DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

Rapid Assembly and Deployment of Domain Visualisation Solutions

T.R. Pattison, R.J. Vernik, D.P.J. Goodburn and M.P. Phillips

DSTO-TR-1100

Rapid Assembly and Deployment of Domain Visualisation Solutions

T.R. Pattison, R.J. Vernik, D.P.J. Goodburn and M.P. Phillips

Information Technology Division
Electronics and Surveillance Research Laboratory

DSTO-TR-1100

ABSTRACT

Information visualisation exploits the natural perceptual capabilities of the decision-maker to facilitate the rapid assimilation and analysis of abstract, complex and often voluminous information. In this paper we argue that a major advance for computer-based information visualisation will be the definition of an open, component-based framework to support the rapid assembly and deployment of visualisation solutions. The proposed visualisation framework should not only support and extend the functionality common to existing visualisation tools, such as view generation and user interaction, but also provide support for the deployment of that tool into the customer's organisational environment. Successful deployment should encompass: the integration of the deployed visualisation tool with existing work processes, the information space and the available information technology tools and services; and ongoing support for the use and evolution of the tool. We describe *InVision* - an open framework for the development and deployment of integrated visualisation solutions, which employs computer-based visualisation techniques, component-based software engineering, and agent-based computing - and describe a research program to validate the *InVision* concepts and promote the adoption, extension, and integration of the *InVision* framework by visualisation researchers and application developers.

RELEASE LIMITATION

Approved for public release

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury South Australia 5108 Australia

Telephone: (08) 8259 5555
Fax: (08) 8259 6567

AR-011-706
February 2001

Rapid Assembly and Deployment of Domain Visualisation Solutions

Executive Summary

Australia's Strategic Policy assigns the highest capability development priority to the achievement of the "knowledge edge" over our adversaries. To see further and more clearly through the fog of war, the ADO requires not only better capabilities in the acquisition and dissemination of information, but also better exploitation of available information sources. One way to better exploit information sources is through the use of computer-based information visualisation.

Visualisation exploits the natural perceptual capabilities of the decision-maker to facilitate the rapid assimilation and analysis of complex and often voluminous information. In particular, information visualisation can aid the understanding of complex systems which have no physical form or cannot be seen by the naked eye, such as software systems, the World-Wide Web, and Defence systems. Military applications which will benefit from information visualisation solutions include the improvement of information management processes in headquarters, decision aids for the procurement of new capabilities and the selection of military response options in relation to changing strategic situations, and intelligence activities.

Visualisation has benefitted enormously from the advent of computers because of the flexibility and speed with which they can generate and adapt visual representations, the variety of visualisation techniques they can employ, and their capability for interaction with the user. Many tools and techniques already exist for computer-based visualisation, and their domain of application is potentially immense. The implementation of the more common visualisation techniques as software components by software vendors presents the potential for rapid, cost-effective, assembly of visualisation solutions customised to specific domains of application.

In this paper we argue that a major advance for computer-based visualisation will be the definition of a unified component-based framework to support the rapid assembly of information visualisation solutions. Our experience has shown that a visualisation "solution" must consist not only of a software tool which implements the requisite visualisation technique(s), but also: the deployment of that tool into the customer's organisational environment, including its integration with existing work processes, the information space and the available information technology tools and services; and ongoing support for the use and evolution of the tool. The proposed visualisation

framework should therefore not only support and extend the functionality common to existing visualisation tools, such as view generation, coordination and user interaction, but also address these generic deployment issues.

We describe a research program which draws from our previous experiences in deploying domain-specific, closed visualisation solutions, developed both in-house and commercially. We employ a number of technologies, including computer-based visualisation techniques, component-based software engineering and agent-based computing to define *InVision*, an open framework for the development and deployment of integrated visualisation solutions. The use of case studies and functional prototypes to validate *InVision* concepts is briefly discussed. A cooperative research and development approach based on open software principles and practice is being used to promote the adoption, extension and integration of the framework by visualisation researchers and application developers.

Expected short-term outcomes from this research are

- the provision to Defence customers of component infrastructure for the flexible assembly and adaptive deployment of visualisation solutions;
- the improved positioning of DSTO to provide advice on complete visualisation solutions, both through its own research program and through TTCP collaborations, in which the *InVision* infrastructure represents a solid Australian contribution.

A long-term goal of the *InVision* research program is to improve the flexibility and adaptivity of future ADO software systems in support of the more flexible command arrangements, including ad hoc coalitions, required by Australia's Strategic Policy. The component-based development of ADO software systems will lead to more cost-effective and robust military computing systems, while the *InVision* approach to software development and deployment will lead to the improved transfer of domain knowledge throughout their life cycle.

Authors

T.R. Pattison

Information Technology Division

Tim Pattison received a B.Sc. (Ma) in Applied Mathematics and Computing in 1988, a B.E. (Hons) in Electrical and Electronic Engineering in 1989 and a Ph.D. in neural network modelling of early vision in 1993, all from the University of Adelaide, South Australia. As an undergraduate he held an Engineering Cadetship with the Electricity Trust of South Australia, and later an Undergraduate Cadetship with the Advanced Engineering Laboratory (AEL) of the Defence Science and Technology Organisation (DSTO). His postgraduate research was supported by a Postgraduate Fellowship from the Communications Division of DSTO, and in part by the Cooperative Research Center for Sensor Signal and Information Processing (CSSIP), Pooraka, South Australia.

Between 1993 and 1996, Dr Pattison worked as a Research Scientist in the Satellite Networks Discipline of Communications Division, primarily on the signal processing and geometrical aspects of dual-satellite geolocation. He then moved to C3I Networks Discipline, working firstly on issues concerning graceful degradation and the optimisation of delivered value in communications networks, and later as leader of Organisational Networks and Processes Section. In 1999 he transferred to Software Systems Engineering Group of Information Technology Division, where he now leads the task entitled "Assembly and Deployment of Defence Visualisation Solutions", concerned with the research and development of information visualisation systems and their application to Recognised Enterprise Pictures.

R. Vernik

Information Technology Division

Dr. Rudi Vernik is employed as a Principal Research Scientist and is Head of Software Systems Engineering Group. His research interests focus on the definition, development, and application of new systems and software engineering approaches to support Defence in its development of capabilities that will facilitate knowledge and information-based warfare. His group is currently undertaking research in the areas of information visualisation,

component-based software engineering, domain characterisation and modelling, and enterprise instrumentation. Rudi has a Ph.D. in Computer and Information Science (Software Engineering) from the University of South Australia. He also has a Bachelor of Electronics Engineering (with Distinction) and a Diploma of Communications Engineering from the Royal Melbourne Institute of Technology.

D.P.J. Goodburn

Information Technology Division

Dr. Daniel Goodburn is employed as a Research Scientist in the Software Systems Engineering Group. His current research interests are software agents and systems visualisation. Previous work includes research into the use of expert systems to support mechanical engineering designers. He has a Bachelor of Science in the Faculty of Mathematical and Computer Science (Honours) from the University of Adelaide and a Ph.D. in Information Technology from the University of South Australia.

Matthew Phillips

Information Technology Division

Matthew Phillips is a researcher employed in Software Systems Engineering Group, Information Technology Division. His research interests include distributed systems, programming language design, software visualisation and object-oriented software engineering. Matthew has a Bachelor of Computer Science (with Honours) from The University of Adelaide.

Contents

1. Introduction

The visualisation literature abounds with information visualisation techniques and the experimental tools which demonstrate them. Some of these techniques have been incorporated into finished products, which range from general purpose visualisation environments to dedicated applications with specific visualisation requirements (see e.g. [7]). A number of the more common visualisation techniques have now been implemented as software components, including commercial off-the-shelf offerings such as ILOG's JViews [8] and Tom Sawyer's graph layout toolkit [9]. This commoditisation of visualisation techniques makes the rapid assembly of component-based visualisation solutions tailored to specific application domains a tantalising possibility. Exploitation of this trend, however, is currently limited by the lack of a common software architecture and framework to guide the design, implementation and integration of these components into a visualisation tool, and an associated methodology for the generation of visualisation solutions through the design, assembly and deployment of such tools into the application domain.

A visualisation "solution" consists not only of a software tool which implements the requisite visualisation technique(s), but also: the deployment of that tool into the customer's organisational environment, including its integration with existing work processes, the information space and the available information technology (IT) tools and services; and ongoing support for the use and evolution of the tool. In section 2 of this paper we summarise our experiences with the deployment of visualisation solutions which have highlighted, among other things, the need to ensure reliable access to the information to be visualised, and the need to support the integration of, and transition between, different view types. These experiences have lead us to the conclusion that a visualisation solution should consist of more than just pretty pictures – it must be integrated into, and evolve with, the application domain context.

Recent trends towards component- and agent-based software engineering [10, 11] are likely to have a major impact on the way in which visualisation solutions are assembled and deployed. Component-based software engineering offers the promise of "plug and play" visualisation architectures, where individual visualisation techniques, implemented for example as JavaBeans or COM/ActiveX components, plug into a framework which caters, among other things, for data collection and view integration. More rapid application development and lower cost is expected to result from increasing levels of code re-use. The potential for incorporation of software agents and agent frameworks into mainstream component-based software engineering as knowledge components holds the promise of intelligent software functionality at comparatively low cost. In visualisation applications, software agents could support users in their understanding of visual representations, provide attention direction [12], and facilitate access to, and filtering of, underlying data sources. The use of agents would furthermore confer greater adaptivity to changes in the application environment, which includes the users and their work processes, the information space, and other tools and services.

In response to these issues and considerations we have put together a research program which draws from the advances in component- and agent-based software engineering and our knowledge of computer-based visualisation approaches to focus on how we can more effectively assemble and deploy visualisation solutions for various domains of use. A result of this work has been the definition of a component-based visualisation approach which we call *InVision* [6]. Key features of this approach, which are discussed further in Section 3, include information and view integration, and model-based and knowledge-enabled visualisation. Future research and development of this approach, including ongoing development of the corresponding software framework, will be based on open software principles, as detailed in Section 4.

2. Domain Solutions

A range of issues need to be considered when deploying visualisation tools into specific domains of use. These issues relate to the user processes that need to be supported, the provision of effective access to and integration of underlying information sources, and the interoperation of the tool with other enterprise tools and services. In this section we outline the concept of domain visualisation solutions by proposing a model which defines the domain context and the processes that need to be supported. We then use this model to discuss some of our experiences in deploying visualisation solutions in software project domains. This is followed by a summary of what we believe are some of the key issues and considerations that need to be addressed.

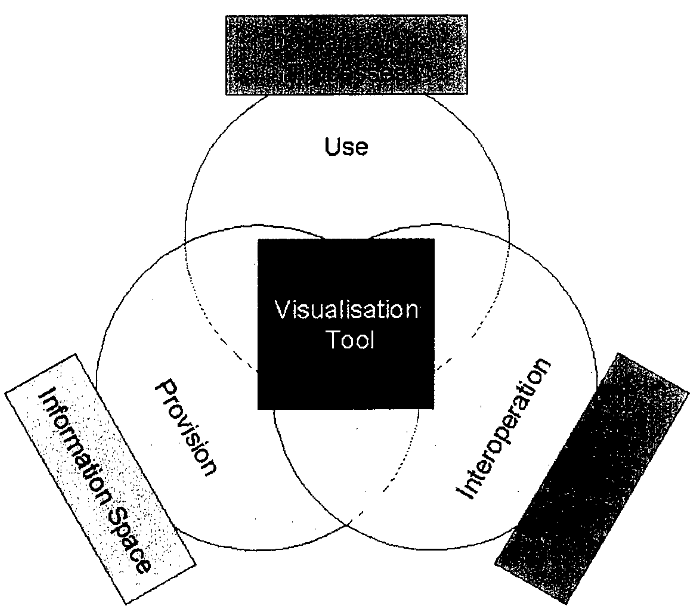

Figure 2-1. Domain solutions require support for Use, Provision and Interoperation processes

As shown in Figure 1, a domain context can be defined in terms of the Domain Work Processes which a user is undertaking, the underlying Information Space from which the information to be visualised must be sourced, and the Enterprise Tools and Services with which visualisation tools must interoperate. Within a domain context, we define three key processes that need to be supported if a visualisation tool is to be deployed as part of an effective visualisation solution. The Use process refers to the support required if the user is to understand and use visualisations to accomplish particular tasks. This includes support for human perception and cognition as well as support for other task related activities such as process workflow and the recording and reporting of results. The Provision process supports the identification, access to, filtering and integration of the information items which provide the basis for required visualisations. The Interoperation process provides for interfacing to other enterprise tools and services, including workflow engines and analysis tools.

2.1 Deploying Generic Tools

A range of generic visualisation tools are available commercially. Some of our earlier work looked at the issues which needed to be addressed if these tools were to be cost-effectively deployed into particular application domains. In one such experiment [4], we looked at how an early version of the Netmap tool could be used to gain improved visibility of large software systems for Independent Verification and Validation (IV&V) activities.

Netmap is a generic tool which has been applied to a variety of domain contexts such as the analysis of telecommunications networks, criminal investigations, knowledge management, and marketing [13]. It uses information sourced from one or more databases to generate its own database of nodes and links (relationships between nodes). Both nodes and links can be characterised by sets of attributes. Several display formats can be used to present this information, the key display type being a "netmap" where nodes are arranged in a circular fashion. Node grouping is shown by displaying nodes of the same group contiguously on the rim of the circle and by displaying satellite circles. Links between nodes of different groups are drawn across the hub of the circles. Colours can be used to represent attributes of both nodes and links. Various novel and visually appealing visualisations can be generated.

In using Netmap as the basis of a domain visualisation solution, a range of setup activities needed to be performed. Analysis was performed to define the types of information that needed to be presented to users, the availability of and access to underlying information sources, and the translation requirements. Several Netmap specific files were generated to define the data format for the Netmap database, attribute translations, and other configuration aspects. Various program scripts (Unix and SQL) were developed to support access to required information, data file concatenation, and the customisation of views. These initial setup activities took considerable effort and required expert knowledge of Netmap and the underlying data sources. Ongoing "maintenance" activities, such as adapting to new data sources, also proved costly.

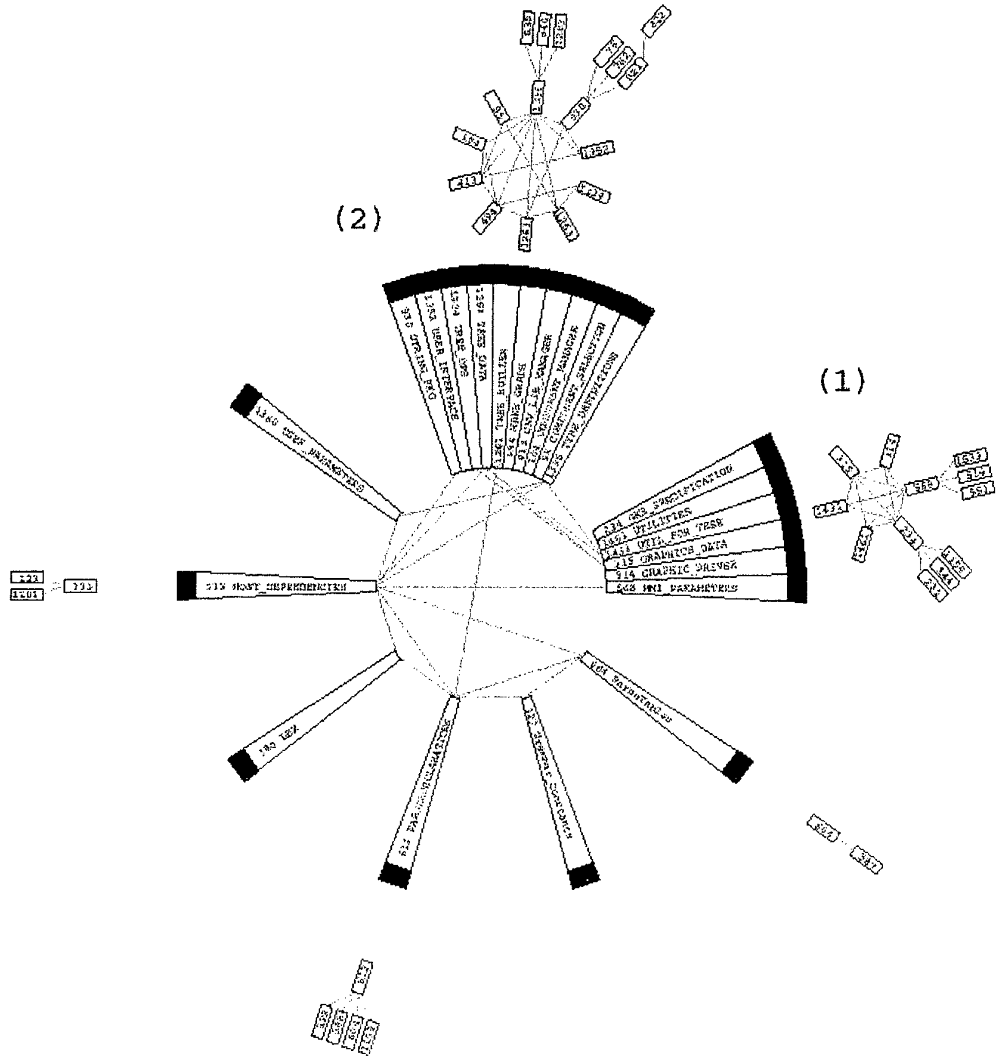

Figure 2-2. A Netmap Visualisation (taken from [4])

The visualisations produced by Netmap provided us with a good basis for analysis and (over time) allowed us to quickly identify important characteristics of the software being analysed. However, other users had more difficulty in using Netmap for specific IV&V tasks. Their feedback identified various concerns such as, "These are pretty pictures, but what are they telling me?", and "What are the important patterns that I should be concerned with?". They also had difficulties in interacting with the tool to adapt custom visualisations so as to provide additional information required for specific tasks.

Since Netmap is a "closed" proprietary tool, we were unable to extend its capabilities to integrate view types and graph analysis algorithms not provided by Netmap. This also made interfacing to other enterprise tools and services difficult. In summary, we concluded that these deployment and use issues conspired to limit the degree to which Netmap could be applied as the basis of a domain visualisation solution. And although our discussion has focused on the Netmap tool, many of the same criticisms apply to a large cross-section of the available visualisation tools.

2.2 Software Project Domain Solutions

Further experimentation was undertaken to investigate many of these issues and to look at how computer-based visualisation approaches could be more generally applied in an overall software project context. Large software projects generate vast amounts of information of various types such as requirements, design, code, test results, configuration data, and metrics. This information is used to describe various characteristics of the work processes, products and resources such as the degree of testing that has been carried out, areas of the software which exhibit poor quality and high defect rates, and progress in terms of the degree to which particular requirements have been implemented in relation to the resources expended. Computer-based visualisation shows promise for presenting this type of information in ways which meet the specific viewpoint needs, particularly for those individuals requiring high-level overviews such as project managers, quality managers, configuration managers, and team leaders. These roles often require the integration of various types of information.

As part of this work, an experimental visualisation environment was developed to provide views which could be rapidly customised and integrated for particular user needs [3]. The environment supported the use of data filters which provided access to underlying project data sources such as the configuration management system, code libraries, design tools, and measurement tools. A core feature of the environment was a systems modelling approach which supported the integration of information into an underlying descriptive model of the software. This model provided the consistent basis for generating the various views required by users. Automated update functions supported the seamless access to, filtering, and integration of underlying data in line with the project schedule. Process-based support allowed visualisations to be automatically generated for specific tasks based on a predefined task sequence. A direct manipulation interface allowed users to interact with the visualisations to further adapt them to their needs. Monitoring and reporting support allowed results to be reported within the context of the visualisations being used. This service also supported research objectives in that the instrumentation data captured helped evaluate what information was being used for particular tasks and the degree to which interactive adaptation of custom visualisations was being done by users. The environment was deployed into an industry-based organisation to provide support for the development of a large Defence software product and to support research activities.

This more extensive application of visualisation approaches highlighted many issues. Major problems resulted from changes to the domain environment. For example, project personnel updated and rearranged file systems which resulted in problems in our automated update process. We also had difficulties in determining when to trigger updates to the descriptive models and views based on changes in the information space. We concluded that approaches for more intelligently mapping and accessing sources within information spaces were required. Setup overheads remained significant because domain knowledge could not be used explicitly to help automate the deployment process. Moreover, we were not able to effectively reuse the knowledge that we had gained in a particular domain context to set up a similar solution in a corresponding domain. Also, even though we had introduced flexible ways of customising and integrating views to user needs, we were not able to easily integrate other visualisation techniques and so draw from the wealth of ideas that have been proposed through various research activities and the array of third party visualisation components that were becoming available (both novel components such as those provided by Inxight [14] and the more standard techniques such as charts and graphs). Another problem was that there was interest expressed in adapting and reusing the approaches that we had developed to other domains outside software development. Although our concepts were portable, our implementation remained domain specific.

2.3 Solutions in Other Domains

More recently, we have conducted studies to looked at how computer-based visualisation could be applied to a range of areas of interest to our Defence clients. Visualisation solutions are required to improve information management processes in military headquarters, to help make decisions relating to the procurement of new capabilities, to help decide on the most appropriate military response options in relation to changing strategic situations, and to support intelligence activities. To do this we need enhanced approaches for describing social networks, enterprise and systems architectures, systems of military strategies, and various forms of abstract data. Of key concern are the deployment issues, which if addressed appropriately, can result in the cost-effective application of visualisation approaches. The *InVision* approach proposed in this paper has been designed to address many of these issues. It provides the basis for assembling and deploying visualisation solutions in a variety of domains from sets of software components and domain knowledge.

2.4 Summary of Issues and Considerations

Many of the experiences gained through our past research activities have relevance to the deployment of visualisation tools in general. We have distilled our experiences into what we believe are some of the key issues that need to be considered if we are to provide effective visualisation solutions. These include:

- **Support for User Processes.** Visualisation involves the presentation of information to the user in order to improve their understanding, and ultimately to assist them in making decisions. Automatically producing a one-size-fits-all "picture" is often not possible, and is rarely sufficient for this purpose. Support is often required to aid the adaptation of custom visualisations to specific user needs, through for example direct manipulation interfaces. Additional support may also be needed to enhance the user's understanding of what is being presented and to direct attention to areas of importance in relation to the task being performed. Process-based support can provide for the automated generation of pre-defined visualisations for particular sequences of actions and the recording of decisions within visualisation contexts.

- **Robust Access to Underlying Information Sources.** Visualisations are typically used to present information stored in underlying information sources. Visualisation solutions need to know what sources of information are available at particular times, be able to automatically and efficiently access and filter desired information in line with user and task needs, and be able to adapt to changing situations such as the moving of information sources, changes in format etc. Since similar functionality is required by other applications, including enterprise portals, it constitutes a prime target for implementation in middleware.

- **Information Integration.** Users often need information from various sources to be integrated and presented as a composite visualisation. Support needs to be provided for integrating data from underlying information sources. However, this only addresses part of the problem. View and representational integration [3] also need to be considered when assembling and deploying visualisation solutions.

- **Reuse and Sharing of Visualisation Assets**. Visualisation assets include not only software but also less tangible assets such as knowledge, views and concepts. The sharing of software assets has been difficult due to proprietary architectures and infrastructure. The move to component-based software systems and open software development shows promise for supporting improved sharing at this level. Sharing of the less tangible assets is more problematic. For example, to more cost effectively deploy visualisation solutions we would like to share knowledge associated with the setup and evolution of solutions in various application domains. Sharing at the perceptive and cognitive levels also need to be considered. This might involve the sharing of views related to individual conceptualisations or mental models.

- **Interfacing to other tools and services**. Visualisation solutions need to be considered part of an overall enterprise computing environment. They need to be able to access other services to support user needs. For example, interfaces to other tools may be required to support advanced data analysis. There might also be a need to interface to other tools to extend the visualisation capability available to a user, by for example providing access to advanced visualisation techniques such as immersion. Consideration may also need to be given to interfacing with enterprise applications such as workflow engines.

Visualisation solutions must be cost effective. The issues discussed above all relate to cost in one way or another. They include costs associated with the development or procurement of visualisation software as well as cost of deployment (e.g. initial setup, evolution with changing needs and environment). Cost of use is also an important consideration. Visualisations are not an end in themselves: they are used to support people in doing particular tasks such as making faster and better decisions. Cost of use relates to the extent of training and cognitive effort required to achieve desired results.

3. *InVision*: A Component-Based Visualisation Approach

In addressing the key deployment issues which have emerged from our previous research, and in response to the ever increasing requirement for visualisation support within our client domains, we have defined a component-based development approach which we call *InVision*. *InVision* is not a tool – it is a set of generic component assets which, together with domain knowledge, can be assembled and deployed as specific visualisation solutions in various domains.

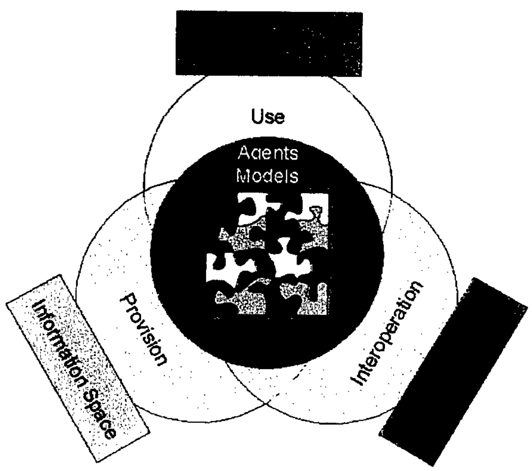

Figure 3-1. Visualisation solutions developed using the *InVision* approach.

3.1 Concepts

Figure 2 highlights a number of the key concepts of *InVision* in relation to the domain context depicted in Figure 1. These include:

- **Integrated component-based visualisation approach:** allows assembly and deployment of a wide range of visualisation solutions from a set of component assets which include infrastructure frameworks and pluggable components.
- **Model-based visualisation:** uses a unified information model to ensure that multiple viewpoints are consistent. Other aspects of the deployment environment, including the information space from which information is sourced and user work processes, are defined using the same modelling framework.

- **Knowledge-based deployment components:** support total domain solutions by providing agent support for interaction between the visualisation tool and its domain context, including functions such as information access and modelling, process-based visualisation, reporting, and user understanding. This intelligent functionality not only frees the user to concentrate on the visualisation task at hand, but also confers greater adaptivity to changes in the deployment environment, such as re-organisation of a file system or upgrading of supporting tools and services.
- **Open approach:** avoids building bloated, stovepipe solutions by intelligently interfacing to existing tools and services where available. *InVision* applications will also offer visualisation services for use by other tools.
- **Workspace support:** allows views to be tailored to specific viewpoints, integrated into view sets and hierarchically organised for ease of access. A simple example of a view set would be a view of a computer network and the specification of a supporting view, such as a chart which shows the recent history of a node's CPU usage, which is to be produced from interactively selected nodes on the principal view. Views and view sets can be shared between workspaces as required.
- **Process and workflow support:** allows the definition, editing, debugging and playback of visualisation processes involving multiple views and decision points, and the integration of these processes into enterprise work flows.

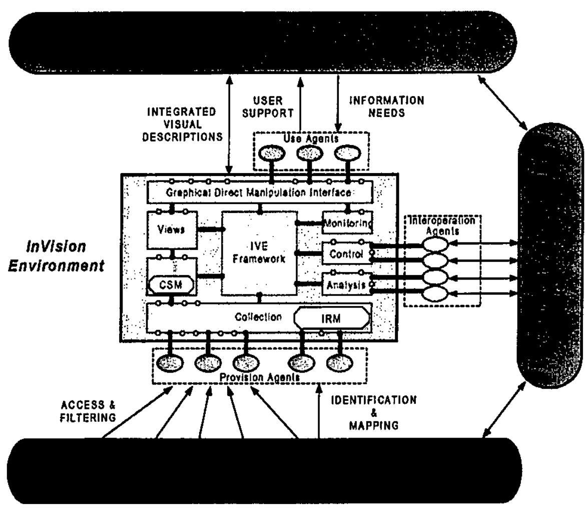

Figure 3-2. Conceptual architecture for *InVision.*

3.2 Architecture Overview

Figure 3-2 shows the conceptual architecture of an *InVision* application, which consists of a customised Integrated Visualisation Environment (IVE) and a set of deployment components (agents). The IVE is assembled from a set of component assets, including component frameworks (shaded rectangles) and individual components, and incorporates only those components which are needed to achieve the required functionality. In addition to the IVE framework, which integrates the various components and frameworks, the IVE will typically include component frameworks responsible for a graphical direct manipulation interface, modelling, information collection, view generation and management, usage monitoring, external control and information analysis. Each of these functions will now be described in turn.

A graphical direct manipulation interface (GDMI) supports the user in constructing and interacting with models, views and view sets. In addition to conventional direct manipulation techniques, components integrated by the GDMI framework will support functions such as animation and morphing. The modelling framework supports modelling of the information to be visualised, along with other aspects of the deployment environment such as the information space from which the information is to be collected. A variety of modelling components will support various model types, including discrete-event dynamic systems and multi-attributed graphs. An information collection framework integrates components, and interacts with agents, which find, collect and filter information from the information space. The view framework is responsible for the creation and management of views, including their integration into composite views and view sets. The monitoring framework records and monitors user interaction with the IVE to facilitate the creation of visualisation processes, and for later analysis for research purposes. The creation of and interaction with these processes is mediated by the control framework, as is interoperation with work flow applications and other tools and services in the deployment environment. The information analysis framework integrates components concerned with the analysis and fusion of the information extracted from the information space in order to convert that information into the form requested by the user. The extraction of simple facts from text documents and their fusion to form higher level facts is one example of an analysis function. Third party analysis and fusion tools would be invoked, if available, to fulfil functions not supported internally by the analysis framework.

3.3 Use of Deployment Components

Deployment components are those components which are responsible for embedding the IVE into its deployment environment. They are distinguished from the IVE in that whereas knowledge about the domain context captured during the requirements analysis phase is represented *implicitly* in the composition of the latter, it is represented *explicitly* in the deployment components. As shown in *Figure 3-2*, deployment components can be divided broadly into three categories on the basis of the environmental interaction process which they serve, and hence also the aspect of the deployment environment with which they interact. Use agents interact with the user to facilitate the production of, and interaction with, the required views, to assist the user in identifying salient features through attention direction, and in understanding their significance through descriptions and comparisons. One type of use agent might take a verbal description of the class of information to be visualised and return both a suggested meta-model of that information for refinement by the user, and a sample data set. Provision agents interact with the information space to locate and extract the requested information. These include both exploration agents, which are responsible for discovering and mapping out the contents of the information space, and collection agents, which must then extract the relevant information. Interoperation agents facilitate the interoperation of an *InVision* application with the available tools and services. To do this, they must first identify which tools and services are available.

In addition to intelligently identifying the relevant aspects of the target environment at the time of deployment, deployment components must also adapt to changes in that environment over time. Our experience has shown that user preferences and work processes, the content and structure of the information space, and the available tools and services continually evolve. The level of adaptivity, and hence also "intelligence", required by deployment components will vary according to the nature of application domain and the stated requirements. As the magnitude and abruptness of the environmental changes increase, so must the sophistication of the internal models of that environment and the ability to reason about that environment.

3.4 Application

InVision will support a large variety of view types, including graph layouts, statistical charts, tables and text views, many of which are now implemented in common off-the-self components. The use of novel and experimental view components is supported by the open architecture. The comparative strengths of these different view types can be simultaneously exploited through their composition or integration into view sets. Consistency between views is ensured by the use of a single model of the information to be visualised. Each view can be flexibly customised to suit the individual viewpoint of the user; the customised view can then be regenerated on demand to reflect changes in the underlying data.

Figure 3-3 presents a screenshot from an *InVision* functional prototype. The main window is divided into two panels: the view pane (right) and the currently selected multi-purpose panel (left). The required multi-purpose panel can be selected using the tabs provided; reading from left to right, these correspond to the user workspace (shown), the features of the model being visualised, the overlays defined for the current view, visualisation processes, and usage monitoring. The workspace panel contains the user workspace (top) and the view accessories (bottom) areas. The user workspace contains models, processes, views and agents, with various view types collected into folders and integrated into view sets. A view entitled "Users and Server allocation" has been selected from the workspace and displayed in the view window; it depicts use relationships between users and databases, and hosting relationships between databases and servers. Two of the pre-defined overlays for this view have been selected from the accessories area for display: a colour overlay on the databases (shown as cylinders) which indicates the percentage of the maximum storage capacity currently in use; and an overlay containing lines corresponding to the set of information flow relationships. A chart of database usage and server load for a selected database is floated (bottom right) as a supporting view. The accessories panel (bottom left) lists other overlays which are not currently applied, and a description agent which provides a textual description of the content of the view. The overlays are defined in terms of the attributes of, and relationships between, the elements of the corresponding system model on which the view is based. An example of these model features and a corresponding overlay is provided by the two left-hand panels in Figure 3-4.

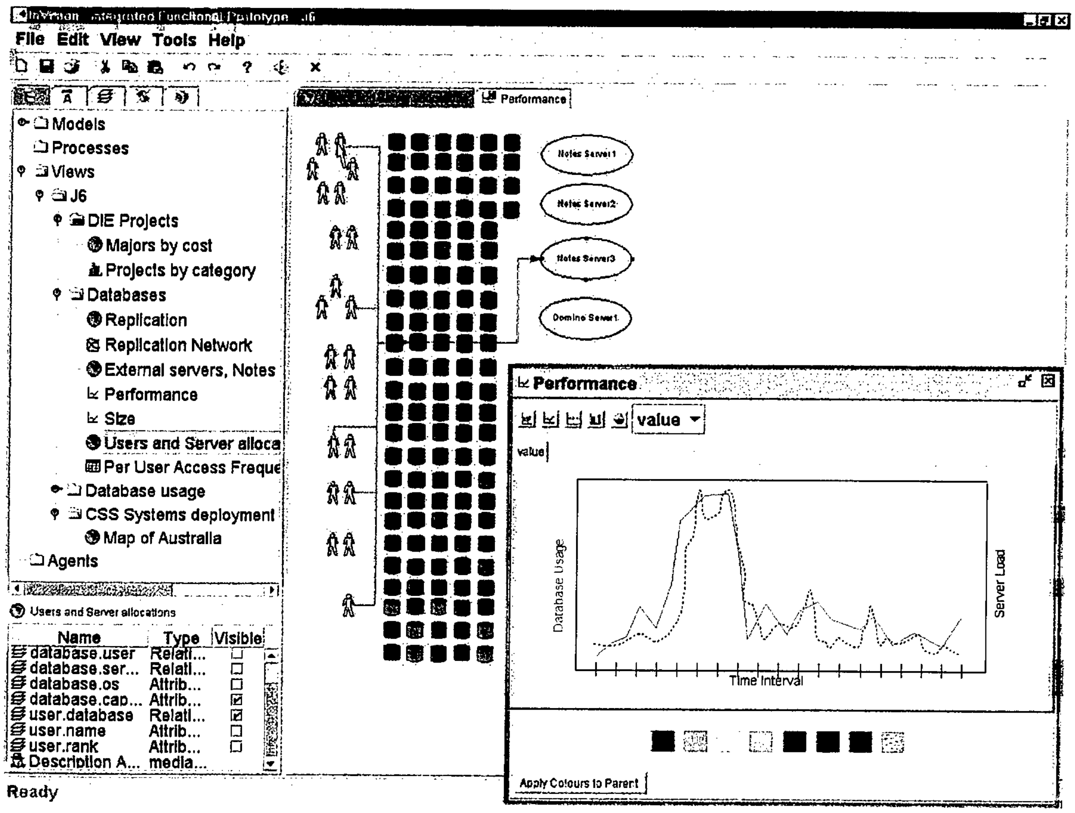

Figure 3-3. Screenshot of an *InVision* functional prototype showing a workspace (top left), CLOVIS view (right), the associated accessories panel (bottom left), and a supporting chart view (floated, bottom right).

The main views in Figure 3-3 and Figure 3-4 are examples of CLOVIS views [15], a novel view type currently being developed within the *InVision* component infrastructure. CLOVIS views use hierarchically nested layouts – a form of representational integration – to provide flexible, customisable and scalable visual overviews, a simple example of which is shown in Figure 3-4. Here the departments of an organisation are laid out vertically (with wrapping at the bottom of the screen). Each department is represented by the horizontal juxtaposition of a (partially elided) tree, corresponding to the departmental hierarchy, and an array layout of the databases maintained by that department. The hierarchical specification of this nested layout is shown in the floated window at bottom right. Within this window, the nesting of layouts is specified using a tree (left), in which each node corresponds to a notional container. The model elements which belong in the selected container, and the layout of those elements, or of the subcontainers between which they are further divided, is specified at right. Tabs are used to select between the different layout strategies, each of which is implemented as a separate component.

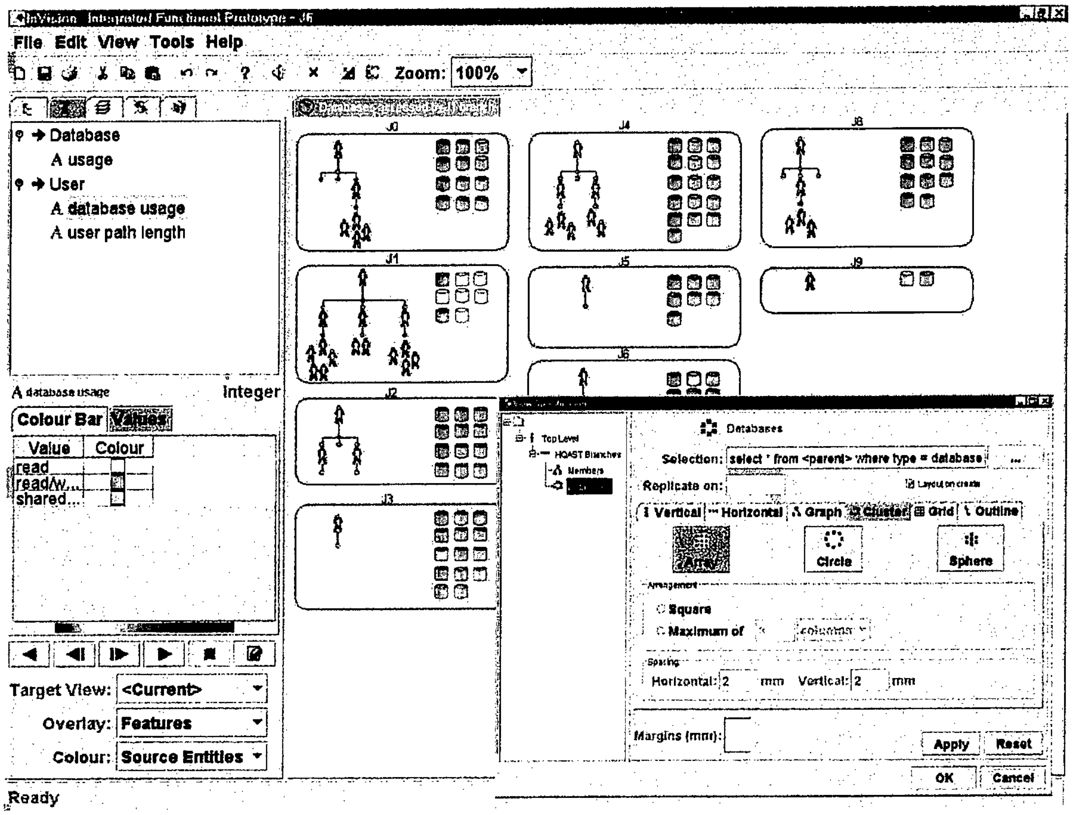

Figure 3-4. CLOVIS view and associated view specification (inset) for visualisation of a socio-technical system.

3.5 Case Studies

The use of case studies has been and remains an important part of our research approach. A number of case studies have been undertaken to elicit visualisation requirements in various military application domains. Knowledge of domain requirements obtained through these case studies has been captured in storyboard and functional prototypes in order to demonstrate *InVision* concepts and to provide initial validation of those concepts. A preliminary implementation of the CLOVIS view component has been used to provide more detailed concept validation for several application domains, including software systems, socio-technical systems and organisational structures. As further component assets are implemented or procured, these will be integrated into experimental prototypes in different application domains in order to continue the work of concept validation and to test the component assembly process. This assembly process has also been tested through the integration of third party graph layout components, produced as a result of collaboration with the University of Newcastle, into one such prototype.

4. An Open Software Approach

Where to from here? To date we have developed the vision, concepts, architecture, and a range of infrastructure components for the *InVision* approach. We have validated many of the concepts through case study activities and a range of prototypes. Moreover, we have established linkages between other researchers who have begun to contribute various concepts and components to the growing asset base. In effect, we have the basis of an open software initiative.

Open software development grew out of the research community's need to draw from and share ideas and infrastructure. This resulted in the GNU General Public Licence [16] which provided the mechanisms and legal basis for the sharing of software. Open software development has since extended beyond the research community to include a number of commercial organisations such as Netscape, IBM, Oracle, Informix, and Corel who are migrating their commercial applications to open software development. These companies have seen the value of allowing a global network of developers to fix problems and provide enhancements to their products. Connectivity through the internet and self interest combine to provide a powerful basis for co-operative development. Commercial value comes not so much from the intellectual property embedded in the software, but from the services and leverage that can come from its effective deployment and use.

We believe that the visualisation area is a good contender for open software development because of its broad applicability and potential for innovation. It seems inconceivable that in the current climate any organisation would find commercial benefit in developing the proprietary infrastructure and component base for this fast moving and diverse area. Significant amounts of research activity in this area is providing a wealth of new ideas and approaches. To provide return on research investment, some organisations such as Xerox [14] and Lucent [17] are beginning to market innovative visualisation techniques as software components. A host of other visualisation components are becoming available commercially, as shareware, and through development environments such as those provided by Microsoft and Inprise. Our aim is to leverage this activity to provide for more rapid transition of visualisation techniques into domain specific solutions to support improved client capabilities. We are also keen to extend the scientific research base to focus on the many issues that need to be considered for the effective deployment and use of visualisation approaches. We plan to use *InVision* as the basis of an open software approach to support these aims.

5. Conclusion

Our past experience has identified a number of problems with the deployment of existing visualisation tools and approaches into specific application domain contexts. In addition to the inflexibility of stovepipe tools based on closed architectures, we have identified three main areas in which support for the deployment of visualisation tools is currently lacking: support for and integration with user work processes, flexible and robust access to the information to be visualised, and interoperation with other tools and services. The development of component-based infrastructure to support the cost-effective assembly and deployment of custom visualisation solutions has been proposed. The necessary infrastructure, component assets and methods are currently under development. Agent support for intelligent interfacing with the deployment environment, including adaptivity to changes in that environment, is seen as an essential feature of the *InVision* architecture.

The *InVision* approach proposed in this paper is based on an integrated program of research into visual representations, component-based software engineering, the mainstream use of knowledge-based deployment components (or agents), and systems modelling. It forms a basis for collaborative research and development between DSTO, other research organisations, academia, and industry. Ongoing experimentation is being conducted to further validate and extend the concepts underlying the *InVision* approach.

6. References

1. Australian Department of Defence, *Australia's Strategic Policy*. 1997: Directorate of Publishing and Visual Communications.
2. Price, B.A., R.M. Baecker, and I.S. Small, *A Principled Taxonomy of Software Visualisation*. Journal of Visual Languages and Computing, 1993. **4**(3): p. 211-266.
3. Vernik, R.J., *Visualisation and Description in Software Engineering*, in *Computer and Information Science*. 1996, University of South Australia: Adelaide. p. 232.
4. Duffett, P.L. and R.J. Vernik, *Software System Visualisation: Netmap Investigations*, 1997, Defence Science & Technology Organisation.
5. Demetriades, P.A. and A. Poulovassilis, *An Architecture for Web Visualisation Systems*, 1998. [Online Accessed 6 May 1999] URL: http://www.dcs.kcl.ac.uk/pg/petros/Publications/www7/ReuseWorkshop.htm
6. Goodburn, D.P.J., *et al.*, *Integrated Visualisation and Description of Complex Systems*, DSTO Research Report, DSTO-RR-0154, 1999.
7. Reed, M. and D. Heller, *OLIVE: On-line Library of Information Visualization Environments*. [Online Accessed 28 September 2000] URL: http://www.otal.umd.edu/Olive

8. ILOG, *JViews Component Suite.* [Online Accessed 10 October 2000] URL: http://www.ilog.com/products/jviews

9. Tom Sawyer Software, *Graph Layout Toolkit.* [Online Accessed 10 October 2000] URL: http://www.tomsawyer.com/glt/index.html

10. Szyperski, C.A., *Component Software: Beyond Object-Oriented Programming.* 1997: Addison-Wesley.

11. Wooldridge, M., *Agents and Software Engineering.* AI*IA Notizie, 1999. **XI**(3): p. pages 31-37.

12. Rogers, E., R.R. Murphy, and B. Ericson. *Agent-Based Expert Assistance for Visual Problem Solving.* in *Autonomous Agents '97.* 1997.

13. Davidson, C., *What your Database Hides Away.* New Scientist, 1993. **Jan 1993.**

14. Inxight, *Inxight Software,* [Online Accessed 10 October 2000] URL: http://www.inxight.com

15. Pattison, T., R. Vernik, and M. Phillips, *Information Visualisation Using Composable Layout and Visual Sets.* To be submitted, 2000.

16. Stallman, R.M., *The GNU Public License.* [Online Accessed 10 October 2000] URL: http://www.fsf.org/copyleft/gpl.html

17. Visual Insights, [Online Accessed 10 October 2000] URL: http://www.visualinsights.com/home/visinsighthome.asp

Rapid Assembly and Deployment of Domain Visualisation Solutions

T.R. Pattison, R.J. Vernik, D.P.J. Goodburn and M.P. Phillisp

(DSTO-TR-1100)

DISTRIBUTION LIST

	Number of Copies
AUSTRALIA	
DEFENCE ORGANISATION	
Task sponsor:	
DGC4	1
Deputy Director Battlespace Digitisation	1
S&T Program	
Chief Defence Scientist)	
FAS Science Policy)	1 shared copy
AS Science Corporate Management)	
Director General Science Policy Development	1
Counsellor, Defence Science, London	Doc Control Sheet
Counsellor, Defence Science, Washington	Doc Control Sheet
Scientific Adviser to MRDC Thailand	Doc Control Sheet
Scientific Adviser - Policy and Command	1
Navy Scientific Adviser	Doc Control Sheet and distribution list
Scientific Adviser - Army	Doc Control Sheet and distribution list
Air Force Scientific Adviser	1
Director Trials	1
Aeronautical & Maritime Research Laboratory	
Director	1
Electronics and Surveillance Research Laboratory	
Director	Doc Control Sheet and distribution list
Chief Information Technology Division	1
Research Leader Command & Control and Intelligence Systems	1
Research Leader Military Information Enterprise	1
Research Leader Advanced Computer Capabilities	Doc Control Sheet
Research Leader Joint Systems	1
Head, Information Warfare Studies Group	Doc Control Sheet

Head, Trusted Computer Systems Group	Doc Control Sheet
Head, Systems Simulation and Assessment Group	Doc Control Sheet
Head, C3I Operational Analysis Group	Doc Control Sheet
Head, Information Exploitation Group	Doc Control Sheet
Head, Intelligence Group	Doc Control Sheet
Head, Intelligence Group	Doc Control Sheet
Head, Human Systems Integration Group	Doc Control Sheet
Head, C2 Australian Theatre	1
Head, Distributed Systems Group	Doc Control Sheet
Head C3I Systems Concepts Group	1
Head, Military Systems Synthesis Group	Doc Control Sheet
Head, Systems of Systems Group	Doc Control Sheet
Head, Advanced Network Integrity Group	Doc Control Sheet
Head, Enterprise Visualisation, Instrumentation and Synchronisation Group	1
Dr Tim Pattison EVIS Group, ITD	1
Dr Daniel Goodburn, EVIS Group, ITD	1
Dr Matthew Phillips, EVIS Group, ITD	1
ITD Publications & Publicity Officer, ITD /Executive Officer, ITD	1 shared copy

DSTO Library and Archives

Library Fishermans Bend	Doc Control Sheet
Library Maribyrnong	Doc Control Sheet
Library Salisbury	1
Australian Archives	1
Library, MOD, Pyrmont	Doc Control Sheet
US Defense Technical Information Center,	2
UK Defence Research Information Centre,	2
Canada Defence Scientific Information Service,	1
NZ Defence Information Centre,	1
National Library of Australia,	1

Capability Systems Staff

Director General Maritime Development	Doc Control Sheet
Director General Land Development	Doc Control Sheet
Director General Aerospace Development	Doc Control Sheet

Knowledge Staff

Director General Command, Control, Communications and Computers (DGC4)	Doc Control Sheet
Director General Intelligence, Surveillance, Reconnaissance and Electronic Warfare (DGISREW	Doc Control Sheet
Director General Defence Knowledge Improvement Team (DGDKNIT)	Doc Control Sheet

Army

Stuart Schnaars, ABCA Standardisation Officer, Tobruck Barracks, Puckapunyal VIC 3662	4
SO (Science), Deployable Joint Force Headquarters (DJFHQ) (L), MILPO, Gallipoli Barracks, Enoggera, QLD 4052	Doc Control Sheet

Intelligence Program

DGSTA Defence Intelligence Organisation	1
Manager, Information Centre, Defence Intelligence Organisation	1

Corporate Support Program

Library Manager, DLS-Canberra	Doc Control Sheet

Universities and Colleges

Australian Defence Force Academy	1
Library	1
Head of Aerospace and Mechanical Engineering	1
Serials Section (M list)), Deakin University Library	1
Hargrave Library, Monash University	Doc Control Sheet
Librarian, Flinders University	1

Other Organisations

NASA (Canberra)	1
AusInfo	1
State Library of South Australia	1
Parliamentary Library, South Australia	1

OUTSIDE AUSTRALIA

Abstracting and Information Organisations

Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts US	1
Documents Librarian, The Center for Research Libraries, US	1

Information Exchange Agreement Partners

Acquisitions Unit, Science Reference and Information Service, UK	1
Library - Exchange Desk, National Institute of Standards and Technology, US	1

SPARES	5

Total number of copies: 53

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)
N/A

2. TITLE Rapid Assembly and Deployment of Domain Visualisation Solutions	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)
4. AUTHOR(S) T.R Pattison, R.J. Vernik, D.P.J. Goodburn and M.P. Phillips	5. CORPORATE AUTHOR Electronics and Surveillance Research Laboratory PO Box 1500 Salisbury SA 5108 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO-TR-1100	AR-011-706	Technical Report	February 2001

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
N9505/19/266	JNT 00/130	DGC4	27	17

13. URL ON THE WORLDWIDE WEB	14. RELEASE AUTHORITY
http://www.dsto.defence.gov.au/corporate/reports/DSTO-TR-1100.pdf	Chief, Information Technology Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, SALISBURY, SA 5108

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS
Information visualization
Complex systems
Visualization
Defense systems

19. ABSTRACT
Information visualisation exploits the natural perceptual capabilities of the decision-maker to facilitate the rapid assimilation and analysis of abstract, complex and often voluminous information. In this paper we argue that a major advance for computer-based information visualisation will be the definition of an open, component-based framework to support the rapid assembly and deployment of visualisation solutions. The proposed visualisation framework should not only support and extend the functionality common to existing visualisation tools, such as view generation and user interaction, but also provide support for the deployment of that tool into the customer's organisational environment. Successful deployment should encompass: the integration of the deployed visualisation tool with existing work processes, the information space and the available information technology tools and services; and ongoing support for the use and evolution of the tool. We describe *InVision* - an open framework for the development and deployment of integrated visualisation solutions, which employs computer-based visualisation techniques, component-based software engineering, and agent-based computing - and describe a research program to validate the *InVision* concepts and promote the adoption, extension, and integration of the InVision framework by visualisation researchers and application developers.