined by the number of times out of the 1000 simulations it was selected as a member of a candidate reserve design (Possingham *et al.*, 2000).

The integration of the MARXAN functionality into the CPT required interactions between the core CPT, the GIS Toolbox and the MARXAN plug-in. For instance, one of the new parameters introduced in our MARXAN plug-in was the location of the file of the planning units map so that output for both the best solution and the irreplaceability analysis could be appended to that base-line map and displayed. To achieve this, the MARXAN plug-in used the MARXAN program to determine the best solution and to calculate the irreplaceability of the planning units. These MARXAN outputs are in the form of text files. The MARXAN plug-in then had to incorporate these text files into the planning units map using the joining operations from the GIS Toolbox. The core CPT could then display the output of the analysis as the best solution map and the irreplaceability map (Figure 4).

2.4. Results

We developed a catchment planning tool that can be used to facilitate decision-making in the management of aquatic ecosystems (Figure 3). The core of the CPT is a GIS user interface that can communicate with user selected plug-ins for specialized tasks. The basic CPT is a standalone application with a look and feel similar to some popular, but expensive commercial GIS products. The basic CPT provides the GIS functionality listed in Figure 2. The GIS Toolbox plug-in includes some river specific functionality such as delineation of

catchments and streams based on a DEM and routing stream properties through the network. The routing is based on knowing the from-node and the to-node of the links that make up the stream system.

We successfully used the MARXAN plug-in to demonstrate how the CPT's architecture allows managers to seamlessly add their specialized models to the CPT as plug-ins. The MARXAN plug-in has a wizard tool for creating the MARXAN input data text files from GIS layers or from a database, if data for MARXAN are not already available in text format. The MARXAN plug-in allows users to select the planning unit map so that the catchments selected for the reserve (Figure 4a) and the irreplaceability of the catchments (Figure 4b) can be displayed on the map. With the MARXAN plug-in, the user can view MARXAN data files in a table and make edits within the CPT.

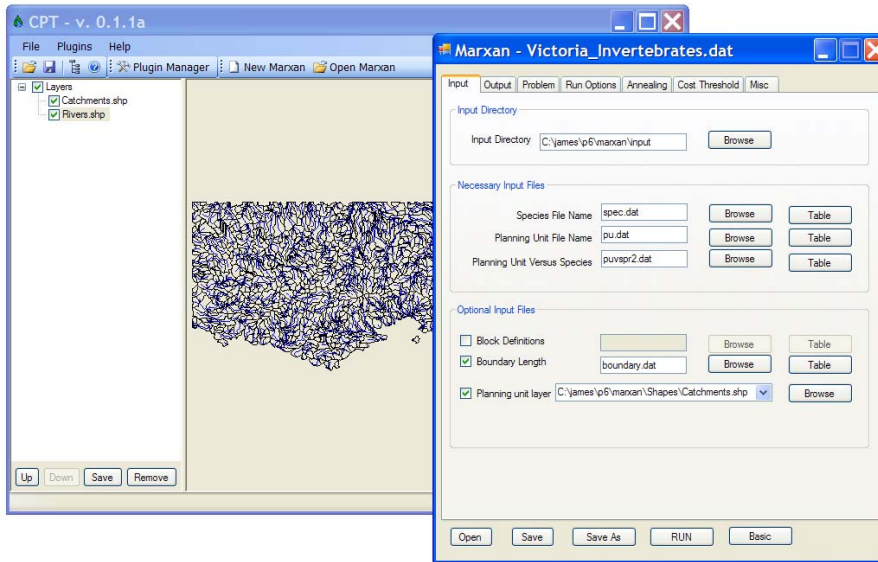


Figure 3. The CPT showing its MARXAN plug-in that is used to set MARXAN parameters. The MARXAN plug-in has additional parameter that can be used to identify a map of the catchment planning units so that MARXAN output can be mapped automatically. The CPT is running a reserve selection analysis for determining the optimal selection of conservation sites for protecting the invertebrate taxa in Victoria based on data from Linke *et al.* (2008).

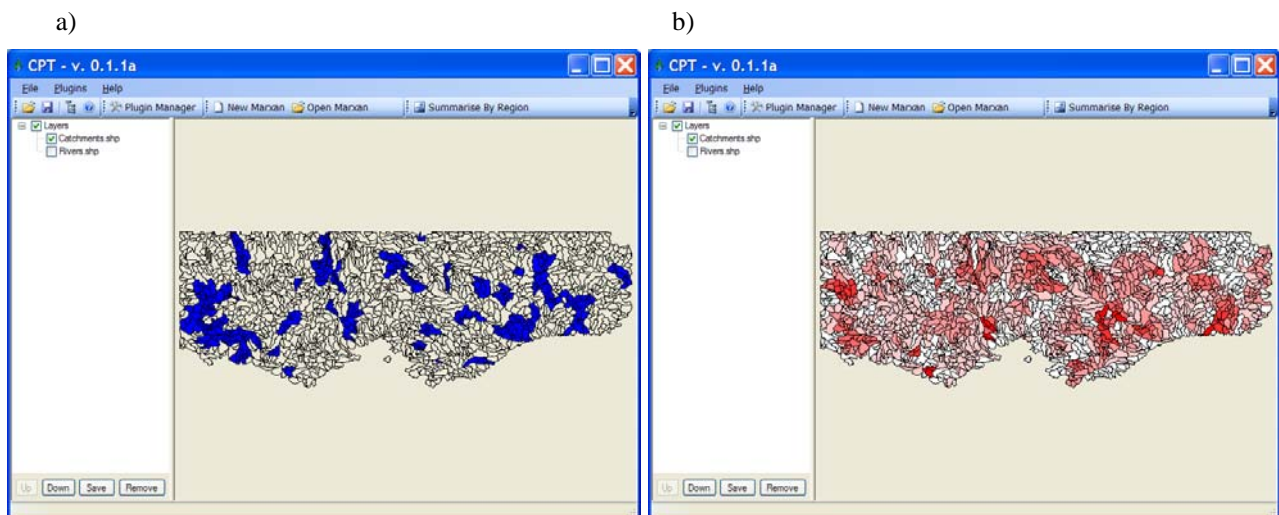


Figure 4. a) Catchments in the most cost effective reserve design, according to the objective function defined for MARXAN, for protecting invertebrate taxa in Victoria. **b)** The irreplaceability map showing how hard it would be to replace given catchments in alternative reserve designs; the darkest sites are the most difficult to replace. The analyses are based on invertebrate data from Linke *et al.* (2008).

3. DISCUSSION AND CONCLUSIONS

Using the architecture shown in Figure 2, the implementation of the CPT program was able to achieve the flexibility required for facilitating decision-making in the management of aquatic ecosystems. Adding the MARXAN plug-in to the basic CPT configuration demonstrated this flexibility and the convenience of seamlessness that comes with the CPT's plug-in approach. The MARXAN plug-in extended the capabilities of the basic CPT configuration by integrating the non-GIS enabled, but widely used MARXAN program. The integration provided the MARXAN program with GIS capabilities for pre-processing input GIS data and for displaying, manipulating and analyzing MARXAN output. The CPT's approach of gradually building up the collection of specialized applications via plug-ins is particularly suited to software that has to address diverse needs such as the challenges faced by different catchment managers.

Ongoing work includes adding more functions to the GIS Toolbox. The open nature of the CPT system will allow users to write and add their own GIS tools as new plug-ins. In addition to the MARXAN plug-in, a river assets database plug-in has been written for the Department of Sustainability and Environment (DSE) in Victoria. There are two immediate future projects. The first will be based in New South Wales and will involve building a biodiversity forecasting tool based on environmental variables. The second project involves interfacing the constituent modeling capacities of WaterCAST with the Ecological Resources Model framework via the CPT plug-in system as part of the South East Queensland rivers management system.

ACKNOWLEDGMENTS

The authors would like to thank eWater CRC for supporting the Catchment Planning Tool (CPT) and the contributors of the TIME code base that was instrumental in the implementation of our solution.

REFERENCES

- Ball, I. R., and Possingham, H. P. (2000). MARXAN (V1.8.2): Marine Reserve Design Using Spatially Explicit Annealing: http://www.uq.edu.au/marxan/docs/marxan_manual_1_8_2.pdf.
- ESRI. (2006). GIS Dictionary: ESRI Support Centre. <http://support.esri.com/index.cfm?fa=knowledgebase.gisDictionary.search&searchTerm=shapefile>.
- Fraschetti, S., D'Ambrosio, P., Micheli, F., Pizzolante, F., Bussotti, S., and Terlizzi, A. (2009). Design of marine protected areas in a human dominated seascape. *Marine Ecology Progress Series*, 375, 13-24.
- Hermoso, V., Linke, S., Prenda, J., and Possingham, H. P. (in press). Addressing longitudinal connectivity in freshwater systematic conservation planning. *Freshwater Biology*.
- Heron, S. (2009). *AVIRA Risk Assessment Process Report*. Melbourne, Victoria: Department of Sustainability and Environment (DSE).
- Jordan, P., Argent, R., and Nathan, R. (2007). *New generation models for catchment modelling: developments in the eWater CRC*. Paper presented at the 5th Australian Stream Management Conference. Australian rivers: making a difference, Charles Sturt University, Thurgoona, New South Wales.
- Linke, S., Norris, R. H., and Pressey, R. L. (2008). Irreplaceability of river networks: towards catchment-based conservation planning. *Journal of Applied Ecology*, 45, 1486-1495.
- Margules, C. R., and Pressey, R. L. (2000). Systematic conservation planning. *Nature*, 405(6783), 243-253.
- Marsh, N., Linke, S., Ogden, R., Wettin, P., and Cottingham, P. (2007). *River and catchment restoration prioritisation tools*. Paper presented at the 5th Australian Stream Management Conference. Australian rivers: making a difference, Charles Sturt University, Thurgoona, New South Wales.
- Peters, G. (2009). *AVIRA Environmental, Social and Economic Values and Threats Report*. Melbourne, Victoria: Department of Sustainability and Environment (DSE).
- Possingham, H. P., Ball, I. R., and Andelman, S. (2000). Mathematical methods for identifying representative reserve networks. In S. Ferson, and Burgman, M. (Ed.), *Quantitative methods for conservation biology* (pp. 291-305). New York: Springer-Verlag.
- Prosser, I. P., Young, B., Moran, C., and Hughes, A. (2001). *Constructing river basin sediment budgets for the National Land and Water Resources Audit*. Canberra.
- Rahman, J. M., Seaton, S. P., Perraud, J.-M., Hotham, H., Verrelli, D. I., and Coleman, J. R. (2003). *It's TIME for a New Environmental Modelling Framework*. Paper presented at the MODSIM, Townsville.
- Rondinini, C., Chiozza, F., and Boitani, L. (2006). High human density in the irreplaceable sites for African vertebrates conservation. *Biological Conservation*, 133, 358-363.
- Schildt, H. (2006). *C# 2.0. The Complete Reference*. California: McGraw-Hill/Osborne.