

Towards the realization of an integrated decision support environment for organizational decision making

Shaofeng Liu¹, Alex HB Duffy², Robert Ian Whitfield², Iain M Boyle² and Iain McKenna³

¹ University of Plymouth, Business School, Hampton Street, Plymouth, PL4 8AA, UK,
Email: shaofeng.liu@plymouth.ac.uk

² University of Strathclyde, Department of Design Manufacture and Engineering Management, 75 Montrose Street, Glasgow, G1 1XJ, UK.

³ BVT Surface Fleet Ltd, Primary Contractor, South Street, Scotstoun, Glasgow, G14 0XN, UK.

Abstract

Traditional decision support systems are based on the paradigm of a single decision maker working at a stand-alone computer or terminal who has a specific decision to make with a specific goal in mind. Organizational decision support systems aim to support decision makers at all levels of an organization (from executive, middle management managers to operators), who have a variety of decisions to make, with different priorities, often in a distributed and dynamic environment. Such systems need to be designed and developed with extra functionality to meet the challenges such as collaborative working. This paper proposes an Integrated Decision Support Environment (IDSE) for organizational decision making. The IDSE distinguishes itself from traditional decision support systems in that it can flexibly configure and re-configure its functions to support various decision applications. IDSE is an open software platform which allows its users to define their own decision processes and choose their own existing decision tools to be integrated into the platform. The IDSE is designed and developed based on distributed client/server networking, with a multi-tier integration framework for consistent information exchange and sharing, seamless process co-ordination and synchronisation, and quick access to packaged and legacy systems. The prototype of the IDSE demonstrates good performance in agile response to fast changing decision situations.

Keywords

integrated decision support environment, organizational decision making, multi-tier integration framework, function configuration and re-configuration

1. Introduction

Over last several decades decision support systems (DSS) have experienced a paradigm shift from a stand-alone system that supports a single decision maker to make a specific decision through group decision support systems (GDSS) to organizational decision support systems (ODSS). Through ODSS distributed decision makers interact with one another and their decisions are co-ordinated towards mutually defined goals, i.e. the goals of organizations. Organizational decision making is a demanding task because the decisions that need to be made involve all aspects of an organization including their products, technologies and personnel management. When considering

the impact from the whole supply chain and global market such as end customers, material providers and product retailers, organizational decision making is further complicated. Due to the nature of organizational decision making in terms of its complexity, dynamics, multiple goals and often opaqueness, various types of decisions need to be made at different times and in distributed organizational units. Different decision processes or even multi-processes may be used. Further the decisions can be well-structured, semi-structured, ill-structured or unstructured (Lee et al, 1999). These decisions can also be made at different levels of organization such as strategic, tactical or operational. Therefore, decision support for organizational decision making is challenging, which has motivated broad interest in research on ODSS in recent years (Carter et al, 1992; Sen, Moore & Hess, 2000). Key challenges for organizational decision making have been widely discussed including decision analysis across different levels (Humphreys & Bekerley, 1985), the management of multi-directional propagation of decisions: vertical, horizontal, or hub-and-spoke (Whitfield et al, 2007), and various presentation of decision problems (Holsapple, 2008). What lacks so far from existing work however is the success of mature decision support software which can address the dynamic nature of the business environment that today's organizational decision making situates (Vahidov & Kersten, 2004). New business situations crop up every day, to develop a new set of decision tools to deal with each new decision situation seems highly unrealistic. It is not only costly and time-consuming, but also it causes existing decision tools to become obsolescent, which could result in a tremendous loss of information, knowledge and business intelligence embedded within the tools (Szykman et al, 2001; Taghezout & Zarate, 2008). Therefore, it is increasingly important to allow the use of both new and existing tools to provide faster and better decision support without causing high cost economically or intellectually (Bangeman et al, 2006; Shi et al, 2007).

This paper proposes a novel Integrated Decision Support Environment (IDSE) aiming to meet the new challenges of organizational decision making, in dynamic situations, through a hybrid integration approach. The IDSE has been designed for providing decision support capability where systematic decision making processes rather than emergent decision making processes are appreciated in the organization (Liu, Duffy, Whitfield et al, 2009). The paper is organised as follows: Section 2 gives an overview of related work. An architecture of the IDSE is proposed in Section 3, followed by Section 4 focussing on an integration framework that enables IDSE functionality. Section 5 presents the development of an IDSE prototype and its application to a case study in organizational decision making context. Finally, Section 6 discusses further issues and draws conclusions.

2. Related work

A decision support environment distinguishes itself from a traditional decision support system, and other information systems, by the feature of functionality reconfiguration. IDSE is a decision support environment that can provide flexible functions according to the changes of decision settings for varied applications. Most traditional decision support systems provide fixed functions despite their success in many application areas (Carlsson & Turban, 2002; Eom, 1999; Respicio & Captivo, 2008; Shim et al, 2002).

Other information systems such as EDP (Electronic Data Processing), MS/OP (Management Science and Operations Research) and MIS (Management Information Systems) have made contributions to decision support from the perspectives of data/information provision and management, but they also do not address the changing nature of decision making and provide corresponding solutions (Hopple, 1988). One stream of research attempted to address this issue was the proposal of DSS generators (Sprague & Watson, 1986; Hemant, Sridhar & Herrick, 1999). The DSS generators can assemble necessary capabilities from a set of DSS tools (new technology, new languages, new hardware and software) to configure specific DSS faster and easier to develop models, data and user interfaces that are customised to the application's requirements. The IDSE however goes one step further, which can integrate capabilities from a set of systems to configure a computer environment for varied decisions under varied situations, including decision making on ill-structured and non-structured decision problems.

IDSE is designed and developed based on the ODSS concept and therefore differs from a GDSS (Group Decision Support Systems) and EIS (Executive Information Systems). GDSS and EIS (now called Enterprise Information Systems) were both developed as complementary to but more powerful support tools than traditional DSS, in the sense that GDSS can provide brainstorming, idea evaluation and communication facilities to support team problem solving (Anson et al, 1995; DeSanctis & Gallupe, 1987; Limayem et al, 2006), and EIS extended the scope of DSS from personal or small group use to the corporate level and can provide a wide variety of information such as critical success metrics, key information indicators, reports with the ability to drilldown to underlying detail, budget information, plans and objectives, competitive information, news and more (Elam & Leidner, 1995; Kosaka, 1994; Power, 2003). ODSS were developed based on the advances in GDSS and EIS, but had its focus on organizational decision making. It provides a mechanism for a large, geographically dispersed, decentralised organization to allow individual managers to make decisions within their own domains while maintaining consistency with the decisions made by other managers and organizational goals (Gachet & Haettenschwiler, 2003). In short, it provides distributed decision support to distributed decision making on varied applications. Carter et al (1992) summarised the difference of an ODSS from a traditional DSS in five aspects including purposes, politics, approach to building, focus on functions and components. This paper will focus on its technical side and discuss IDSE from system integration viewpoint (i.e. the components and relationships between components) (Liu, Duffy, Whitfield et al, 2009) and explore how the system integration approach will provide new strengths to ODSS.

ODSS have been evolving along two main categories. One category is to support organizational decisions where systematic decision making processes are mainly undertaken. This category includes organizations such as most manufacturing firms (Jaganathan, Erinjeri & Ker, 2007), service operations (Sen, Moore & Hess, 2000) and maintenance (Zeydan & Colpan, 2009) companies. The other category is to support organizational decisions where emergent decision making mostly occur such as in the Accident and Emergence service in medical organization (Knudtson, 2005), army force (battlefields) (Gardner & Filipelli, 2001) and fire fighting services. The IDSE is intended

to support organizational decisions in the first category, but with its new strength in that it is an open platform which allows users to define their own decision processes, choose other existing tools and new developments from other parties to be readily plugged-in and play, and therefore extra functions from those tools and developments can be configured into the IDSE and work together to support the decision processes.

The characteristics of the IDSE lie in two dimensions, as summarised in Figure 1: firstly its flexibility of functionality, and secondly its capability to support organizational rather than individual or team (group) decision making.

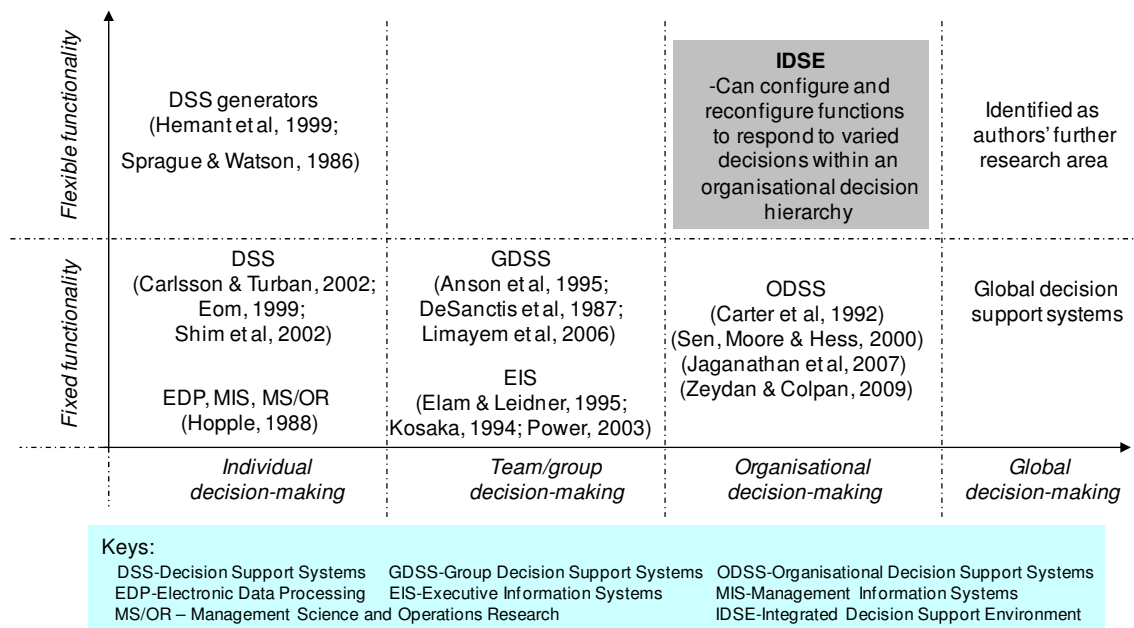


Figure 1 IDSE relevance to other types of decision support systems

3. An architecture of the Integrated Decision Support Environment

The basic paradigm for a traditional DSS is that it consists of three major components: a model base management system (MBMS) with a model base, a database management system (DBMS) with a database, and a user interface (UI) dialog system that manages the interaction between the user, and the model base and the database. Due to the limitation of the functions provided by the three components and the “hardwiring” between the components, a DSS is destined to support specific applications with specific decisions under specific settings for specific users (decision makers), as illustrated in Figure 2(a). In an ODSS, these three basic components are still often the same as those of a traditional DSS, although there may be differences on how the components are designed and used.

To support organizational decision making, with varied applications that deal with varied decisions under varied settings for varied users (decision makers), an ODSS requires additional elements and functions. For example, network for communication. More importantly, it requires flexible but reliable mechanisms that allow agile configuration of the system components to provide support to the varied applications.

This is realised through hybrid integration approaches within IDSE, as shown in the Figure 2(b). The three components (i.e. the UI, the DBMS and the MBMS) comprise the basic components of the IDSE. The three basic components provide constant and fundamental support to applications (represented by a straight through symbol \uparrow in the Figure 2(b)). IDSE has been designed and developed as an open platform which allows additional tools and modules to be readily plugged-in and play. To illustrate how the tools and modules can be seamlessly integrated within the platform to provide extended functions and capabilities for organizational decision making, this section introduces four additional components: a DM (decision management) component, an RR (resource reasoning) component, a CPM (change prediction and management) component, and a CW (collaborative working) component. Their support to applications is flexible based on the configuration of the components (represented by a switch symbol \boxtimes in the Figure 2(b)). This section discusses the architecture of the IDSE, and Section 4 will discuss the integration issue in detail.

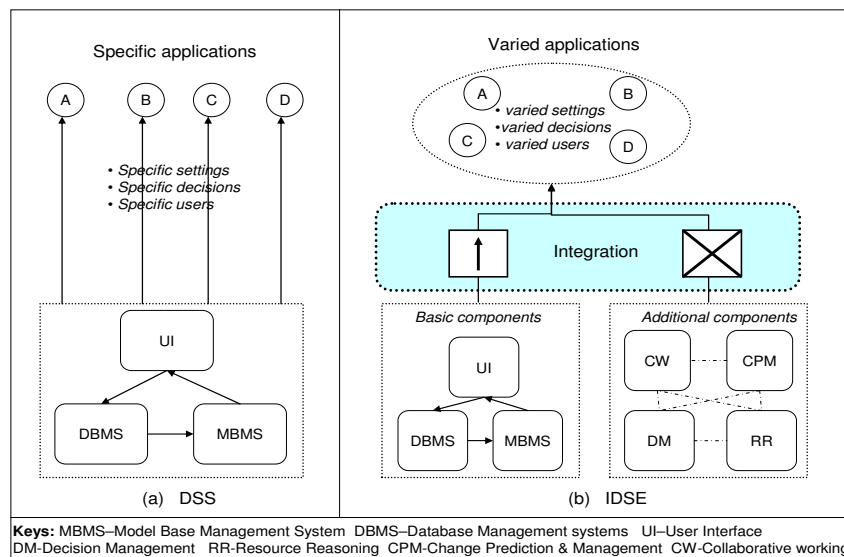


Figure 2 How IDSE differs from a traditional DSS

IDSE components and their functions

From computer software system viewpoint, system architecture comprises three important aspects: components; the externally visible properties of those components; and the connections between the components (Bosch, 2000). Figure 3 shows the architecture of the IDSE, with illustration of IDSE basic and additional components, integration across the components, and a network-based Client/ Server (C/S) infrastructure which enables the connection of disparate components.

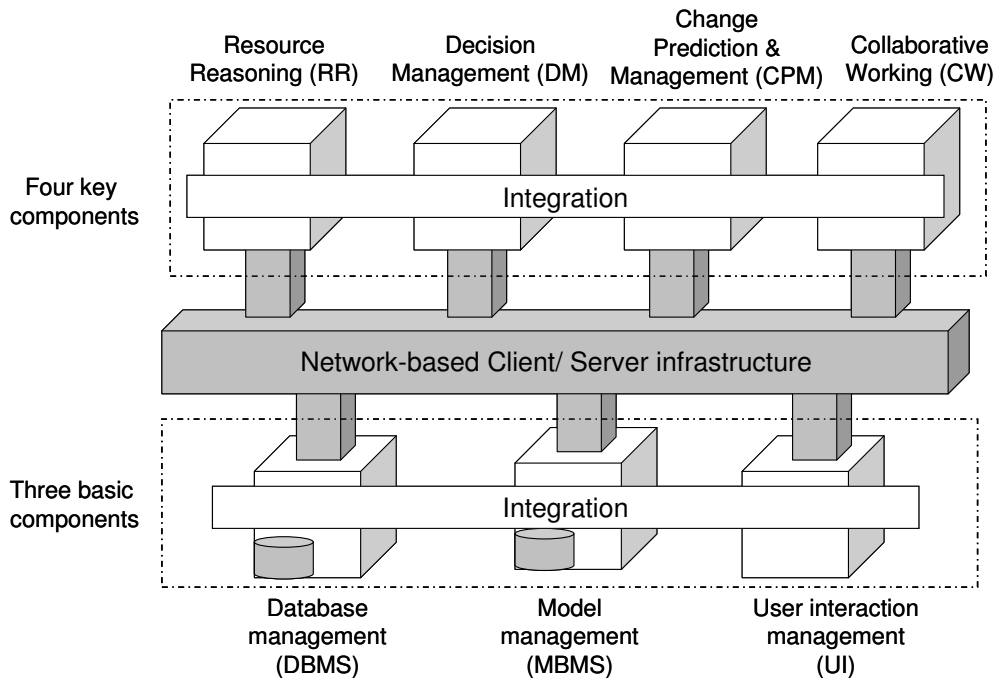


Figure 3 An architecture of the IDSE

The functions of the three basic components (i.e. database management, model management and user interaction management) of the IDSE remain the same as in traditional DSS and have been well defined in literature (Hopple, 1988). This section focuses on the functions of the four additional components, i.e. the Resource Reasoning (RR) component, the Decision Management (DM) component, the Change Prediction and Management (CPM) component and the Collaborative Working (CW) component.

The DM component is designed to manage decision hierarchies and dependencies in an organization as well as CoA (Course of Action) planning. The RR component provides IDSE with the capability to search for the right resources including facilities and human resources across organization units for decision tasks. This is developed from state-of-the-art ontology mapping techniques and a well-developed resource knowledge repository. CPM component provides the IDSE with the capability of assessing any changes of decisions and their consequence propagation along hierarchies and organizational units before a change is carried out. Finally, the CW component provides interactive and collaborative capability to team decision making in an organization when mutual decision goals (i.e. the organizational goals) are defined but decision preferences vary for different decision makers at different organizational levels. While these four components have their distinguished functions, the specification of the relationships between the components allows them to communicate, to share information, and to invoke and call services from each other when needed.

The relationships between the three basic components

Figure 4 shows how the MBMS, the DBMS and the UI identified as the basic components of IDSE work together to support decision makers (users). Decision makers initiate the communication with the IDSE and provide necessary inputs. The UI subsystem then talks to the MBMS to answer user queries, performs relevant sensitivity (what-if) and other analytical tasks. In the meantime, it talks to the DBMS to access data as required. The DBMS can also provide direct data and information support to the UI. The solid arrows in the Figure 4 show that direct data and information access and sharing occur between the components. The direction of the arrows represents information flow. The close relationship between the three basic components implies that a tight integration approach would be appropriate in this case (to be discussed in Section 4 in detail).

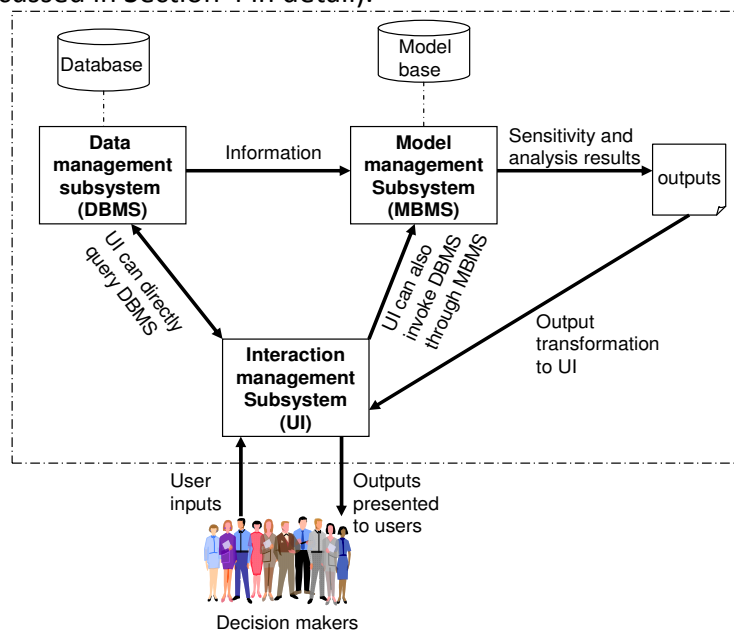


Figure 4 Relationships between the three basic components

The relationships between the four additional components

Efficient and effective organizational decision making depends not only on the decision makers to make good judgement on resources that are required for the decision tasks, but also on good management of decision hierarchy where decision makers position in, and decision maker's interaction and collaboration with fellow decision makers. Especially when decision makers try to make changes of the decisions, how the consequences will propagate along the decision hierarchy. While these four components (DM, RR, CPM and CW) have their distinguished functions, the specification of the dependencies between the components allows them to seamlessly interact with each other. Figure 5 illustrates the relationships between the four components represented with SysML (System Modelling Language). In the Component Diagram, the directions of the arrows show the information flow from one component to another, and the labels attached to the arrows show the nature of the messages and services that are communicated between the components.

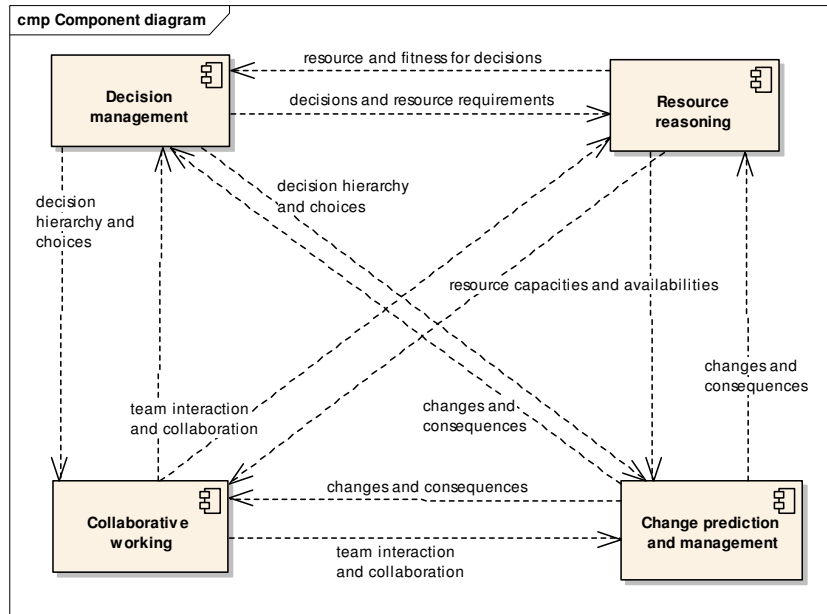


Figure 5 Relationships between the four additional components

4. An integration framework for the IDSE

Section 3 has discussed IDSE basic components, additional components and their relationships. This section discusses an integration framework which comprises hybrid (both tight and loose) integration strategies across multi-tiers (system tier, business logic tier and data tier). It is the integration that brings together disparate decision components, so that IDSE can work as a coherent software environment to provide reliable and flexible support to organizational decision making. This integration framework is an evolution of the authors' previous research on a hybrid integration approach for distributed design co-ordination (Whitfield, Duffy & Coates, 2002).

Hybrid integration strategy

The hybrid integration approach taken to develop the IDSE is a combination of tight integration (through integration standards) and loose integration (through integration middleware). Specifically, the integration of the three basic components is undertaken through a tight approach, and the integration of the four additional components is undertaken through the loose integration approach. The difference between the tight integration (also called coupling) and loose integration (also called cohesion) within the IDSE is that tight integration binds components (such as DBMS and MBMS) together in such a way that they are dependent on each other, sharing data, methods and interfaces. In contrast to tight integration, loose integration is the "act or state of sticking together" or "the logical agreement" (Linthicum, 2004). Cohesively integrated components (such as DM, RR, CPM and CW) are independent from one another. Changes to any source and target components should not affect the others directly. In this case, information is still shared between components but without worrying about changes to the components, leveraging some type of middleware layer to move information between components, and make adjustments for differences in

component semantics. The tradeoffs have been considered in the IDSE through a combination use of integration middleware for cohesion and integration standards for tight coupling, as shown in Figure 6.

The advantage of having a combination of tight and loose integration within the IDSE is that the balance between reliability and flexibility is maintained. Through the loose integration with middleware, components such as DM, RR, CPM and CW can be added to, changed or removed from IDSE without typically requiring changes to any of the other components according to the varied application requirements in the organizational decision making domain. Integration middleware (a special piece of software in the case of IDSE) thus provides the technology infrastructure of most-cohesive integration solution. It is able to account for the differences between components, accommodating differences in application semantics within a middle-tier process. Despite the flexibility provided by the integration middleware, common decision making processes are to be reused within IDSE, therefore tight integration through standards such as XML (Extensible Mark-up Language) and XML-RPC (Remote Procedure Call) provides high speed and method sharing with great reliability. The disadvantage of having the hybrid integration approach is its complexity of implementation. In the future, IDSE will look into exploration of Web Service as an integration standard, and Java EE (Java Enterprise Edition platform) will be investigated as the new integration broker for IDSE.

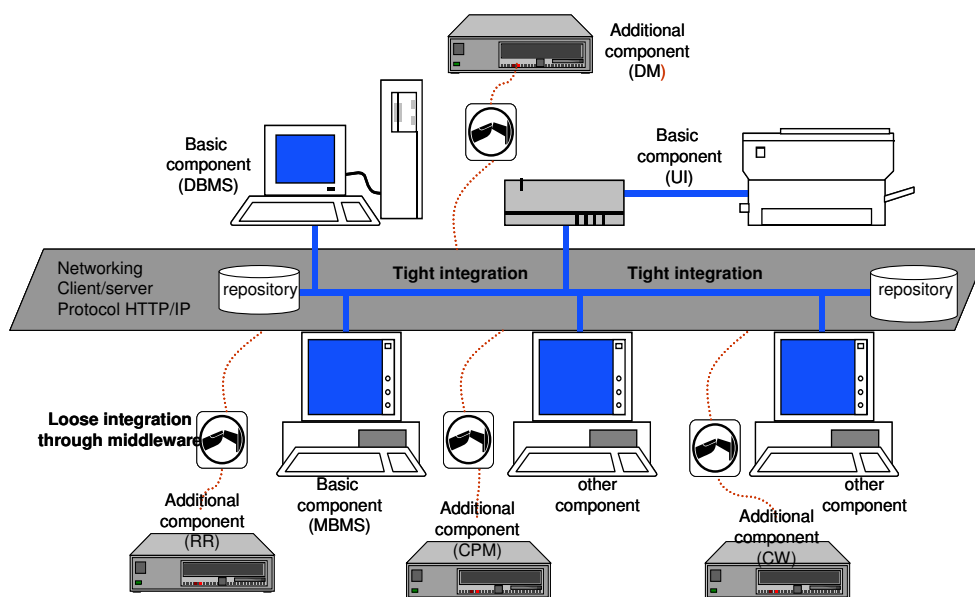


Figure 6 Tight and loose integration strategies

A multi-tier integration framework

The above tight and loose integration strategies have been implemented for the IDSE through a multi-tier integration framework. Figure 7 presents the integration across three tiers: system tier, business logic tier and information tier. At the system tier, the integration focuses on tool integration which supports easy tool plug-in and play, so that the IDSE can provide quick access to packaged, custom and legacy systems. At the

business logic tier, seamless process integration enables co-ordination and synchronisation across ranges of decision making processes and process units. Semantic information and knowledge integration at data tier is the foundation for higher (business logic and system) tiers because it addresses data integrity across decision processes and decision tools.

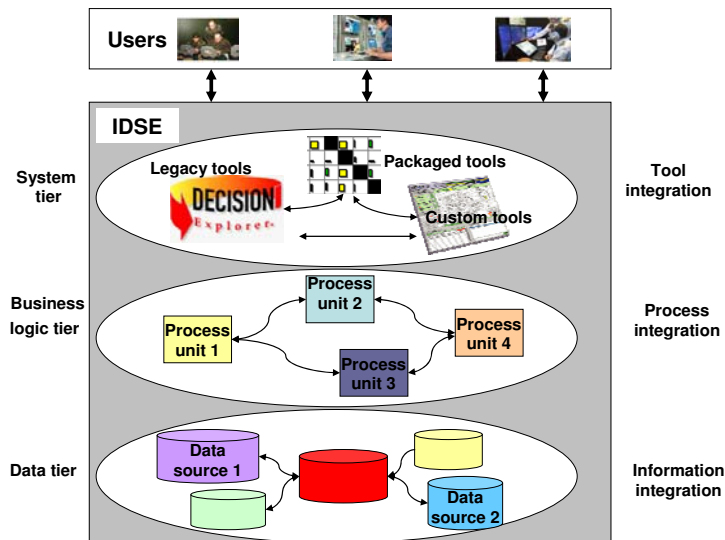


Figure 7 A multi-tier integration framework

Information and knowledge integration

To maintain data integrity, information and knowledge integration within the IDSE is achieved through three means: mapping, sharing and access. A semantic map has been defined and maintained so that meta-data such as the format, types, sources and versions of information and knowledge are explicitly captured and available for all decision components. An advantage of defining the semantic map is that the consistency of the information and knowledge can be maintained so that distributed decision makers in an organization can make use of the consistent information and knowledge to make better decisions. To provide such a high level of integration for disparate information and knowledge islands is currently the most demanding task because it depends on the creation of the meta-model representing the “schema” of information and knowledge contained in various decision tools, which can be time and intellectually consuming. At the other end, the information and knowledge access within the IDSE has been achieved with the least challenge. In this case, a decision tool can invoke another tool through a pre-defined interface, and access the information and knowledge embedded within that tool. But the information and knowledge can only be managed within respective tools. For example, the CPM tool can invoke RR search engine and access information within the cross-organization data-space, but the request will be rejected if the CPM tool tries to modify the information within the data-space. Lying between the above two means, information and knowledge sharing allows the management of information and knowledge contained in one decision tool from another. This provides the ability to add, modify and delete contents of the information and knowledge repository maintained in one tool from another. Take the DM tool and Process Management tool - a legacy system inherited by IDSE from authors’ earlier work (Wu, Duffy & Whitfield, 2007) - as an example, the DM tool can

not only use existing decision processes, but also can modify existing processes and create new processes. All the processes can be of course directly managed by the Process Management tool.

Process integration

Seamless process integration has been achieved from the following perspectives:

- workflow: this perspective is about what should be done (i.e. identifying activities and tasks for the process) and when should they be done (i.e. specifying the sequences and flow of the activities and tasks).
- informational: with what information can the process be done?
- organizational: who are allocated to do the tasks and processes?
- operational: how are the tasks actually done?

Within the IDSE, the above perspectives have been implemented through the integration of process units, process events and process constraints. A process unit is a unit of work that yields a result such as an identified task from the workflow perspective. A process event is a condition that arises during a process step that may result in the execution of an associated action, for example, the generation of a notification from one process unit to another, including informing the responsible decision makers (operational and organizational perspectives). A process constraint imposes constraints on some aspects of the process. It can be in the form of assumptions, status or requirements (informational perspective) of a process unit. Successful process integration provides the capability for co-ordination and synchronisation across ranges of decision tasks, decision tools and decision makers within the IDSE.

Tool integration

Tool integration at system tier is based on the loose integration strategy, considering that the IDSE should provide one single standard interface, for the benefit of users, to allow various new and existing tools to plug-in and play. Therefore, the integrator needs to be de-coupled from the platform itself and the tools to be wrapped, in which case it has to be a middleware in nature. To achieve this, the IDSE provides a standard interface, named as a tool wrapper, to allow flexible configuration and re-configuration of tools. The tool wrapper has the intelligence to detect input requirements of the tool and the outputs generated from the tool. The format of the tool can also be flexible, no matter they are executable tools, jar files or in other format. The tool wrapper provides the capability for the IDSE to quickly respond to changing environments by readily configuring and re-configuring various tools and making use of their functions, as well as the embedded information and knowledge. However, it may require data conversion for the embedded information and knowledge in one tool to be available and used by other tools.

Integration across the above three tiers (i.e. tool integration at system tier, process integration at business logic tier, and information integration at data tier) within the IDSE is achieved through the combination of business process flow and information flow simultaneously. Because the requirements of integration of tools (what tools to be integrated, and what functions of the tools are to be used) are mainly determined

by the tasks that are identified for the business processes. All tools are configured against the tasks within the process, and the co-ordination and synchronisation across the tasks provide the connection (requirements for interaction) between the tools. In particular the information flow, supported by information integration, through the tasks provides the key link between the tools. Therefore, implementation of the multi-tier integration framework provides the capability of allowing IDSE to be developed as a coherent but flexible decision support environment.

5. IDSE prototype and application

To test the IDSE concept and evaluate the architecture (Section 3) and integration framework (Section 4) proposed in this paper, a software prototype, named as VIP-DS (Virtual Integration Platform for Decision Support), has been developed by adopting a platform base from authors' earlier research (Wu, Duffy & Whitfield, 2007). Its application has been undertaken with a case study. Firstly, to allow the software prototype to provide collaborative decision support in an organization, it is important to look at the enabling technologies that have been employed for the development of the prototype.

Technologies for VIP-DS development

The requirements and functions specified for the VIP-DS are closely dependent on the software architecture and integration strategies. But whether these requirements will be met and whether the functions will perform as expected is also directly dependent on the fundamental technologies behind the software development.

Key requirements of IDSE include support for distributed, multiple users, varied software platforms such as different operating systems (Windows, Unix, etc.), and different data and information format. Based on the above understanding, the VIP-DS has been developed based on the following key technologies:

- Distributed networking for Client/Server communication to support distributed users (decision makers);
- Multi-threading for concurrency management;
- Java programming for portability; and
- XML for interoperability.

Distributed networking for communication

The Client and Server architecture-based VIP-DS is developed on distributed networking technology to support multiple users involved in organizational decision making. The VIP-DS server side tackles the issues such as platform administration, project data management, process definition, and task dependency management. The association of expertise (including decision makers in terms of organizational decision making process) and tools (including legacy decision systems) with decision tasks is also dealt within the process and task management. Registered users can log onto the platform from the client side, download platform functions and use decision tools that have been configured for their decision purposes. The VIP-DS takes advantage and makes use of the widely adopted and mature communication mechanism – TCP/IP

(Transmission Control Protocol/ Internet Protocol), so that all users speak the same communication language. Of course, the VIP-DS can be run on any type of network including company regional network and Internet, where distributed decision makers can access.

Multi-threading for concurrency management

While the VIP-DS aims to satisfy the requirement that multiple users can work on the platform at the same time, the issue of concurrency management has been raised during the platform development. Different decision makers involved in the organizational decision making should be able to process their decision tasks simultaneously, but without getting slowed down because of multiple users' access traffic. To solve the issue, multi-threading technology is employed so that each user can be served by a special thread. Because multiple threads are running in the same application, they share the same memory space in the computer, which allows them to share information seamlessly. Figure 8(a) shows the multi-threaded environment that the VIP-DS is developed in. Multi-threaded environment is different from the traditional multi-tasking environment (Oaks & Wang, 2004), as illustrated in Figure 8(b). In a multi-tasking environment, even though all the applications can access various types of shared memories, the shared memory is often restricted to information put there by other programmes. Also the APIs to access the shared information are usually quite different from the APIs used to access other data in the program. This type of data sharing is fine for dissimilar programmes, but it is inadequate for the VIP-DS. In the VIP-DS, for example, the process server sends process status information to all its clients through a network. Sending a process status to a client is a discrete task and may be done in a separate thread. In fact, if a process client must acknowledge the process status, then sending the data in a separate thread is recommended because we do not want all clients to wait for a particularly slow client to respond. Here the data to be sent to all the clients are the same. We do not want each client to require a separate server process which then must replicate all the data held by every other server process. Instead, multi-threading in one programme is an ideal solution so that they may share data and each perform discrete tasks on that data. Conceptually, multi-threading seems to be the same as multi-tasking. But there is a key difference in a Java environment, in which global memory is the entire Java heap: threads can transparently share access between any object on the heap. Each thread still has its own space for local variables that are specific to the method that the thread is executing, but objects are shared automatically and transparently between threads. Multi-threading is important to VIP-DS because it allows all clients to share a consistent view of the process.

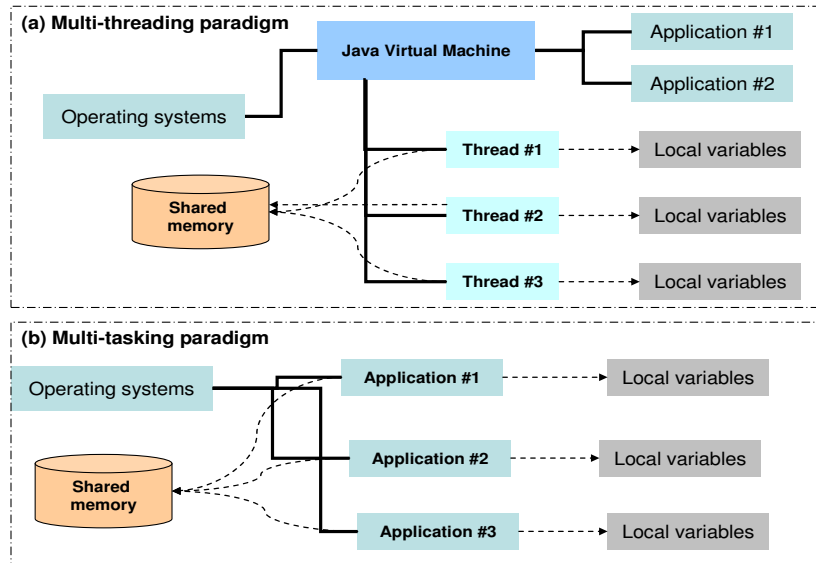


Figure 8 VIP-DS multithreaded environment contrast to multi-tasking environment

Java programming for portability

The VIP-DS prototype has been developed using Java programming language given that Java is platform independent. The VIP-DS first version was developed on a Windows platform, and also works on a Unix platform because of the Java’s cross-platform portability. Especially, Java’s strengths in network and multi-thread programming have consolidated the VIP-DS performance in terms of reliability and efficiency.

XML for interoperability

Because XML has been widely recognised as a standard for engineering information, it has been chosen as the format for the information integrated within the platform and shared between users. However, local users can have their own databases and store their information in any format they wish, and information transfer between local users also can be in any other format rather than in XML.

Accordingly an XML database was chosen for VIP-DS to store meta-information including project information, decision making process information, and other shared information between decision makers such as parameter data. The advantage of employing a pure XML database is that the information stored in the database doesn’t need to be well-structured, which is the case for information required for most decision situations in an organization. Also, the indexing mechanism of an XML database is powerful in the sense that users can easily and quickly retrieve information fragments within the XML documents at varied granularity if needed (Liu et al, 2006).

VIP-DS core functionalities

Core functionalities of VIP-DS are based on the functions provided by the four key components, i.e. the Resource Reasoning (RR) engine, the Decision Management (DM) tool, the Change Prediction and Management (CPM) tool and the Collaborative Working (CW) environment.

Functions provided by RR engine

RR component has been designed to provide the capability to characterize and abstract resources used within a decision-making system and the properties of decision-making technologies, within an organizational context and a distributed networked environment. This requires characterisation of state (what is known from through life system management at a given time) and prediction (what is known in the future, given current state); actions (the potential decisions) and cost (that captures the consequences of actions). Based upon this understanding, a software tool – RR engine – has been developed. Currently, a cross organization data-space (Chung & Liao, 2008) has been created and the RR engine has the following functions:

- to allow multiple distributed users to register for the cross organization data-space, implemented with authentication and authority mechanisms to control the level of access rights each user may have to the data-space;
- to allow users to contribute information to the data-space using commonly used means such as emails (email itself and attachment); and
- to allow users to search for resource information in single source and across multiple sources within the data-space.

Functions are being implemented within the RR engine include to:

- expand its capability of providing resource fitness assessment and resource configuration responding to decision requirements; and
- take into account the impact of collaboration and breakdown points between resource components.

Functions provided by DM tool

The DM component has been designed to develop a co-ordination framework to enable decision-making resources to be recruited, configured and employed within the VIP-DS and the development of mechanisms that allow network enabled decisions to be made, by controlling the interaction between decision making system components, and the constraints imposed by the target system and user goals. A particular focus of the DM requirement is to ensure constraints of time criticality are met and that account is taken of the multiple levels of interactions between elements of the organizational system within a distributed environment. A DM tool has been designed and developed to meet the above requirements. Functions of the DM tool that have been implemented are (Boyle et al, 2008):

- an internal model of the hierarchical nature of a capability delivery process;
- manipulation of this internal model;
- the ability to define expected situations;
- the ability to incorporate monitoring of expected situations;
- triggering mechanisms and the control mechanism by which is selected; and
- CBR (Case-Based Reasoning) control mechanism for recalling decision cases.

Further implementation of the DM functions includes:

- main process visualisation and interface development;
- file management services;
- control of logical relationships within a process;

- ability to interact with external software components to facilitate creation and execution of a process; and
- ability to incorporate decision support provisions into a process.

Functions provided by CPM tool

The CPM component has been designed to develop techniques to identify critical features and interfaces of target systems susceptible to change. This allows the impact of proposed decisions on the target system to be assessed more readily and decreases the size of the decision space (a collection of related decision options) that must be explored in the decision making process. Accordingly, a CPM tool has been developed. In organizational decision context, often there are a collection of related decisions to be made. For example in a ship maintenance case, there are decisions that need to be made on the shaft repair, hull repair and bearings etc. If the decisions on the hull repair change, then they could have an impact on the shaft and bearings maintenance for example regarding cost and lead time. CPM tool currently provides the following functions (Keller & Clarkson, 2008):

- to allow users to model systems;
- to analyse combined risks of change;
- to predict change propagation path.

Functions that are to be implemented include: through life support to the above functions, and to take into account decision requirements, resource configuration and collaboration requirements for change analysis through interactions with RR engine, DM tool and CW environment.

Functions provided by CW environment

The CW component has been designed to provide the interactive and collaborative tools that are required to address the human aspects of decision support. Research has been undertaken to gain an understanding of how individuals and teams can be supported in organizational decision making. A CW environment has been designed and is under development to allow effective interaction, distribution, communication and co-ordination of decision making activities across teams. An ADEPT (Advanced Design Environment for Prototyping with Task models) framework has been successfully used for modelling user, task and interface characteristics that provides designers with access to techniques that emphasise the system user and the tasks that the user needs to carry out (Johnson et al, 1993). For the VIP-DS, a revised ADEPT-C (for collaboration) framework has been proposed that enables the development of systems that can support collaborative decision-making (Carrigan and Johnson, 2008). Currently the ADEPT-C framework is being instantiated into a software tool – the CW environment. Main functions of the CW environment are:

- to assist with the development of models in the ADEPT-C framework (a collaboration model, a user/group model and a system model); and
- to automate the generation of instantiation of the concrete system model, i.e. to generate system interfaces that support collaborative work.

VIP-DS application

This section discusses the application of VIP-DS to a case study. The core concept of the case is to generate a work plan to repair a damaged ship through a Support Solution coalition which represents a decision network across organizations. The ship is required to be back to service in a certain amount of time period and the damage of the ship has been assessed. The Support Solution coalition is composed of decision makers along the supply chain of the ship maintenance. To evaluate the VIP-DS support in network-based decision making, role-play exercises have been undertaken by different types of decision makers. The purpose of the role-play exercises was to create a decision making situation, by giving the participants a set of tasks to finish for generating the work plan. During the case, they needed to make a series of decisions. All role-play exercises were videoed, transcripts were then generated from the video and analysed (Boyle et al, 2007). Immediately following the role-play exercises, semi-structured face to face interviews with the decision makers were conducted so that probing questions could be asked to clarify what was observed from the video, and to find out what was behind the decisions and how the participants made the decisions. As this paper is about how the VIP-DS can flexibly configure and re-configure its functions to support varied decisions involved in decision making, the following Figure 9 shows a process comprising seven tasks captured by VIP-DS (the activities/ tasks are pre-defined and the process is originally formulated by the user based on organization business requirements and competences) that are required for the ship maintenance case:

- Task 1: Identify if the maintenance work is suitable for the Support Solution coalition based on the availability of resources, current workload and business profit prospect etc.
- Task 2: Identify tasks involved in order to repair the ship based on the damage assessment, for example the need to repair the shaft, propeller, bearings, hull etc.
- Task 3: Identify decision criteria such as minimum cost or shortest repair time.
- Task 4: Identify potential options for the ship maintenance. For example, one option would be mending the propeller, using an alternative shaft with new bearings.
- Task 5: Evaluate feasibility of a particular option based on its performance parameter against the decision criteria.
- Task 6: Identify change impacts for options by analysing changes of decision choice. For example, by using an alternative propeller rather than mending the propeller, what's the knock-on effect on subsequent decisions on the decision network?
- Task 7: Select options if they are considered as satisfactory. If not, go back to Task 4 for more options.

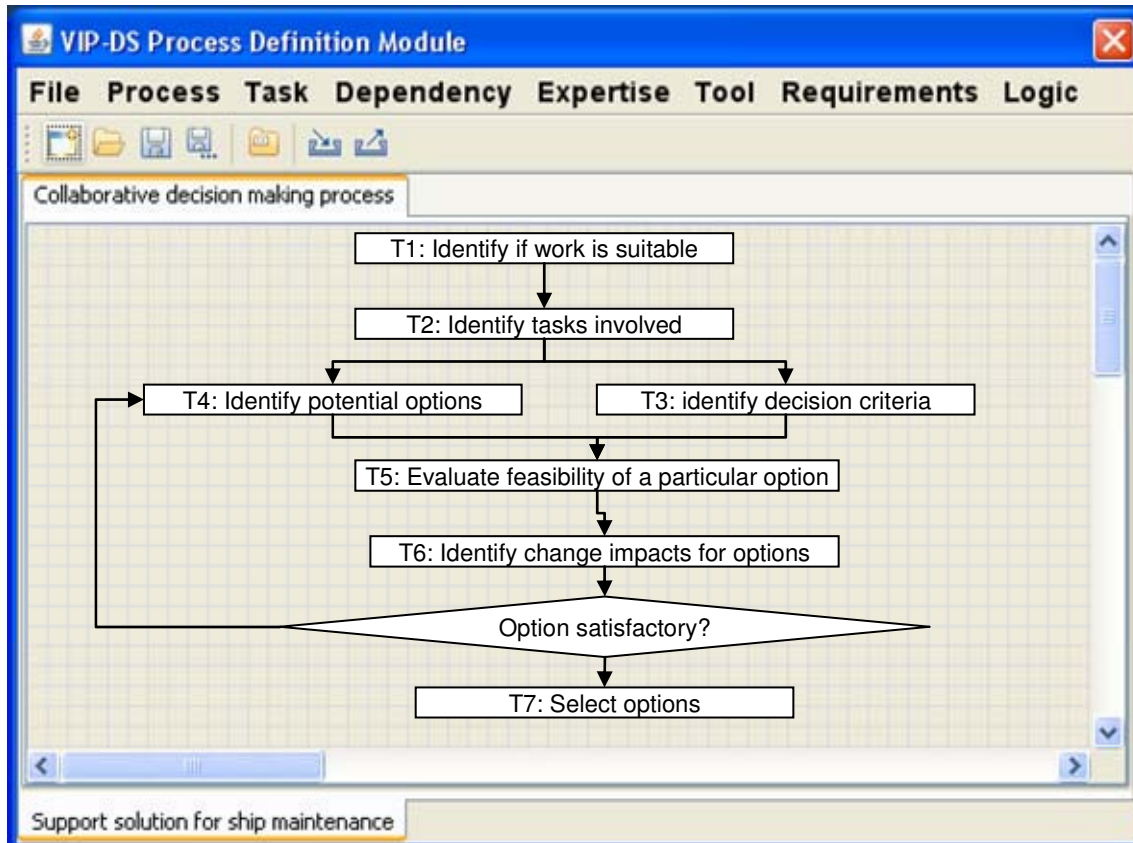


Figure 9 A collaborative decision making process for ship maintenance case

The dependency between the tasks has been defined and represented by arrows in Figure 9 to control the process flow. For each task, decision makers (through “Expert” in menu bar) were identified and allocated, specific decision tools (through “Tool” in the menu bar) were specified and configured, decision requirements (through “Requirements” in the menu bar) were captured, and process co-ordination and synchronisation rules (through “Logic” in the menu bar) were defined. The information flow through the tasks, decision makers and decision tools was enabled through the implementation of the integration framework discussed in Section 4. The details of the decision tools (the user can decide on what tools to be integrated) and decision makers’ roles for the above process are summarised in Table 1 (only Task 1 to Task 6 are shown in the table; task 7 is done mainly based on decision makers’ judgement, with no tool identified for it.). From the Table, we can see that decision tools that are configured not only include new developments within the VIP-DS (such as Resource Reasoning engine and Change Prediction and Management tool), but also commercial tools (such as Decision Explorer for Task 1) and legacy systems (such as Designer for Task 5). Four types of decision makers participated in the role play exercises. They are Project Managers from Primary Contractor, Joint Heads of Support Solution from MoD (Ministry of Defence), Finance Controller from MoD, and Production Engineers from Company A. The Primary Contractor, MoD and Company A have their respective positions in the ship maintenance network, their own decision goals but with a mutual aim, i.e. to get the damaged ship back to service in time. Primary Contractor’ primary purpose is to facilitate and manage the Support Solution package in conjunction with

the MoD. But it is MoD who own and operate the ship. Company A specializes in maintaining vessels for the MoD.

Task	Tool name	Tool type/nature	User role	User location
T1	Decision Explorer	Commercial tool	Project Manager, Joint Head	Primary Contractor, MoD
T2	CPM tool	Analysis tool, new development within VIP-DS	Production Engineer	Company A
T3	DM tool	Modelling tool, new development within VIP-DS	Project Manager, Joint Head	Primary Contractor, MoD
T4	RR engine	Search engine, new development within VIP-DS	Production Engineer	Company A
T5	Designer	Legacy system, developed from previous projects	Financial Controller Project Manager	MoD, Primary Contractor
T6	CPM tool	Analysis tool, new development within VIP-DS	Production Engineer, Project Manger	MoD, Primary Contractor

Table 1 Tools and users integrated within the collaborative process

Within VIP-DS, the creation of the decision making processes, capturing the decision requirements, configuration of the identified tools, allocation of decision makers, and definition of information flow through the processes for the above case study have been completed in very short time period. It is less than one hour if conducted by experienced users of the VIP-DS. It is however not a trivial job to define the decision making process and to identify the most appropriate tools for specified tasks. It requires considerable domain knowledge and experience of software packages. In the ship maintenance Support Solution coalition, it is often clear who would make decisions on what matters and with what authority, so it is not difficult to allocate decision makers to decision tasks. After the configuration is completed and if the users have defined the execution mode as automatic, the whole decision making process will be enacted automatically following the process co-ordination and synchronisation rules. Corresponding information input and output required was directed to the right tasks at the right time for the right decision makers to enable them to make the right decisions.

It is worth mentioning that if a new tool is identified for a specific decision maker to work on a specific task, and previously configured tools are not needed for that task any more, VIP-DS provides a mechanism within the tool wrapper to allow users to easily remove previous configurations and add new tools through the standard user interface. Figure 10 shows a screen shot of the interface. Users just need to select the tools that they wish to be de-coupled from the process, and then click on the “remove” button. VIP-DS will automatically handle the button event, and delete the link between the tools and the task. To add new tools, users just need to click the “Add

Tool” button and reconfigure required tools as necessary. The interface is user friendly, and the whole re-configuration process does not require users to have substantial knowledge about the VIP-DS.

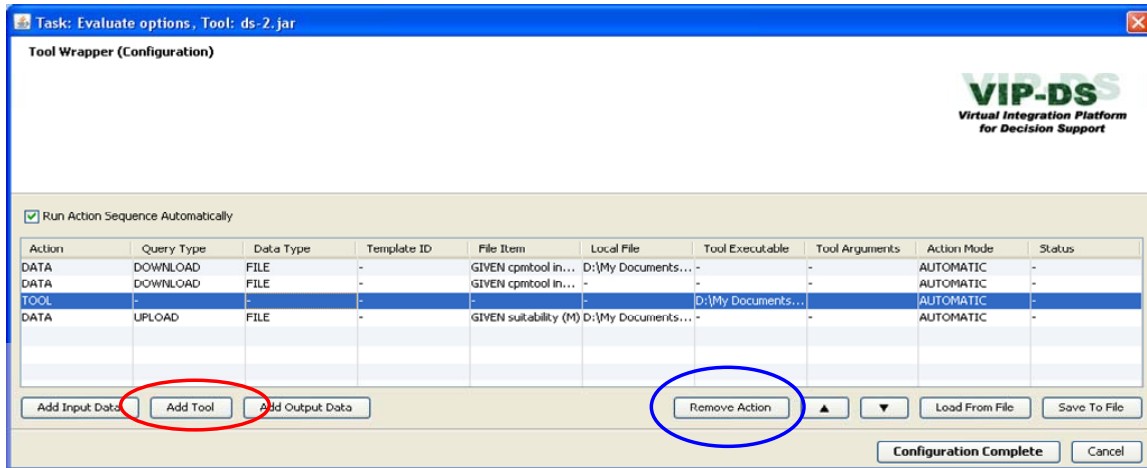


Figure 10 User interface for re-configuration of tools

The case study demonstrates that VIP-DS provides fast response to the decision making situation by creation and management of appropriate business processes, configuring and re-configuring required decision tools, allocation of decision makers to execute the processes, and provision of integrated information for the decisions. When decision situations change, a different process can be defined. Users just need to use the same approach to configure decision tools and to allocate maybe different decision makers for the new process. Therefore, VIP-DS can readily support varied decision applications to respond to dynamic decision settings.

6 Discussion and conclusions

This paper has proposed an Integration Decision Support Environment (IDSE) based on a study of DSS evolution and challenges of decision making in modern organizations. The key features of the IDSE which distinguishes itself from a traditional DSS can be summarised as:

- (1) IDSE users can configure and re-configure its functions to support varied applications, i.e. varied decisions under varied situations for varied decision makers. It differs from traditional DSS that normally support specific applications with specific decisions under specific situations for a single decision maker working on a stand-alone computer.
- (2) IDSE consists of more functional components than a traditional DSS. In addition to the basic components of a database management system, a model base management system and a user interaction system, the IDSE also has a decision management component, a resource reasoning component, a change prediction and management component, and a collaborative working component. These four additional components empower the IDSE with extra functionality that can manage decision hierarchy, reason the right resources for decisions based on ontology

mapping, predict changes and propagation path, as well as team interaction and collaboration.

- (3) The combined use of a tight integration and loose integration approach across multi-tiers within IDSE provides a balance between the integration reliability and flexibility. The processes are co-ordinated and synchronised, and information flowing through decision processes and presented to decision makers working on the process is seamlessly integrated and consistent.

For clarity, we would like to re-emphasise that this paper presents a software platform rather than a specific decision support application case or domain. That is, the problem space is defined and specified by the user and the platform provides the means to support the decisions within the area. For instance, similar to Microsoft's Word and Excel packages, or general database packages, the actual application and context is defined and set by the user.

Application of the IDSE prototype – VIP-DS (Virtual Integration Platform for Decision Support) – has been so far studied within the structured decision context. Its appropriateness for supporting ill-structured and non-structured decisions is yet to be investigated with further case studies. Immediate future work has been identified as to exploit the platform in industrial business situations, and validate the value of the VIP-DS by comparing it with other decision support systems that have already been used within industrial partners. An improved version of the VIP-DS needs to be developed based on the exploitation results. In the longer term, further work will be on new additional components to expand IDSE functionality to support global decision making (Liu & Young, 2004). In the meantime, exploration on new integration mechanisms including Web Services and Java EE technology will be undertaken to enable global communication and more robust performance.

Acknowledgement

The research reported in this paper was undertaken at the University of Strathclyde, UK. It was funded by both Primary Contractor and UK Engineering and Physical Science Research Council (EPSRC) under grant number EP/D505461/1 for project "Network Enabled Capability Through Innovative Systems Engineering (NECTISE)". The authors would also like to thank Bath University, Cambridge University and Loughborough University for their contribution to the development of the VIP-DS.

References

- Anson, R., Bostrom, R.P. & Wynne, B.E. (1995). An experiment assessing group system and facilitator effects on meeting outcomes. *Management Science* 41(2): 189-208.
- Bosch, J. (2000). *Design and Use of Software Architecture: Adopting and Evolving a Product-line Approach*. Addison-Wesley, Harlow, England.
- Boyle, I.M., Carrigan, N., Keller, R., Liao, Z., Liu, S. & Whitfield, R.I. (2007). Report on Decision Support Topic Year 1 demonstrator. NECTISE project report: NEC-DS-D-01.2.
- Boyle, I.M., Duffy, A.H.B., Whitfield, R.I. & Liu, S. (2008). An introductory note to software deliverable: decision support module. NECTISE project report: NEC-DS-D-2.6.

- Carlsson, C. & Turban, E. (2002). DSS: directions for the next decade. *Decision Support Systems* 33: 105-110.
- Bangeman, T., Rebeuf, X., Reboul, D., Schuzle, A., Szymanski, J., Thomesse, J.P., Thron, M. & Zerhouni, N., 2006. PROTEUS – creating distributed maintenance systems through an integration platform. *Computers in Industry* 57: 539-551.
- Carter, G.M., Murray, M.P., Walker, R.G. & Walker, W.E. (1992). *Building Organizational Decision Support Systems*. Academic Press Inc. Boston, USA.
- Carrigan, N. & Johnson, P. (2008). Initial design principles and guidelines for supporting collaborative decision-making. NECTISE project deliverable: NEC-DS-D-4.4.
- Chung, P.W.H. & Liao, Z. (2008). Cross-Organization Dataspace (COD) – Architecture and Implmentation. Accepted by International Conference on Computer Science and Softwarre Engineering (CSSE08). Beijing, China. December 2008.
- DeSanctis, G. & Gallupe, R.B. (1987). A foundation for the study of group decision support systems. *Management Science* 33(5): 589-609.
- Elam, J.J. & Leidner, D.G. (1995). EIS adoption, use and impact: the executive perspective. *Decision Support Systems* 14(2): 89-103.
- Eom, S.B. (1999). Decision support systems research: current state and trends. *Industrial Management and Data Systems* 99(5): 213-220.
- Gachet, A. & Haettenschwiler, P. (2003). A Decentralized Approach to Distributed DSS. *Journal of Decision Systems* 12(2): 141-158.
- Gardener, S. & Filipelli, L. (2001). A virtual collaboration testbed for joint campaign battle management and mission planning. *IEEE Aerospace Conference Proceedings*. pp. 3507-3515.
- Hemant, K., Sridhar S & Herrick C, (1999). Beyond spreadsheets: tools for building decision support systems. *Computer*, March 1999.
- Holsapple, C.W. (2008). *Decision Support Systems: Foundations and Variations*. Wiley Encyclopedia of Computer Science and Engineering.
- Hopple, G.W., (1988). *The state of the art in decision support systems*. Published by QED Information Sciences Inc. ISBN 0-89435-247-4.
- Humphreys, P. & Bekerley, D. (1985). Handling uncertainty: Levels of analysis of decision problems. In G. Wright (Ed.), *Behavioural Decision Making*. London: Plenum Press.
- Jaganathan, S., Erinjeri, J. J. & Ker, J. I. (2007). Fuzzy analytical hierarchy process based decision support systems to select and evaluate new manufacturing technologies. *International Journal of Advanced Manufacturing Technology* 32(11-12): 1253-1262.
- Johnson, P., Wilson, S., Markopoulous, P., & PycocK, J. (1993). ADEPT – Advanced Design Environment for Prototyping with Task Models. Paper presented at the INTERCHI'93, Amsterdam, Holland.
- Keller, R. & Clarkson, P.J. (2008). Change prediction prototype. NECTISE project deliverable NEC-DS-D-3.6.
- Knudtson, M.L. (2005). Emergent decision making: defining the patient perspective. *Canadian Journal of Cardiology* 21(5): 433-434.
- Kosaka, T. (1994). The first investigation of executive information systems practices in Japanese firms. *Proceedings of the TIMS/ORMA Conference*, Boston.
- Lee, D., Newman, P. & Price, R., (1999). *Decision Making in Organizations*. Published by Financial Times Management. ISBN: 0-273-631136.

- Limayem, M., Banerjee, P. & Ma, L., (2006). Impact of GDSS: opening the black box. *Decision Support Systems* 42: 945-957.
- Linthicum, D.S. (2004). *Next Generation Application Integration: from Simple Information to Web Services*. Addison-Wesley, Boston.
- Liu, S., Duffy, A.H.B., Whitfield, R.I. & Boyle, I.M. (2009). Integration of decision support systems to improve decision support performance. *Knowledge and Information Systems – An International Journal*. DOI 10.1007/s10115-009-0192-4.
- Liu, S., McMahon, C.A., Darlington, M.J., Culley, S.J. & Wild, P.J. (2006). A computational framework for retrieval of document fragments based on decomposition schemes in engineering information management 20(4): 401-413.
- Liu, S. & Young, R.I.M. (2004). Utilising information and knowledge models to support global manufacturing co-ordination decisions. *International Journal of Computer Integrated Manufacturing* 17(6): 479-492.
- Oaks, S. & Wong, H. (2004). *Understanding and Mastering Concurrent Programming: Java Thread*. O'Reilly, Cambridge.
- Power, D.J. (2003). A brief history of decision support systems. DSS Resource.COM, available at <http://DSSResources.COM/history/dsshistory.html>
- Respicio, A. & Captivo, M.E. (2008). Marketing-production interface through an integrated DSS. *Journal of Decision Systems* 17(1): 119-132.
- Sen, T.K., Moore, L.J. & Hess, T.J. (2000). An organizational decision support system for managing the DOE hazardous waste cleanup program. *Decision Support Systems* 29: 89-109.
- Shi, Z., Huang, Y., He, Q., Xu, L., Liu, S., Qin, L., Jia, Z., Huang, H. & Zhao, L. (2007). MSMiner – a developing platform for OLAP. *Decision Support Systems* 42: 2016-2028.
- Shim, J.P., Warkentin, M., Courtney, J.F., Power, D.J., Sharda, R. & Carlsson, C. (2002). Past, present, and future of decision support technology. *Decision Support Systems* 33: 111-126.
- Sprague, R.H. & Watson, H.J., (1986). *Decision Support Systems: Putting Theory into Practice*. Prentice-Hall International Editions. ISBN: 0-13-197766-0.
- Szykman, S., Fennes, S.J., Keirouz, W. & Shooter, S.B. (2001). A foundation for interoperability in next generation product development systems. *Computer-Aided Design* 33: 545-559.
- Taghezout, N. & Zarate, P. (2008). Negotiation process for multi-agent DSS for manufacturing system. In *Proceedings of International Conference on Collaborative Decision Making: Collaborative Decision Making – Perspectives and Challenges*. Edited by Zarate P, Belaud JP, Camilleri G and Ravat F. IOS Press. ISSN: 0922-6389.
- Vahidov, R. & Kersten, G.E. (2004). Decision station: situating decision support systems. *Decision Support Systems* 38: 283-303.
- Whitfield, R.I., Duffy, A.H.B., Coates, G. & Hills, W., (2002). Distributed design co-ordination. *Research in Engineering Design* 13: 243-252.
- Whitfield, R.I., Duffy, A.H.B., Thomson, A., Wu, Z. & Liu, S. (2007). An architecture for organizational decision support. *Systems Engineering Conference (SyEC'07)*. Loughborough, February 12 – 13, 2007.
- Wu, Z., Duffy, A.H.B. & Whitfield, R.I. (2007). Virtue integrated platform: holistic support for distributed ship hydrodynamic design. 16th International Conference on Engineering Design (ICED07), Paris, France, 28-31 August 2007.

Zeydan, M. & Colpan, C. (2009). A new decision support system for performance measurement using combined fuzzy TOPSIS/DEA approach. *International Journal of Production Research* 47(15): 4327-4349.