INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

DECISION-FOCUSSED RESOURCE MODELLING FOR DESIGN DECISION SUPPORT

Shaofeng Liu[1], Alex Duffy[2], Robert Ian Whitfield[2], Iain Boyle[2], Wenjuan Wang[2] and Khaled Mohamed[2]
[1]University of Plymouth, Plymouth, UK [2]University of Strathclyde, Glasgow, UK

ABSTRACT
Resource management including resource allocation, levelling, configuration and monitoring has been recognised as critical to design decision making. It has received increasing research interests in recent years. Different definitions, models and systems have been developed and published in literature. One common issue with existing research is that the resource modelling has focussed on the information view of resources. A few acknowledged the importance of resource capability to design management, but none has addressed the evaluation analysis of resource fitness to effectively support design decisions. This paper proposes a decision-focused resource model framework that addresses the combination of resource evaluation with resource information from multiple perspectives. A resource management system constructed on the resource model framework can provide functions for design engineers to efficiently search and retrieve the best fit resources (based on the evaluation results) to meet decision requirements. Thus, the system has the potential to provide improved decision making performance compared with existing resource management systems.

Keywords: decision-based design, resource model framework, design decision support, UML, EJB

1. INTRODUCTION
There has been a growing recognition that decisions are a fundamental construct in engineering design since 1990s, which resulted in the emergence of decision-based design (DBD). DBD is an approach to engineering design that recognises the substantial role that decisions play in design and other engineering activities, largely characterised by ambiguity, uncertainty, risk and trade-offs [1]. The importance of decisions in design has been demonstrated through many aspects including decision formulation, analysis, evaluation and solutions. Key decisions required to make in design are not only on all areas related to products, design processes and design teams, but also related to organisation and project management [2]. Design decision making is an important but a complex task. This paper is concerned with decision support for design through an enhanced resource model framework.

Resource management has been identified as a crucial task within design management focussing on organising and controlling resources to enable their continuous optimised utilisation throughout a changeable design development process [3]. Research in design resource management has addressed wide ranges of issues covering resource allocation, levelling, configuration, monitoring and integration [4-6]. Various resource definitions and models have been developed to promote the understanding of resource management in design [7]. Most of existing research investigated the issue from the resource information viewpoint, a few considered the need for resource capability analysis, but none focussed on the decision-based design requirements of resource management. This paper proposes a decision-focussed resource model framework (DFRMF) aiming to provide better understanding of design resources from both decision analysis and information viewpoints with a user-focused purpose. The DFRMF has been discussed through the development of a set of computational models with UML (Unified Modelling Language). The application of the improved resource model framework has been demonstrated through the construction of a resource management software system, which can be used to automate and assist the process of decision making under complex design situations.

2. RELATED WORK
A decision-focussed resource model differs from existing resource models in that it is designed and developed to support decision making, and therefore it provides not only the information about

resources but also provides evaluation on the resource capability and fitness to design tasks, so that designers can make informed and rationalised decisions on resource allocation, levelling and configuration etc. Most existing research on resource definitions, models and systems are built upon the understanding of processes and activities. The transformation model shown in Figure 1(a) has been widely adopted in generic operations management [8]. All the inputs and some of the outputs (for example produced goods) are considered as resources. Typically those inputs which are used up in creating goods and services are defined as transformed resources, and those play a part in the creation process but not used up as transforming resources. This transformation model was expanded by Duffy and applied to design activities as illustrated in Figure 1(b) [9]. This latter model takes an activity as its central interest rather than a transformation process. An activity is taken to be a physical or cognitive action that creates an outcome, and the author explicitly stated that "an activity is carried out by a resource of some kind". Compared with the transformation model, the activity model distinguishes resources and goals from inputs and outputs of a design activity. Resources were defined as means to carry out the activity while the other inputs provide the conditions or elements upon which the means act. That is, the resources facilitate the activity whereas the inputs and goals are used in the activity. This to some extend reflects the concepts of transformed and transforming resources in the transformation model. The performance of resources to undertake activities was further investigated through the definition of an E^2 model, i.e. efficiency and effectiveness [7]. In Haffey's work, efficiency was defined as the relationship (often expressed as a ratio) between what has been gained (outputs minus inputs) and the level of resources used, as shown in Figure 1(c). On the other hand, the effectiveness is determined by the relationship between outputs and goals, as shown in Figure 1(d).

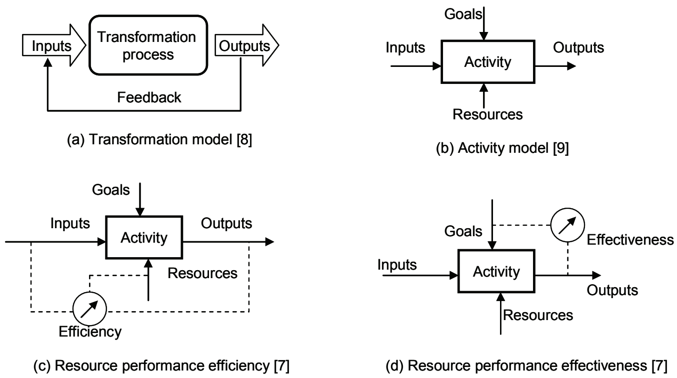

Figure 1 Various process/activity-based resource definitions and models

Real-time resource management within a distributed collaborative design environment was discussed in [3]. Coates et al developed a Design Co-ordination System (DCS), within which a Resource Model (RM) was defined to hold available resource information and knowledge. In the DCS, the RM was directly accessed by a Resource Manager, an agent who is primarily responsible for maintaining accurate knowledge of all available resources within the RM. There was also a Resource Monitor, another agent who is responsible for sensing, forecasting and reporting resource performance efficiency. The interactions between the RM, Resource Manager and Resource Monitor are illustrated in Figure 2.

A resource model was created by Whitfield et al as a component of a process control tool for a Virtual Integration Platform (VIP), which is a software environment to support the management of distributed design [10]. In their work, the VIP users (including designers and project managers) are considered as resources. Compared with previous work, this resource model for the first time explicitly identified the resource capability (the measured ability to perform an activity) as an integral part of the resource. But the evaluation of the resource capability is limited to designer's commitments and being as project managers.

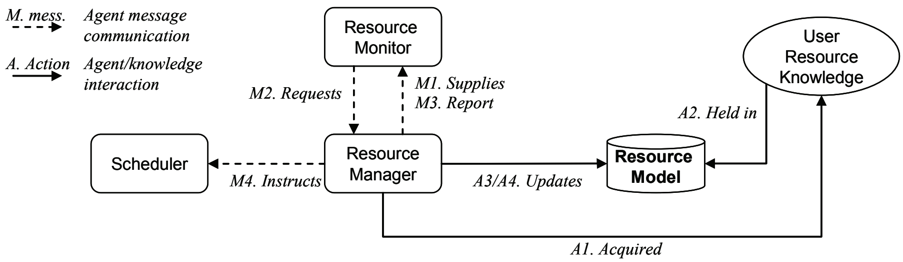

Figure 2 Resource Model in the Design Co-ordination System [3]

Extending the capability definition to a network enabled environment, Wang et al's work defined two types of resource capability [11], i.e. resource's networked capability (RNC) and networked resources' capability (NRC). RNC is the capability possessed by a resource in a networked environment within which a set of resources are linked via various types of relationships. NRC is the holistic capability derived from a set of resources in such environment. RNC and NRC have been used in deploying resources aiming to maximising resource utilisation in the VIP.

The DFRMF defined in this paper has taken in the strengths from previous resource models, especially from [3], [10] and [11], i.e. a combination consideration of resource basic information, its performance and resource capability to undertake activities. But the evaluation of the resource capability has been undertaken in more depth and therefore the enhanced DFRMF can provide better support for more informed and rationalised decision making. This is one feature of the DFRMF which distinguishes itself from previous work.

Another feature of the DFRMF is that even though it is developed based on the existing resource models that aim to provide the resource information, DFRMF goes one step further, which defines attributes and operations based on systematic design approaches with formal computational modelling language UML (Unified Modelling Language), and it is able to be mapped to database implementation such as through EJB's Java Persistence API (JPA). This provides the potential for decision makers to use the power of computer systems for decision analysis and consistent information provision to support decision making, as pointed out in the literature that humans operate within the limits of what has been referred to as "bounded rationality", that is, limitations to their ability of processing information usually keep them from reaching optimal decisions [8]. This limitation can be remedied through using resource software systems that are constructed on computational models.

3. A DECISION-FOCUSSED RESOURCE MODEL FRAMEWORK (DFRMF)

This section discusses the DFRMF which comprises a systematic approach taken and a series of models developed to define resources from multiple perspectives.

3.1 A systematic approach to resource modelling

There have been resource models developed in the literature for design management, but different researchers explored different approaches at ad hoc basis, and there has been no consensus on which methods perform better than others. In the past this was not recognised as an issue when designers worked on their own design tasks independently. It is now however an issue that cannot be ignored or overlooked in the distributed and collaborative design environment, especially on large design projects when networked virtual design teams are involved. In modern collaborative design, complicated resource information and behaviour are required for design decision making, many of the design activities (such as design parameter optimisation, costing and simulation analysis) often deal with different perspectives of design resources. There is a need to facilitate the intention and reasoning to integrate the resources across design teams. Therefore, a systematic approach with unified representation of the resource models that can be correctly and easily interpreted by different designers, stakeholders and software agents is essential to meet the collaboration need. This paper discusses object-oriented design and UML for the development of the resource model framework. It has been shown that object-oriented design simplifies the task of modelling complicated information and behaviour [12]. Figure 3 illustrates the systematic approach taken for the development of the resource model framework.

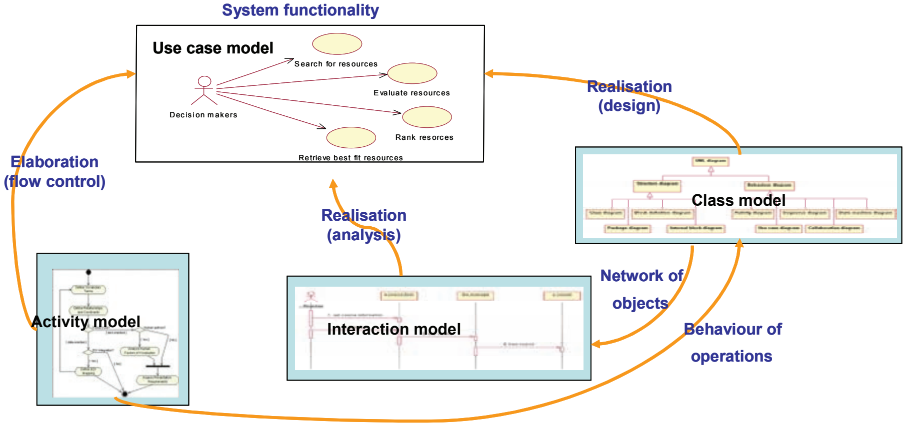

Figure 3 The systematic approach to developing the DFRMF

From the Figure 3, four types of interdependent models have been identified to represent multiple perspectives of the resources. These include a use case model (function perspective), an activity model (flow control analysis perspective), an interaction model (sequence behaviour analysis perspective) and a class model (structure perspective). Their relationships defined in the Figure show how the models work together to form a comprehensive understanding of resources and support integration between different perspectives of the resources. The use case model defines how the resource model framework to be used, that is, when the DFRMF is used to construct a computational resource management system, what functions the system can provide to users. From an external viewpoint or user's viewpoint, the use case model is the most important that will dictate the design and development of other models. For DFRMF, four main use cases functions have been defined as (1) to search resources; (2) to evaluate resources; (3) to rank resources; and (4) to retrieve best fit resources. The following sub-sections will discuss the class model, activity model and interaction model that are developed to fulfil the functions defined by the use case model.

3.2 Resource information model – class diagram

Class model provides a means to define the necessary information classes and the relationships between the classes in a domain. In object-oriented design, an information class is an abstraction of objects that interact with and collaborate to each other to realise functions. For each class, UML provides a standard representation which consists of three compartments: the top compartment for the name of the class, middle one for attributes and the bottom one for operations that can be imposed using the attribute information.

The definition of the class diagram has been undertaken through three major steps:

- identification of the main classes for resources;
- specifying the relationships between the classes; and
- definition of attributes and operations for each class.

Identification of main classes has been carried out through the study of types of resources based on literature, ongoing research projects and authors' own experience in the area. Initial investigation has identified five main types of resources:

- human resource: these are people including expertise and skills people have;
- material resource: such as facility, equipment, machine, power, water etc;
- informational resource: including standards, regulations and procedures;
- technical resource: for example, methodology, technical tools such as Rational Rose.
- financial resource: money.

To specify the relationships between classes, internal and external links between the types of resources have been examined. Three main categories of links have been identified between the different types of resources:

- association: general link, for example people use computers;
- composition/ aggregation: link between “whole” and “part”, such as a car has wheels, body and chassis etc.
- abstraction/ specialisation/ generalisation/ inheritance: link between parent and children, for instance staff can be designer engineers, project managers, or sales staff.

Figure 4 shows the top level of the class diagram of resources in UML representation.

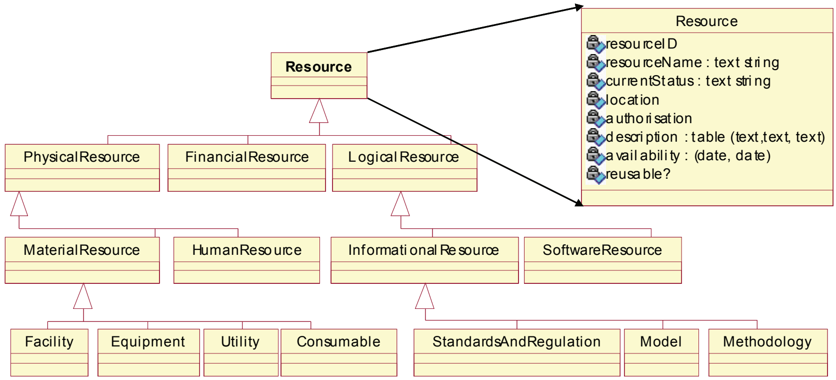

Figure 4 A class diagram of resource information model

Finally, attributes of resources have been defined, including generic properties that belong to all types of resources and specific properties that only belong to some types of resources. Examples of generic attributes are ID, name, status, availability, location, etc. Specific attributes include roles (for human resource), maintenance (for equipment), modified date (for standards), version (for regulations). Figure 4 shows an example, i.e. the *Resource* class, of how attributes are captured and represented with UML.

For clarity, the class diagram in Figure 4 only shows the main classes of resources without attributes of the classes. It also leaves out information of operations of the classes and low level classes. The class hierarchy however clearly shows how the resource information is organised in the model, and ready to be transformed into computer systems for the construction of a resource management system.

3.3 Resource evaluation model – activity diagram

It is important that resource fitness to design activities and decisions be assessed before a resource is actually allocated to design tasks to ensure resource performance efficiency and effectiveness [7]. The criteria to assess the resource fitness should be the resource request requirements coming from the design process and decisions. When there are a large number of candidate resources available, and there is a wide range of resource configurations (combinations of resources) potentially possible, the evaluation process is complicated and can only be done with the facilitation from computer systems. A computer-based resource management system can evaluate each resource configuration against multi-criteria and make sure that only the best matched or fittest resources will be returned to decision makers or process planners. Figure 5 is a simplified UML activity diagram which shows how a Resource Manager (a computerised agent) receives the resource request requirements from the decision makers and process planners, evaluate the resource fitness, and then retrieve the resources that are best mapped onto the activities and decision requirements.

After the resource request requirements are received from the decision makers and process planners, the Resource Manager will first identify the attributes (identified in previous section and modelled in the class diagram) that can contribute to meet the request requirements. Then the evaluation consists of three parallel sub-processes: ranking the resource, check the availability, and check the affordability. Resources are ranked against both resource appropriateness and performance. Resource availability is evaluated against a combination of related attributes such as available dates (work plan Gantt chart), location and authorisation. Affordability is mainly evaluated against the budget. In the end, only the resources that are affordable, available and on top of the ranking list will be returned to process planners and recommended to decision makers, as the approved fittest resources. On some extreme occasions, if no results are returned, it may need to adjust or re-define the resource request requirements so that the nearest matches can be found and retrieved by the system.

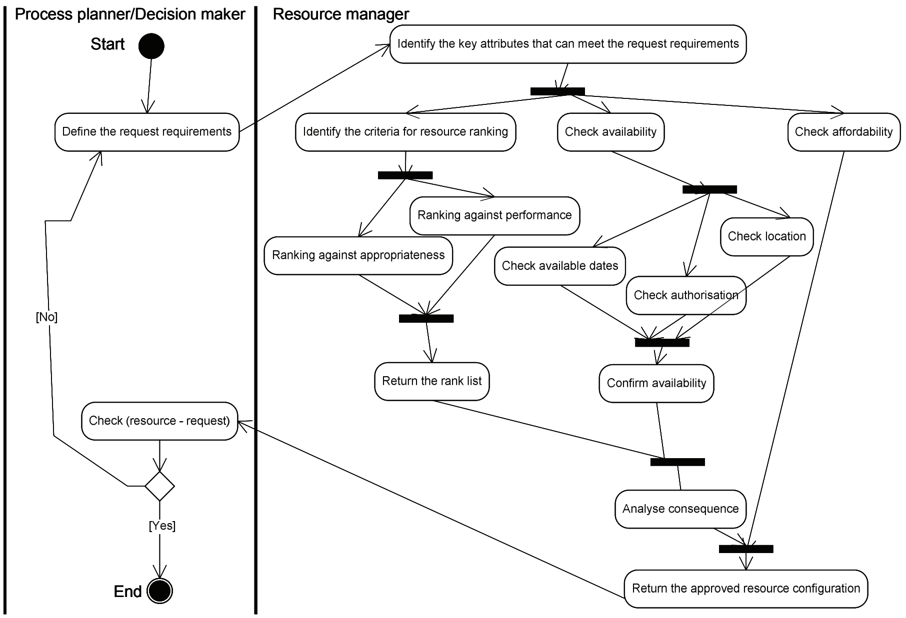

Figure 5 An activity diagram for resource evaluation model

3.4 Resource interaction model – sequence diagram

The success of sharing resource information and capability depends on many factors such as trust and economic value of resource information. However, once the participating members in a virtual design team are willing to share information, it is technically crucial to design common structure to be able to capture and retrieve the right resource information (class diagram discussed in Section 3.2), to define processes to evaluate the resource capability (activity diagram as discussed in Section 3.3), and to make them available for decision makers to make good decisions [13]. In object-oriented systems, information is represented by many information classes and their object instances. System performance is achieved through the collaboration of the objects in the system, which is often referred to as one object sending a message to another object. It is information interaction that provides the conduit for object interaction and collaboration. This sub-section discusses the interactions between different resource objects and agents so that the right information will be served to the right decision makers (through agents) at the right time.

Figure 6 presents the interactions between resource objects and agents with UML sequence diagram. In the diagram, resource objects (physical, software, finance and informational) and agents (resource manager and process planner) are illustrated with a rectangle box including object names underlined, as shown on the top. Messages flowing between the objects and agents to trigger the interactions are labelled on the respective arrows. The time sequences and lengths are also shown through the timeline with vertical bars.

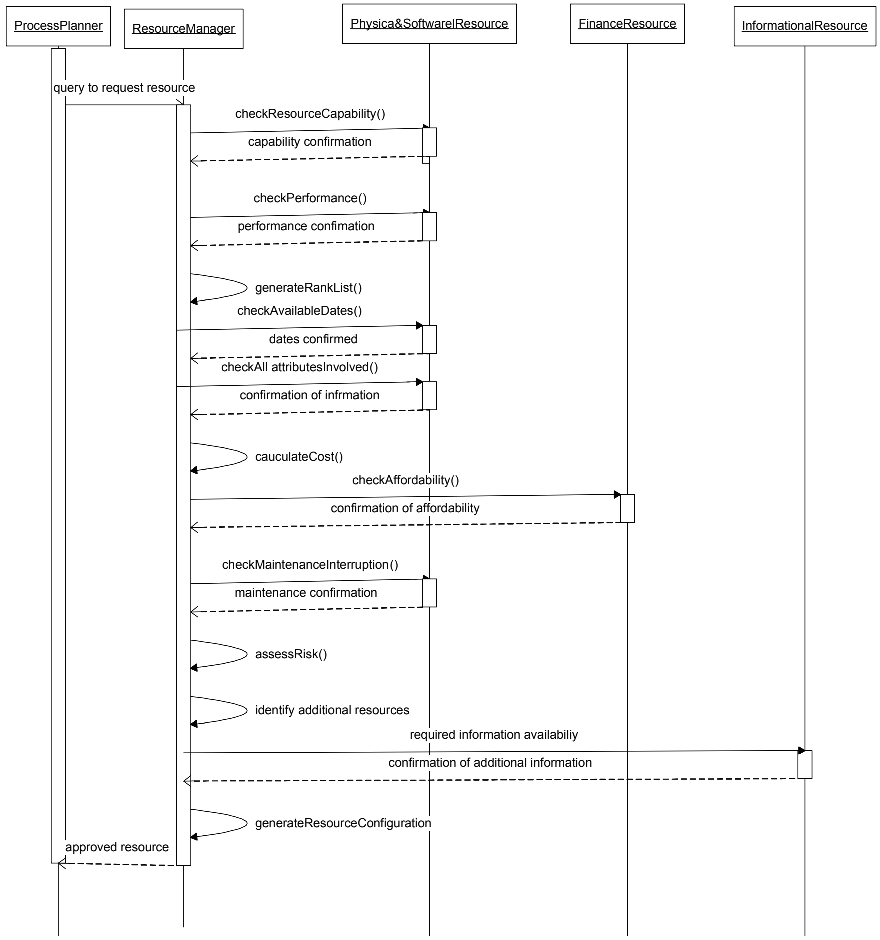

Figure 6 A sequence diagram for resource interaction model

4. APPLICATION OF THE RESOURCE MODEL FRAMEWORK TO CREATION OF A RESOURCE MANAGEMENT SYSTEM

This section discusses the application of the resource model framework DFRMF to the construction of a computer-based Resource Management System (RMS), which can be used by collaborative design teams to search for and retrieve the "right" resources for the design tasks. The technology support for the implementation of the RMS is discussed first, followed by the object-entity mapping, and finally the design of the system user interface.

4.1 EJB infrastructure for system adaptability, portability and extensibility

It is essential that the right technology is chosen and employed for the implementation of the RMS so that the integrity of the resource models developed within the DFRMF can be maintained, and that the RMS can be reliable, extensible and scalable. Based on the above consideration, the state-of-the-art Java technology - Enterprise JavaBeans (EJB) – has been chosen as the programming platform and infrastructure for the development of the RMS. Figure 7 highlights how EJB layers seamlessly address the three views of the resource model framework.

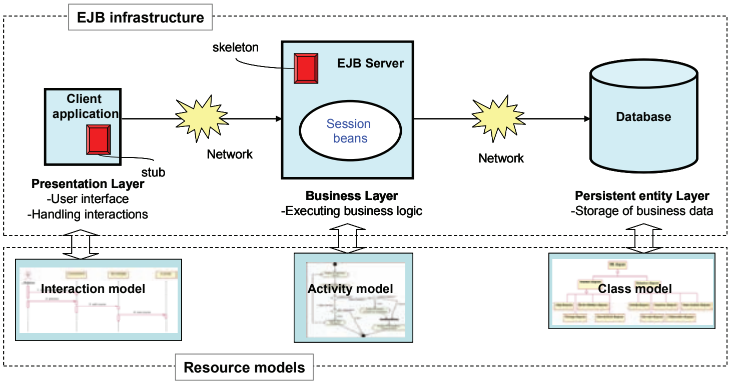

Figure 7 EJB for the RMS implementation

EJB has been widely renowned for its creative design of three separate layers, i.e. a presentation layer, a business layer and a data layer (persistent entity). Separation of business logic from resource and lifecycle management allows system developers to concentrate on business logic (value added) rather than infrastructure issues. In terms of the RMS based on the DFRMF discussed in Section 3, three models address three important views of resource modeling: a class model addressing information (data) structure, an activity model addressing business logic view, and a sequence model addressing interaction view. These three models and views can be directly mapped onto the three layers of the EJB infrastructure. Therefore EJB is considered as the appropriate technology for the implementation of the DFRMF to develop RMS. Key advantages of using EJB infrastructure for the RMS include:

- adaptability: As EJB is a ubiquitous industrial standard, it will allow RMS to interoperate and be integrated with other enterprise applications built upon the standard.
- portability: Since EJB is a platform independent standard, it allows RMS to be ported from one platform to another without heavy cost.
- extensibility: The EJB specification is published and available freely to all, future extension and evolution of the RMS will not be restricted by long-term viability and proprietary architecture of a single vendor.

4.2 Object-entity mapping for resource persistence and searchability

In Section 3, resources have been modelled as classes and they are instantiated as objects. The resource information can be stored in two types: persistent and transient. It is advantageous for resource information to be stored and maintained as persistent objects because the system will not lose information after the programme is closed down, and it does not require re-populate for next run. In RMS, persistent objects are used, which also facilitates to maintain the consistency of resource information. It is however not a simple decision on how objects should be persisted because system performance is closely dependent on it. The simplest way to persist objects in Java is to use Java's native serialisation API that lets you to write objects to files. For RMS data, it is insufficient as resource data persisted in this way is not efficiently searchable, nor is concurrent access protected by transactions. One key requirement of the RMS is that the resources can be efficiently searched, and concurrent access to resource data be protected by transactions to avoid information inconsistency. Therefore, resource information is stored as persistent objects using a relational database management system MySQL. To implement this, it is essential to understand the mapping of objects to relational databases. EJB object-relational mapping technology is used for the implementation of the RMS.

Technically speaking, object-relational mapping is the act of converting and un-converting in-memory objects to relational data. EJB provides an object-relational mapping engine to allow us to map a Java class to a SQL table definition. For example, as shown in Figure 8, an instance of a *Resource* class is mapped to a row in a Resource table, while attributes of the *Resource* instances are mapped to individual columns in the table. Using EJB object-relational mapping for persisting resource information in the RMS rather than the simple object serialisation offered by the Java language, is not because it is a much more complicated mechanism of persisting objects, but because it offers advantages to the RMS including:

- By decomposing resource objects as relational data, users can issue arbitrary queries for resource information. For example, users can search through all the database records of resources that have specific capabilities and commitments, and retrieve only those resource objects that fulfil the query;
- More advanced queries can be easily formulated by using advanced SQL grammar such as through the combination use of AND, OR and XOR operations;
- Users can visually inspect the database data of resources (refer to the following Section about RMS user interface). Data are not stored as bit-blobs, which is great for debugging and auditing.

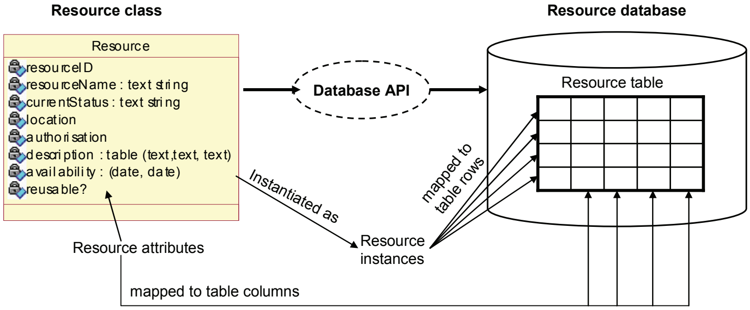

Figure 8 Mapping resource objects to relational databases

4.3 RMS graphical interface for usability

To make the RMS user friendly, easy for decision makers to search and retrieve the best fit resources for design tasks, the user interface design is based on the following three principles:

(1) Compact: As shown in Figure 9, the user interface window includes four main areas: top left is resource hierarch area that users can expand and collapse to browse through all categories and sub-categories of resources. This hierarchy of resources reflects the class structure defined in Section 3.2. Bottom left is the views area, which allows users to switch between different focus on resources. For example, a project view focuses on how well resources are matched on project work plan such as a Gantt Chart, a process view focuses on how well resources are matched on process, tasks and activities. The above information is helpful to make decisions on allocating resources to tasks, and organise resources to undertake tasks according to the project schedule. The area on top right is a graphical view area which vendors the different views of the resources. In the Figure, it is a screenshot of the *Process view* (selected by user and highlighted in the Figure) of the selected resource type (*Equipment* as highlighted). Design tasks are listed on its left side, vertically; resources (identified by IDs) are listed on its top part horizontally. Different colours in the middle represent the fitness level of the resources to perform the tasks. The area at the bottom right is a table area to display resource detail information as stored in the database. The above four areas together provide a compact display of both resource types and detail information of specific resource objects, with different views of resource suitability based on the resource evaluation model defined from Section 3.3

(2) Visualisation: Three types of visualisation techniques have been employed in the design of RMS user interface: a tree structure of the resource classification, a graphical representation for different views on selected resource types, and a table to show detail information of specific resource objects. Tree structure display of resource information provides users an easy way to search for required resources merely by browsing through the hierarchy. It is a well-structured means, which gives users guidelines to reach the results step by step, and users always know where they are in the hierarchy. After they use the system a few times, they can accumulate experience and will reach the optimal searching results more and more quickly. The graphical representation of the different views of resource types offers users a unique view on the distribution of resource fitness

level against the chosen evaluation criteria. What users get is a more complete picture of many resource candidates that may have the capability to perform the same design tasks but with different level of competence. Finally, the table representation of detail information of resource objects shows comparison between the objects according to different attributes (presented as table columns). This is useful when users wish to rank the resource fitness against different criteria. All the users need to do is to highlight a table column, then click the “Rank” toolbar, and they will get the desired ranking results.

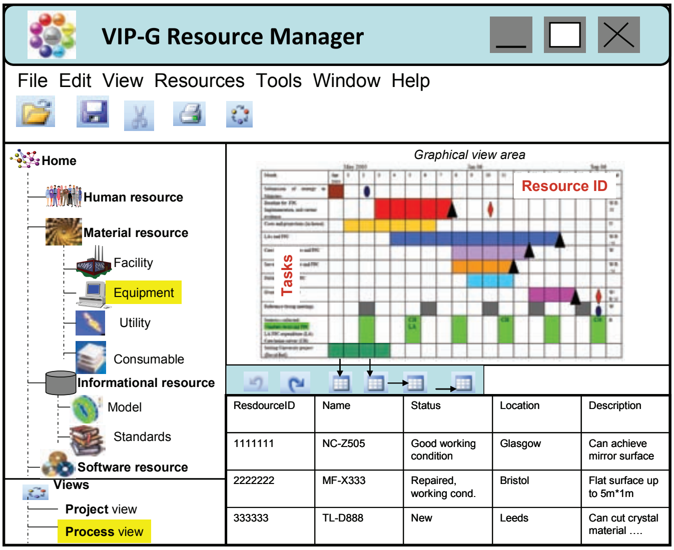

Figure 9 Graphical user interface of RMS

(3) Easy to use: RMS user interface design focuses on its usability. To ease users guessing effort which many other search engines require such as by typing in and refining keywords, RMS takes the approach to let users select information by simply clicking buttons and selecting menus. Apart from the system main menu on the top (with menu items to be navigated to, not expanded and shown in the Figure), the system provides rich toolbars (both on the top underneath the main menus for common use, and at the bottom right area just above the table for manipulating the table rows and columns). To switch between different views of the resource is also only one click away. The system is easy to use even for novice users.

5. DISCUSSION AND CONCLUSIONS

A decision-focussed resource model framework has been discussed. The framework includes a systematic approach, and a series of models which represent multiple perspectives of resources: a use case model - function perspective, an activity model - flow control perspective, a sequence model – interaction perspective, and a class model – structural perspective. These models together capture not only the information of resources, but also the capability and fitness evaluation of the resources, which is crucial to decision analysis for complex design situations. All the models have been created and represented with formal computational language UML, which are ready for the production of a resource management system. The resource management system under development is based on the state-of-the-art Java Technology – Enterprise JavaBeans (EJB) – to ensure the system’s adaptability, portability and extensibility. The systems utilises object-entity mapping to persist resource objects so that they are efficiently searchable and transaction protected.

Immediate future work is to complete the implementation of the resource management system and to test the system with real design cases to evaluate the performance of the system. In longer term, the work will investigate the integration of the resource management system into the Virtual Integration

Platform [10, 14] which can support distributed design decisions in a holistic manner, i.e. through the integration of resource management, process control and project management with decision management approaches.

ACKNOWLEDGEMENT
The research reported in this paper was undertaken at the University of Strathclyde. All the help in relation to facilities and professional guidance which enabled the work is highly appreciated. The authors also wish to thank former colleagues, Dr John Hopkins and Dr Raji Tenneti, for their contribution to the discussion of the design of an architecture for a generic virtual integration platform, from which the resource models have benefited.

REFERENCES
[1] Lewis KE, Chen W and Schmidt LC. Decision Making in Engineering Design, *Chapter 2: The Open Workshop on Decision-Based Design,* 2006 (ASME Press).
[2] Liu S, McMahon CA and Bracewell RH. A web-based semantic information retrieval system to support decision-making in collaborative design. *In Proceedings of International Conference on Engineering Design, ICED'07*. Paris, France, August 28-31, 2007.
[3] Coates G, Duffy AHB, Whitfield RI and Hills W. Engineering management: operational design co-ordination. *Journal of Engineering Design* 2004, 15(5): 433-446.
[4] Andreasen MM, Duffy AHB, MacCallum KJ, Bowen J and Storm T. The design co-ordination framework: key elements for effective product development. *Proceedings of the 1st International Engineering Design Debate, Glasgow, UK*. September 23-24 1996, pp. 151-174.
[5] Chakrabarti A and Singh V. A method for structure sharing to enhance resource effectiveness. *Journal of Engineering Design* 2007, 18(1): 73-91.
[6] Reich Y and Paz A. Managing product quality, risk, and resources through resource quality function deployment *Journal of Engineering Design* 2008, 19(3): 249-267.
[7] Haffey MKD. An approach, insights and methodology for performance improvement through process activity management. *University of Strathclyde. PhD thesis*. 2007.
[8] Dilworth JB. Operations Management. *The McGraw-Hill Companies Inc*. 1996. New York.
[9] Duffy AHB. Designing design. *Third International Seminar and Workshop on Engineering Design in Integrated Product Development*. 10th -12th October 2002. Zielona Gora, Lagow, Poland.
[10] Whitfield RI, Duffy AHB and Coates G. Real time resource scheduling within a distributed collaborative design environment. *16th International Conference on Engineering Design (ICED07)*. 28-31 August 2007, Paris, France.
[11] Wang W, Duffy AHB, Whitfield RI, Liu S and Boyle IM. A design view of capability. *Conference of Realising Network Enabled Capability*. 13-14 October 2008. Leeds, UK.
[12] Steele J, Son YJ and Wysk RA. Resource modelling for the integration of the manufacturing enterprise. *Journal of Manufacturing Systems* 2001, 19(6): 407-427.
[13] Liu S and Young RIM. Utilising information and knowledge models to support global manufacturing co-ordination decisions. *International Journal of Computer Integrated Manufacturing* 2004, 17(6): 479-492.
[14] Liu S, Duffy AHB, Whitfield RI and Boyle IM. An integrated decision support environment for organisational decision making. *International Conference on Collaborative Decision Making (CDM08)*, 1st-4th July, Toulouse, France.

Contact:
Shaofeng Liu
University of Plymouth
Business School
Drake Circus
PL4 8AA, UK
Phone: 0044 1752 585 721
Fax: 0044 1752 585 713

E-mail: shaofeng.liu@plymouth.ac.uk
URL: www.plymouth.ac.uk/staff/sliu1

Dr **Shaofeng Liu** is currently a lecturer in operations management and decision making. She worked as a Research Fellow at the University of Strathclyde with Professor Alex Duffy before she joined the University of Plymouth. She obtained her PhD from Loughborough University, UK in 2004. Her research interests include decision support systems and enterprise resource planning (ERP) systems.

Professor **Alex Duffy** is the Director of the CAD Centre within the Department of Design Manufacture and Engineering Management at the University of Strathclyde. He lectures in design management, product development, knowledge intensive CAD, advanced computational techniques, and databases. His main research interests have been the application of knowledge based techniques in design, product and domain knowledge modelling, machine learning techniques and design re-use, performance measurement, and design co-ordination. He has been the Co-ordinator and Topic Leader of a number of EU and UK funded projects, is the Editor of the Journal of Engineering Design, and on the editorial board of a number of international journals in the area of design, design computing and Artificial Intelligence in design. He is currently the vice-president, and prior to this the President, of the Design Society, an international body encompassing all aspects and disciplines of design.

Dr **Robert Ian Whitfield** is a lecturer in the Department of Design Manufacture and Engineering Management at the University of Strathclyde. His research interests include: co-ordination, collaboration, integration, resource management, process modelling and optimisation and modular design.

Dr **Khaled Mohamed** is a Research Fellow within the department of Design Manufacture and Engineering Management at the University of Strathclyde, UK. He is currently undertaking research within a number of European Union FP6 research projects.