DecMS: An Approach to Providing Decision Support within NEC delivery

Iain Boyle, Alex Duffy, Ian Whitfield, and Shaofeng Liu
CAD Centre, Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow G1 1XJ
E-mail: iain.m.boyle@strath.ac.uk

Abstract

Decision-making across the military capability lifecycle phases can vary considerably in terms of the types of decisions made and the manner in which they are made. Although decision-making has received considerable attention within the research community, much work has concentrated on providing decision support for particular styles of decision-making. However, within capability delivery there is a need to develop approaches that can both map styles of decision-making to particular decision problems, and provide decision support at an executable level of detail. This paper presents the Decision Management and Support (DecMS) approach to providing decision support during capability delivery. The approach is based upon refining a fundamental model of decision-making to an executable level of detail. Refinement is controlled using analogical reasoning to ensure that the model is refined in accordance with the needs of the decision problem at hand. Future work will involve testing the effectiveness of the approach.

Keywords: *decision support, capability delivery, capability, analogical reasoning.*

1 Introduction

A current trend within the MoD is the acquisition of flexible, ready, and rapidly deployable Armed Forces and to achieve this objective the MoD has identified Network Enabled Capability (NEC) as a potential solution [1]. This need for greater flexibility and the move towards capability-based acquisition poses significant challenges to suppliers within the defence supply chain. From an organizational perspective, the concern is to understand what role industry can play within the "Plan", "Deliver", "Support (to readiness)", "Deploy", "Support (deployed)", and "Create effect" phases of the military capability lifecycle [2] (Figure 1). Although limited in terms of the contribution that can be made within the "Create effect" phase, there is considerable scope for industry to assume a significant role within the delivery of that NEC: i.e., within the "Plan", "Deliver", "Support (to readiness)", "Deploy", and "Support (deployed)" phases.

Within any of these five NEC delivery phases there exists the potential for decision-making activities to be undertaken within an organizational context (i.e., among the multiple distributed organizations that comprise the NEC delivery supply chain). It is reasonable to assume that both across and within each of these life phases decision-making will vary in terms of:

- The type of decisions made.
- The process by which decisions are made.
- The type of information used to make.
- The knowledge used to process this information.

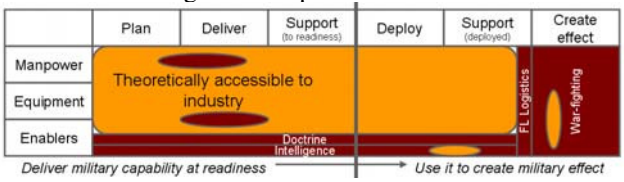


Figure 1: Industry role in capability delivery [2]

The potential for complexity within distributed decision-making to support NEC delivery is significant, and can arise from the volume of decisions being undertaken, the dynamic nature of the capability delivery process, the technical complexity involved, understanding the impact of decisions both within and across lifephases, and so on. Thus there is a need to develop an approach for the provision of decision support within the NEC delivery process, and the Decision Management and Support (DecMS) approach detailed in this paper is proposed as a potential solution. In particular, the focus falls upon how DecMS can be used to facilitate the identification of the need for decision support, and how that support can itself be generated.

Section 2 provides a review of related work within the field of decision support, and clarifies the fundamental research need that has driven the development of the DecMS approach. Section 3 outlines the fundamental decision-making model that lies at the heart of the DecMS approach. Section 4, 5, and 6 outline the means by which the need for decision support can be identified and formulated, and section 7 outlines how suitable decision support can be generated to satisfy the identified need.

2 Literature Review

The study of decision-making has received considerable attention within the research community. Research foci have varied from work aimed at understanding the fundamental nature of decision-making [3] to the development of specific tools that support decision-making within a particular domain [4]. The

focus within this paper is on the provision of support that *facilitates* decision-making for the wide variety of decision problems encountered within the capability delivery process.

The manner in which decision-making is performed can vary quite markedly in different decision situations. For example decision-making can range from being structured (explicit and formalized) [5] to something that is unstructured (implicit and cognitive) in nature [6]. However, at an abstract level, attempts have been made to characterise the fundamental phases of decision-making: i.e., to generate a fundamental model of decision-making. Simon [3] identified three phases that a decision must go through: intelligence, design and choice (Figure 2). The intelligence phase consists of finding, identifying, and formulating the problem that calls for a decision. Alternatives are developed within the design phase, and the choice phase involves evaluating, ranking, and selecting the alternatives. It is worth noting that this definition omits the implementation of a decision. The implementation of a decision choice is an intrinsic component of decision-making as a decision that is not implemented can have no effect upon solving the problem for which it is intended.

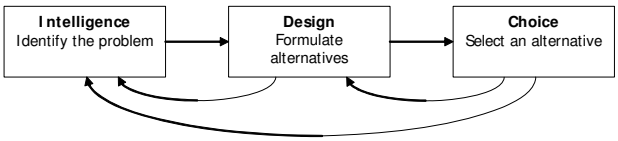


Figure 2: Simon's model of decision-making [2]

The provision of decision support within a particular decision situation represents an enactment of this model (either in whole or in part). However, this generic model needs to be refined to a more concrete level of definition that enables enactment. For example, the "Choice" phase needs to be developed to a level of detail in which the method for comparing options is defined. There are multiple methods of comparing decision options, such as linear and non-linear weighting, multi-attribute utility analysis [5], Pugh's matrix approach [7], fuzzy-based rule sets [8] and so on. The provision of decision support requires that the model is refined to a level of detail which specifies at an executable level how decision-making (or a particular aspect of it) can be performed.

Research has been undertaken to identify specific decision-making approaches that can be considered as refinements of a fundamental decision-making model. Identified approaches include the classical, administrative [9], incremental [10], mixed scanning [11], garbage can [12], and recognition primed decision [6] approaches towards decision-making. Scherpereel [13] attempted to classify decision-making problems at a slightly higher level of abstraction and identified generic types of decision problems. At one end of the scale are decision problems characterised by their simplicity and static nature, whilst at the other end of the scale are decision problems characterised by their complexity, uncertainty, and dynamic nature.

Additional research has sought to identify specific situations in which particular types of decision-making approaches are required. Tarter and Hoy [14] developed a set of ten premises, with corollaries, that could be used to determine an appropriate decision-making approach for a particular decision making situation. Sherpereel [13] developed a decision-order methodology for mapping generic problem types to generic decision-making approaches (heuristic, probabilistic, and deterministic decision-making).

There are some general similarities between Sherpereel's taxonomy and that of Tarter and Hoy [14], if viewed in terms of their levels of decision-making "structure" (Figure 3). The classical model of decision making is a highly structured form of decision-making based upon classical economic theory that attempts to find an optimal solution that is heavily computational. In contrast, less structured approaches, such as Klein's Recognition Primed Decision (RPD) [6] do not provide optimal solutions in this vein, but rather look to obtain satisfactory solutions that work above an acceptable level of performance. Such approaches tend to be more heuristic in nature and are typically employed in repetitive or time constrained decision situations.

As discussed in section 1, it is likely that decision-making will vary across in the military capability life phases. For example, loosely structured decision-making may be prevalent within the "Support (deployed)" phase due to its time-critical nature in which decisions need to be made within short time spans. In contrast, in less time-critical phases such as the "Plan" and "Deliver" phases, more structured forms of decision-making may be prevalent (Figure 3).

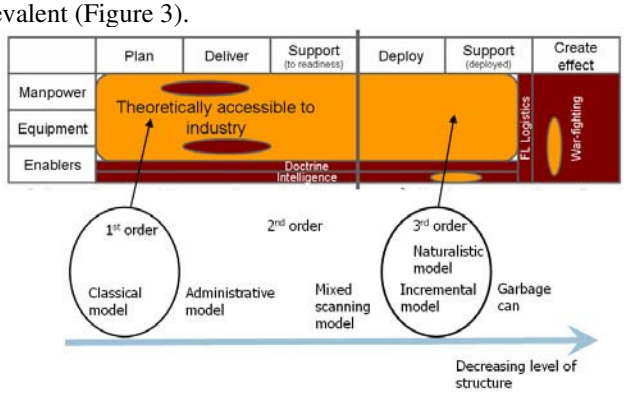


Figure 3: Mapping decision-making approaches to military capability life phases

Although current research efforts can potentially map decision-making approaches onto particular life-phases, this represents only a partial refinement of the decision-making model. An approach itself cannot be directly enacted. Rather it requires further refinement to a level of definition that facilitates the execution of decision-making. Thus, there remains a clear need to conduct research into how a fundamental model such as Simon's [3] can be refined to a process level, where processes themselves are defined in terms of tasks that when executed result in the creation and subsequent implementation of decisions (Figure 4).

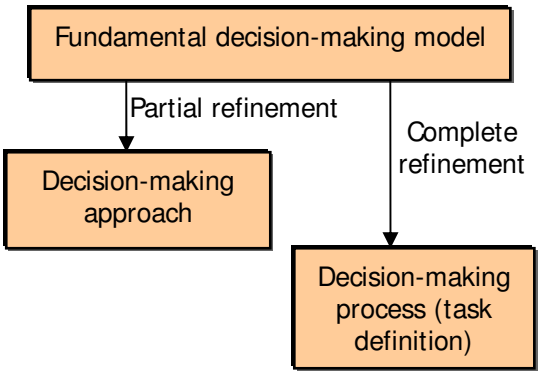


Figure 4: Refining the model of decision-making

The remainder of this paper describes the DecMS approach for identifying the need for and subsequent generation of support for different types of decision situations within the capability life phases. The key contributions of the method are argued to be that it:

- Allows the refinement of a fundamental model of decision-making to a process level at which decision support is defined at an executable level of detail.
- Refines this model such that the developed decision support is appropriate for the current decision situation.

3 Fundamental Model of Decision-Making

Provision of decision support within a decision problem represents an instantiation of the fundamental model of decision-making illustrated in Figure 5. The model represents an extension of Simon's [3] and includes the addition of a detection and implementation phase around the core phases proposed by Simon. Thus the five phases of the model are:

- The detection phase, the purpose of which is to identify that a potential decision problem exists and that there is a need for decision support.
- The identification phase, the purpose of which is to formulate the decision problem.
- The generation phase, the purpose of which is to obtain possible solutions and also partially evaluate those solutions.
- The evaluation phase, the purpose of which is to predict the outcome of generated solutions if they are implemented, determine how well they satisfy the requirements of the formulated decision problem, and select the one that provides the highest satisfaction levels.
- The implementation phase, the purpose of which is to implement and monitor the selected decision.

This fundamental model is equally applicable to structured and unstructured, explicit and implicit, and individual and group-based decision-making. It is the *refinement* of this model that will vary in accordance with the type of decision-making taking place. For example consider the two simplistic refinements (to basic tasks) of the generation and evaluation phases of the model presented in Figure 6.

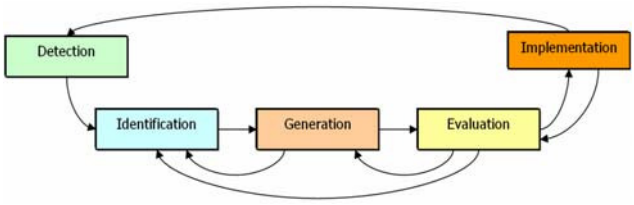


Figure 5: The fundamental model of decision-making (extension of Simon [2])

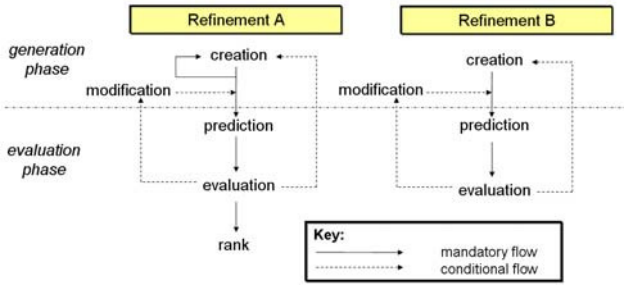


Figure 6: High level refinements of the model

In Refinement B, a potential decision solution is created, the effects of implementing it predicted, and the effects subsequently evaluated. If the evaluation reveals that the potential solution will resolve the current decision problem, then there is no need to return to the creation task in order to create a second alternative solution. Rather the decision-making continues on into the implementation phase. It is only if evaluation of the potential decision reveals that it will not resolve the problem that the feedback loop from the evaluation to generation phase would come into effect to modify the potential solution or to generate a fundamentally different alternative. In a more structured approach such as Refinement A, decision-making would execute in a fundamentally different manner. Initially ideas would be generated, their effects predicted, and evaluated against the decision goals. Any alternative that fails the evaluation could then be reconsidered for modification and subsequent re-evaluation. Finally the alternatives would be compared with each other to determine which best resolves the decision option.

As Figure 6 illustrates, the two refinements differ in terms of the number of tasks that they perform, and in the manner in which the feedback loops are invoked during the process. With respect to the tasks, the less structured approach (Refinement B) does not have a ranking task as decision solutions are considered individually and not compared with others. The objective is simply to reach a decision that satisfactorily resolves the decision problem. In contrast however the more structured approach (Refinement A) involves the evaluation of different potential solutions and thus has a ranking task in which these decisions are compared such that the optimal decision solution from the set of options can be identified and subsequently implemented.

In addition to variations in tasks, the flow of the process can also vary. For example in Refinement A the loop to the creation task is executed at least once as the decision-making approach involves the generation of a number of potential decision solutions. In the less

structured approach represented by Refinement B, the feedback loop to the creation task is only executed if the potential decision solution fails during evaluation.

4 Supporting Decision-Making

Decision support is obtained through instantiating all or part of the fundamental model (Figure 7). In this regard the model is split into two parts as different control mechanisms are used to refine different parts of the model. A basic trigger selection mechanism controls refinement of the detection phase, whereas a case-based reasoning mechanism is used to control refinement of the remaining four phases (Figure 8).

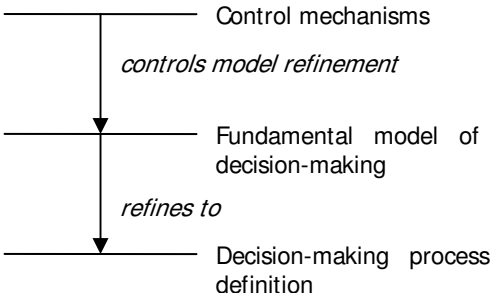


Figure 7: Refining the model

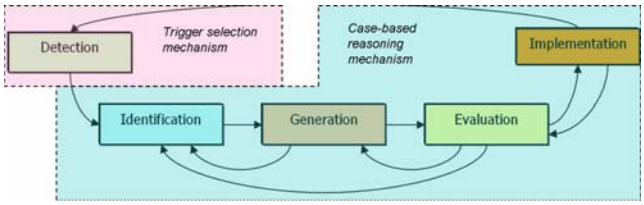


Figure 8: The different control mechanisms

The trigger selection mechanism supports the selection of a trigger mechanism from a number of pre-defined mechanisms for monitoring when the need for a decision exists. The case-based reasoning mechanism employs an analogical form of reasoning to determine how the remaining four phases (which are considered as the "solution" phases) can be instantiated once the need for a decision has been triggered in the detection phase. Each of these control mechanisms are discussed in greater detail in sections 6 and 7 respectively, but before proceeding it is important to clarify the structure of a capability delivery process and the capability "viewpoints" of a process, which describe specific aspects of a process, as both play a significant role when refining the decision-making model to provide decision suppport.

5 Processes and Capability Viewpoints

The capability delivery process consists of a number of tasks that if executed should result in the delivery of the desired capability (Figure 9). Four abstract viewpoints are used to represent these tasks. These are the capability objectives, products, enablers, and constraints. Duffy [15] has defined relationships between goals, tasks, inputs, outputs, and resources within the realm of engineering design, which can be revised to illustrate the relationships that exist between the capability viewpoints (Figure 10). Points of interest are that a task consists of activities which represent how a task is to be performed and that a task can contain one or more activities.

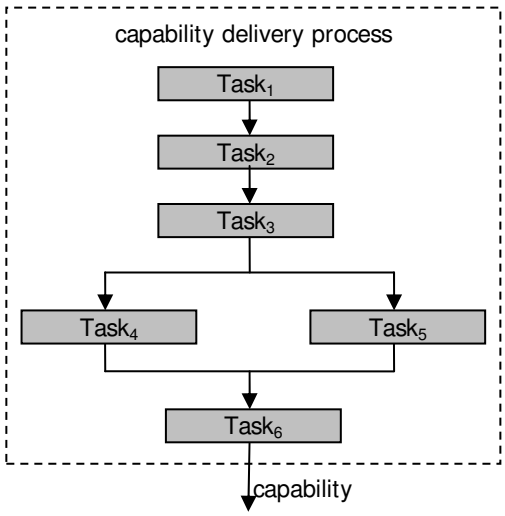


Figure 9: A capability delivery process

The capability objectives represent the capability to be delivered by the process. However, within a process there exist a number of tasks with their own capability objectives which may either:

- Contribute directly to the desired capability; or
- Create enablers that can subsequently be used within the process to produce the desired capability.

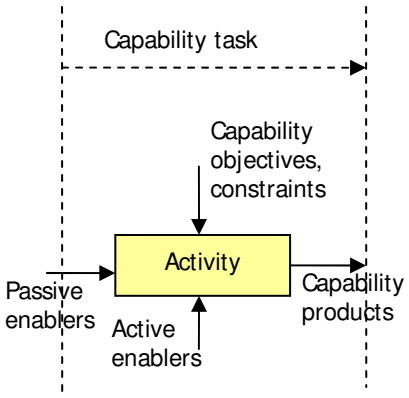


Figure 10: The relationship between capability tasks, enablers, objectives and constraints

The capability enablers facilitate the creation of the capability products (which are the solutions that satisfy the objectives). The capability constraints place limitations on behavioural aspects of the enablers. Whereas capability objectives represent what is desired of a process or task, the capability constraints represent boundaries that place limitations upon the capability enablers used to deliver the objectives. The capability enablers must therefore satisfy both the capability objectives and constraints.

There are two fundamental types of enablers within processes: active and passive enablers. Active enablers

are the processing enablers within any task: i.e., they are the means by which the outputs are created. In contrast, the passive enablers are used by the active enablers to produce the outputs. Thus a typical example might be a manager (active enabler) analysing market data (passive enabler) to identify potential markets for exploitation.

Figure 11 illustrates a number of possible themes proposed as being defining characteristics of a network enabled capability. The list is by no means complete, and current research efforts are still ongoing to identify and define suitable themes. However, these themes have been tentatively grouped under the four capability viewpoints as illustrated in Figure 11. An interesting feature to note is that themes can belong to different viewpoints. For example decisions are classified as both capability enablers and constraints depending upon their nature. Decisions can be classified as enablers as they are means-to-an-end rather than being an end in their own right. Thus decisions can be used to enable a capability performance that satisfies the capability objectives and constraints. For example, an enabling decision can be made which grants authority to individual A to perform a particular task. In contrast decisions can also be classified as capability constraints as they can limit the options that may be available within the execution of a process. For example, a decision may be made to limit a resource's role, such as "Individual A is not allowed to authorise completion of task D."

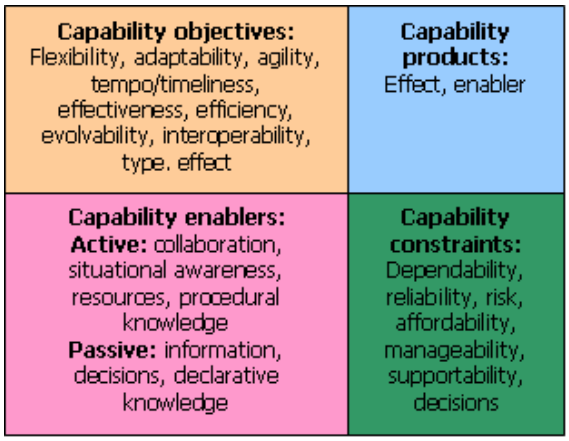
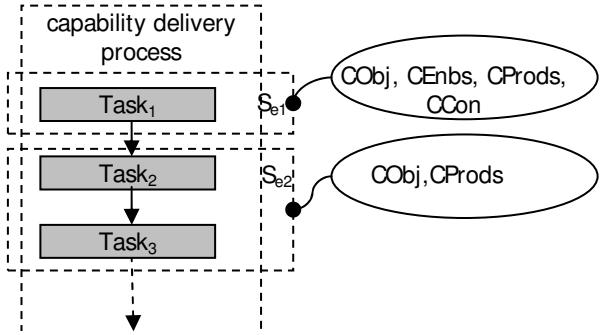


Figure 11: The four capability viewpoints

6 Triggering Decision Support

Three triggers are proposed for identifying when decision support is required and a common feature to all is that they use an understanding of expected situations that can be associated with individual segments of a process. Within a process, expected situations can be defined in terms of the capability objectives, constraints, enablers, and outputs. They need not be defined at the start of the process, but may be defined as the process executes and can be attached with either individual tasks, particular aspects of a task, or groups of tasks (Figure 12). These situations represent distinct points within the progress of the process and are explicitly defined. One of the reasons for using this idea of expected situations within NEC delivery is that it facilitates shared situational awareness. Defining these situations allows resources within the network delivering a capability to be aware of the overall situation in which they are participating. They can understand their own role within the process as well as the role of others.



Key:
CObjs – Constraints objectives CCntrs – Capability constraints
CEnbs – Constraint enablers CProds – Capability products
S_e – Expected situation

Figure 12: Defining expected situations

Three mechanisms for triggering the need for decision support are proposed, each of which uses expected situations although the manner in which they are used varies:

- **Trig1**: The expected situation states that a decision is to be made, thus triggering a need for decision support.
- **Trig2**: The actual situation that arises when the process executes varies from that which is expected. This deviation from the expected situation triggers the need for decision support.
- **Trig3**: A resource within the process can initiate the need for support in experimental decision-making by suggesting potential deviations.

In Trig1, realisation of the need for decision support comes from a listed decision contained within an expected situation defined for a task or group of tasks. Recall that decisions are considered one of the capability viewpoints and thus decisions known in advance can be included in the expected situations. The nature of the decisions can vary. For example they can relate to a specific aspect of how the process executes, e.g., "decide upon which resources are required to perform the task associated with this situation." Alternatively they can be directly concerned with the desired capability. By example, if the desired capability is to launch a coastline assault from offshore, then there will inevitably be a stage in the capability delivery process when decisions need to be made regarding how many ships will be involved in the operation, when they will fire, what weapon systems they will use, etc.

Alternatively decisions can arise dynamically when the process is executed and the process deviates from the expected situations with respect to the defined capability enablers, outputs, objectives, and constraints. When the deviation exceeds defined boundaries, then that deviation triggers the need for a decision to handle the deviation (Figure 13). It is worth noting that deviation can be

detrimental or beneficial in nature. When the deviations are beneficial, the decision support required can revolve around taking full advantage of the deviation. In contrast, when the deviation is detrimental in nature, decision-making revolves around correcting process performance. Deviation can occur with respect to each of the four capability viewpoints. The example in Figure 13 illustrates how a delivery process has failed to meet the expected situation S_{e3}. This has triggered the need for a decision to be made with regard to modifying the process. S_{e3} evolves to S_{e3}' as a result of the decision made to reconfigure that part of the process by performing tasks T_3, T_4, and T_5 in a different order. In these circumstances, decision support is required to facilitate decision-making about the process and its execution.

In this example only the capability enablers have been modified in the evolved situation S_{e3}' – the constraints and objectives have remained the same. However it is possible that deviations can occur with regard to each of the four capability viewpoints:

- The capability objectives can deviate. This can be a user driven change in which the capability need is altered in response to a changing environment. For example new objectives can be defined, existing ones modified, refined, or removed. This can become an issue if unexpected events take place which result in the need to review not only the objectives of (and indeed need for) particular capability tasks, but also the need to review the ultimate capability the process is required to deliver.
- The capability enablers can deviate from that expected, which can affect the process performance. Typical examples include:
  - Some enablers can become unavailable, for example resources (such as human decision-makers) may be absent and thus cannot perform their required tasks.
  - Resources can under perform, for example computer networks can slow down when placed under heavy loads, etc.
  - The nature of the collaboration between resources can change, for example conflicts can arise that cause their method of working together to fail.
  - The quality of information (passive enablers) within the delivery network can be less than that expected: e.g., some information may be incomplete or uncertain.
  - The need for new decisions can become apparent as the process executes, thus the decisions associated with a particular situation can deviate from those which are expected (Figure 14).
- The capability constraints can deviate. Again this can occur is a similar vein to deviations associated with capability objectives. Thus, new constraints may be added for example in response to unexpected events that take place, and existing ones removed or modified.
- The capability products may not satisfy the objectives and constraints thus it may be necessary to make adjustments to the enablers used to generate the outputs.

An alternative method by which the need for decision-making is triggered is a proactive approach in which decision-makers initiate decision-making, not because a delivery process is not meeting its objectives, but rather to experiment with aspects of the process to determine what effects potential deviations might have. Whereas Trig1 and Trig2 are concerned with triggering decision-making in which the need is formulated prior to solution development, this approach allows decision-making to be triggered such that potential solutions can be generated prior to identification of a specific decision need. It is only when decisions are generated and evaluated that the need for them can become clear. The triggering method is almost identical to that used within Trig2. The only difference is that *potential* rather than *actual* deviations trigger the need for decision support.

7 Generating Decision Support

Upon triggering the need for decision support, it is necessary to generate suitable support. Recall from Section 3 that this support is provided through using the control mechanisms to refine the fundamental model of decision-making to an executable level of detail. Triggering support is provided through a selection mechanism, but refinement of the identification, generation, selection, and implementation phases of the model use an analogical approach known as case-based reasoning in which knowledge of similar decisions that have been made before is used to solve the current decision need in the form of a Decision Support Provisions (DSP). Applicable decision knowledge is identified using case-based reasoning (CBR) [16][17]. This is an analogical approach in which previous decision experiences are recalled for use in new decision situations. The heuristic drive behind CBR is that similar decision situations require similar types of decision support, thus CBR allows the DM module to determine what decision support is required in new decision problems by searching for, recalling, and re-using the decision support required for similar decision problems.

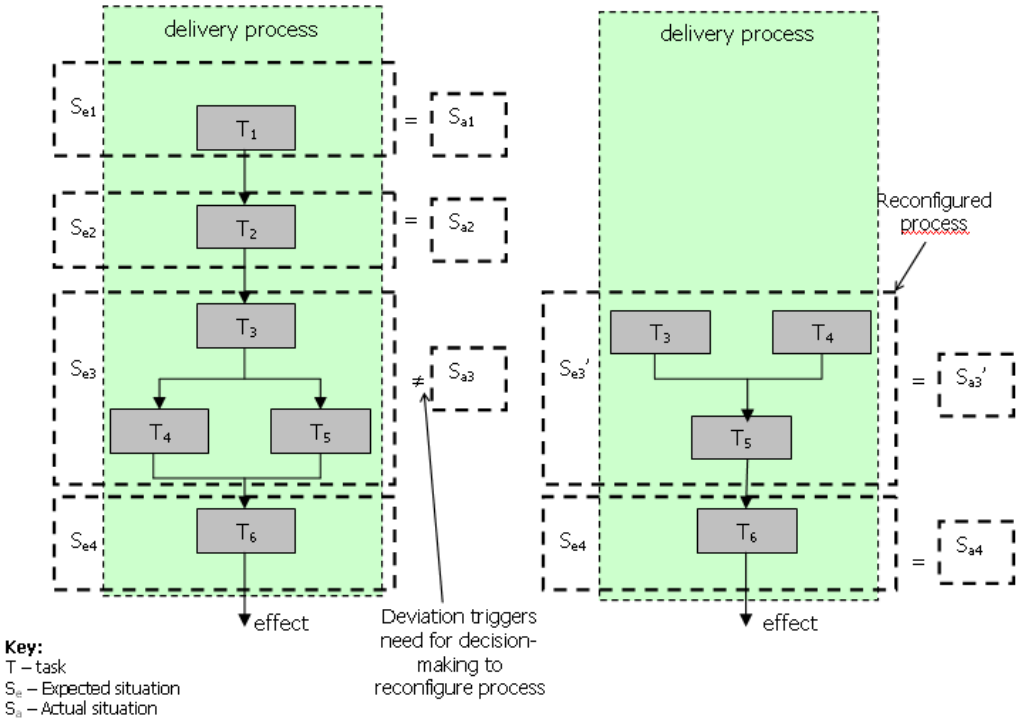


Figure 13: Decision-making triggered by deviation from expected situations

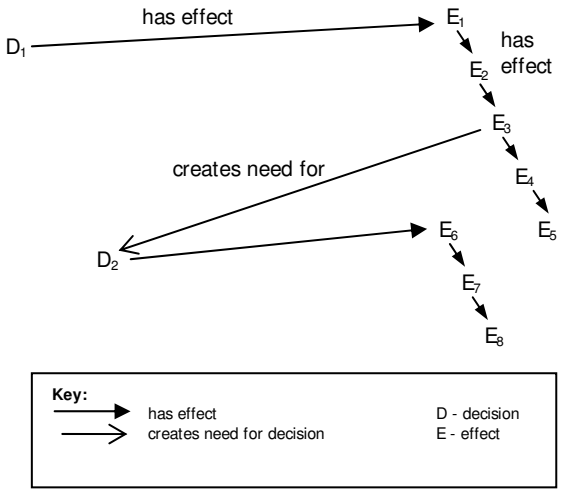


Figure 14: Decision and effect mapping

Four abstract DSPs are proposed within the DecMS approach. These are decision insertion, decision process re-enactment, decision process modification, and decision process creation:

- **DSP_1: Decision insertion.**
  - Recall an existing decision from a similar decision situation.
  - Implement this decision.
- **DSM_2: Decision-making process re-enactment.**
  - Recall the decision-making process by which the recalled decision was reached.
  - Modify the recalled decision by re-enacting the decision-making process by which it was originally created.
  - Implement the modified decision.
- **DSM_3: Decision-making process modification.**
  - Modify the recalled decision-making process.
  - Modify the recalled decision by re-enacting the modified decision-making process.
  - Implement the output.
- **DSM_4: Decision-making process creation.**
  - Generate a new decision-making process manually.
  - Implement decision-making process.
  - Implement decision produced by this new process.

Essentially the contents of DSP_1 instantiate the implementation phase of the decision-making model, and the contents of DSPs 2, 3, and 4 instantiate the identification, generation, and evaluation phases (Figure 15). In the DecMS approach, a suitable decision must be recalled to execute DSP_1, and a suitable decision and the process by which it was created must be recalled to execute DSP_2 or DSP_3. This is achieved using the CBR technique in which decision cases (which contain this decision knowledge) are recalled. It should be noted that the DecMS approach does not support DSP_4 to any level of detail. DSP_4 is only used when DSPs 1, 2, and 3 are found to fail and the contribution of DecMS is that it identifies when it becomes the user's responsibility to develop appropriate decision support. However, DSPs 1,2, and 3 are generated using CBR.

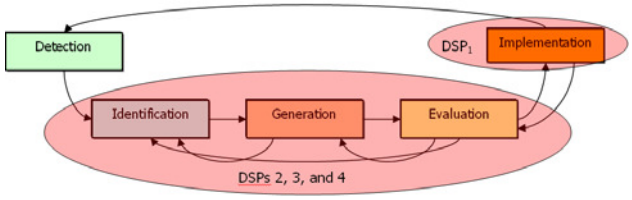


Figure 15: Instantiating the model

7.1 Using CBR to generate a DSP

CBR is an example of analogical reasoning – a problem solving technique that relies on the use of experiential knowledge to solve new and unfamiliar problems. Within the design domain for example, analogical reasoning allows a designer to recognize something that has not been encountered before by associating it with something that the designer is familiar with. Analogies can occur at different levels of abstraction and indeed analogies can be drawn between two entirely different domains. For example, instinctively there is little to relate the domains of train design and bullet design but at a high level of abstraction it is possible to relate the domains when considering the fact that aerodynamically they present similar problems: i.e., both must travel through the air and a common concern is how to minimize the drag forces they are subjected to as they move.

CBR is a very specific form of analogical reasoning that is generally only capable of forming analogies within one particular domain and is not able to make the leap of comparing trains to bullets. Instead of relying solely on general knowledge of a problem domain, or making associations along generalized relationships between problem descriptors and conclusions, CBR uses specific knowledge of previously experienced, concrete problem situations (cases). When confronted with a new situation, this knowledge is retrieved and adapted to form a new solution. In addition to previous cases, CBR also employs general knowledge about a particular domain. Typically this general domain knowledge is used to adapt the retrieved case so that it satisfies the new problem.

In operational terms, CBR operates on a four stage retrieve, re-use, revise, and retain cycle [18], in which when faced with a new problem a similar case is retrieved from the case base, directly re-used in an attempt to solve the problem, revised if its direct re-use fails to result in a satisfactory solution, and this new case is subsequently retained within the case library.

Within the DecMS approach, a new decision problem within the NEC delivery process constitutes a new decision case (Figure 16) for which a decision is sought. Retrieval from a library of previous decision cases is achieved via an indexing approach in which the capability viewpoints are used as the indexes. Decision cases are discussed in greater detail in Section 7.2 but essentially they contain a list of indexes representing the decision problem, the decision-making process by which the problem was solved, and the decision itself.

Case indexing is achieved through the use of the capability viewpoints detailed in Section 5. The manner in which these viewpoints are used to index decision cases varies depending upon the trigger method, but the viewpoints for particular decision situations are used as the attributes that define that situation. The objective of retrieval is to obtain from the case library the previous decision cases that exhibit some or all of the same capability viewpoints such that they can be ranked in terms of similarity to the current decision problem and the top one selected as the potential solution.

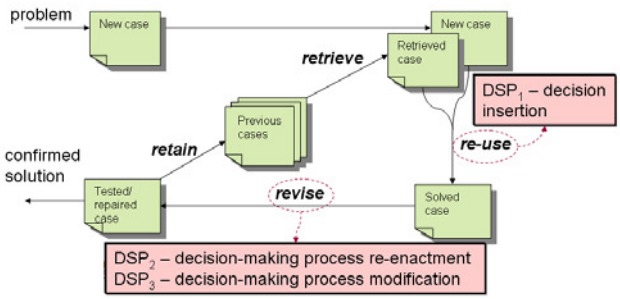


Figure 16: Using CBR to support the DSMs

The decision of a recalled decision case is then proposed as an initial solution for the new decision case (DSP_1), but if it fails during testing then it is modified by replaying the decision-making process by which that decision was initially obtained to determine if replaying the process creates a decision that is suitable (DSP_2). A particularly interesting feature of the DecMS approach with regard to the DSP_2 revise mechanism is the manner in which the modification of the recalled decision is performed. Rather than have an additional domain knowledge base to modify the decision, the recalled case modifies itself by replaying the decision-making process by which it was initially created. If that process is replayed in a different context then a suitable decision may be obtained. Consider for example a business process set up to decide upon a supplier for a piece of equipment. The nature of the information flowing through the process can remain the same for different pieces of equipment (e.g., all equipment has a cost, lead times, etc. associated with it), but the content of that information will change (i.e., the values associated with cost, lead time, etc. will be different). Thus although the process remains the same, the variation in the content of information flowing through it can result in a number of different outputs from the same process. Essentially therefore, within the DecMS approach, the recalled decision case is responsible for modifying itself.

However, should DSP_2 fail then the decision-making process is modified (DSP_3), for example through re-ordering existing or adding in new tasks. Techniques for doing this can vary. One option is to allow the user to alter the process manually or alternatively algorithms [19] aimed at improving process performance can be used as a means of modifying a process until a suitable decision is obtained.

7.2 A decision case

One important feature of the CBR is the structure of the decision case (Figure 17). Cases contain details of a decision problem, the decision, and the process by which that decision was generated. The scope of the decision can vary. For example some cases will contain decisions that exist within the delivery process and are thus related to the subject of the delivery process (i.e., the capability). Alternatively, others may contain decisions about the delivery process. The nature of the case recalled is determined by the trigger mechanism during the retrieval stage of DSP generation (see Section 7.3 for details).

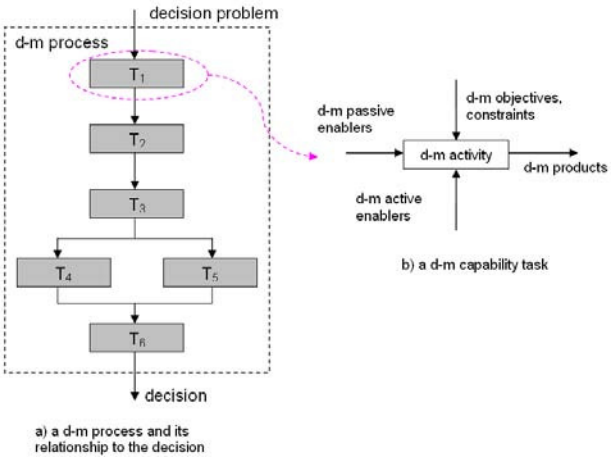


Figure 17: The contents of a decision case

Decision-making processes within a decision case are modelled in the same fashion as any other process, and are thus described in terms of tasks and the viewpoints associated with those tasks (Figure 17b). The decision problem specifies the situation in which the d-m process was implemented and is defined in terms of the viewpoints. It acts as the basis for decision case recall and is discussed in greater detail in Section 7.3. Decision-making processes are described in terms of the decision-making tasks that make up the process (Figure 18). These tasks are defined in terms of their enablers (both active and passive), constraints, objectives, and products.

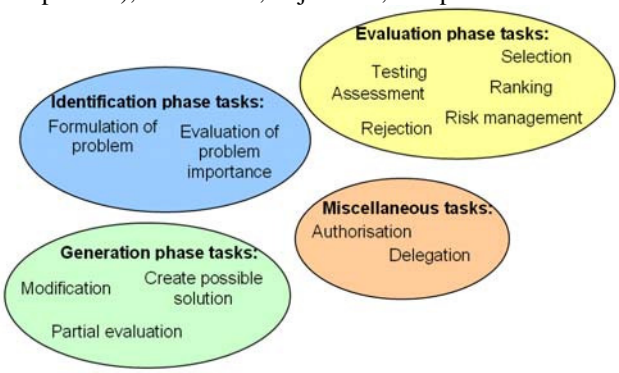


Figure 18: High level decision-making tasks

7.3 Retrieving decision cases to create DSPs

Decision case retrieval is concerned with identifying earlier decision cases that can be used to address identified decision problems. The knowledge retrieved from cases can vary, for example DSP_1 only needs to recall the *decision* associated with a case whereas DSP_2 and DSP_3 recall the *decision-making process*. The DecMS approach advocates the use of indexing as the primary means by which retrieval is performed. Thus the decision problem of a case is specified using a series of indexes, which in their simplest form might assume the structure "capability viewpoint: attribute-operator-value". For example, cost might be a metric of affordability that would be indexed in its simplest form as "affordability: cost = low" or "affordability: cost = £100k". The indexes of the current decision problem can be compared to those of existing cases within the case library to determine the similarity of these existing cases and the most similar one subsequently used to create the DSP.

The key task therefore is to determine the indexes for the current decision problem that should be used as the basis for retrieval of decision cases and within the DecMS approach the viewpoints that are used to define expected situations are employed as the indexes. The exact indexes are dependent upon the triggering mechanism in operation, but the commonality for all of the triggering mechanisms is that the capability viewpoints of an expected situation are used as the basis for the indexes. These indexes therefore define the decision problem.

As discussed in Section 6.2, three trigger mechanisms are supported within the DecMS approach. Decision support can be triggered when the expected situation states that a decision needs to be made (Trig1), when the actual situation that arises within a capability delivery process deviates from that which is expected (Trig2), and when decision-making is actively initiated in an experimental approach and potential deviations from expected situation are investigated (Trig3). For each mechanism, the indexes are formed in a different manner (Figure 19).

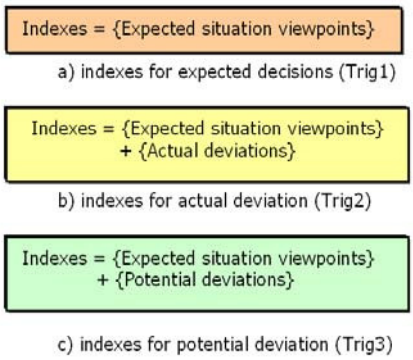


Figure 19: Case indexes for each trigger mechanism

Decision support triggered by expected decisions uses the viewpoints attached to the expected situation as the indexes. Decision cases are retrieved from the case library that exhibit some or all of these indexes which can then be compared and ranked using standard weighting techniques [4]. However trigger mechanisms Trig2 and Trig3 use these viewpoints in addition to the deviation from the expected situation viewpoints as the basis for recall. Thus continuing with the affordability viewpoint example, the expected cost associated with a task may be under £100k: During execution of the process however it may be observed that the cost is greater than this, for example £200k. Thus the decision problem index listing includes an additional set of indexes that represent the deviations and in this instance would include a shortfall index for "affordability: cost". It is important to note that these deviations can be positive or negative in terms of their effect. Negative effect deviations require corrective decisions to be taken to bring performance of the delivery process up to the required levels. However, when the deviations are positive in nature such that performance exceeds the stated performance levels within the expected

situation, then exploitative decisions are required to investigate if decisions can be taken to exploit the benefits of this unexpected deviation.

8 Conclusion

A fundamental nature of the capability delivery process is that given the wide variety of decisions to be made and the different demands that different life phases place upon decision-making, it is likely that a similarly wide variety of decision support must be provided to aid this decision-making. There is a need therefore to develop an approach that can determine the nature of decision-making (and requisite support) for specific decision problems and that can subsequently generate that decision support to an executable level of detail. The DecMS approach detailed in this paper is proposed as a means to satisfy this need, in which appropriate decision support is provided through instantiating a fundamental model of decision-making, where the instantiation is based upon an understanding of the current decision problem. The fundamental model consists of five phases – detection (of a decision problem), identification (formulation of the decision problem), generation (of potential solutions), evaluation (of these potential solutions), and implementation (of the selected decision).

Two control mechanisms are used to instantiate the model to provide appropriate decision support. A trigger selection mechanism is used to control the detection phase through monitoring user defined expected situations. These situations are defined using the capability viewpoints that are attached to a capability delivery process. Once the need for decision support is triggered, an analogical approach is used to instantiate the solution phases of the decision-making model (identification, generation, evaluation, and implementation) such that decision support appropriate to the current decision problem is provided. To ensure that suitable decision support is provided, the analogical reasoning process uses the capability viewpoints attached to expected situations to find similar decision knowledge from a library of past decision cases that is suitable for re-use in the current decision problem..

Currently, the DecMS approach proposed in this paper is being tested within a number of capability management scenarios. Two principal objectives of this testing are to:

- Elicit how the viewpoints can be represented such that they can comprehensively define expected situations within the capability delivery process whilst still being generic enough in structure that they can be incorporated into the decision case retrieval mechanism.
- Evaluate how effective the DecMS approach is at providing a suitable breadth of executable decision support across the life phases of capability delivery.

Acknowledgements

The research has been funded by BAE Systems and the UK Engineering and Physical Science Research Council (EPSRC) under grant number EP/D505461/1.

9 References

[1] UK MoD, NEC JSP 777 Edition 1, UK MoD, 2004.

[2] Secretary of State for Defence, Defence Industrial Strategy: Defence White Paper, UK Secretary of State for Defence, 2005.

[3] Simon, H.A., *The New Science of Decision-Making*, Harper and Row, New York, 1960.

[4] Shortliffe, E.H., *Computer-based Medical Consultations: MYCIN*, Elsevier, New York, 1976.

[5] Thurston, D., "A Formal Method for Subjective Design Evaluation with Multiple Attributes", *Research in Engineering Design*, 1991, Vol. 3, pp. 105-122.

[6] Klein, G., "A Recognition Primed Decision (RPD) Model of Rapid Decision Making", *Decision-Making in Action: Models and Methods*, editors Klein, G., Orasanu, R., and Zsambok, C.E., Abbex Publishinng: New Jersey, 1993, pp.138-148.

[7] Pugh, S., *Total Design*, Addison-Wesley, 1990.

[8] Juang, Y-S., Lin, S-S., and Kao, H-S., "Design and Implementation of a Fuzzy Inference System for Supporting Customer Requirements", *Expert Systems with Applications*, 2007, Vol. 32, pp. 868-878.

[9] Simon, H.A., "Decision-making: Rational, Non-rational, and Irrational", Educational Administration Quarterly, 1993, Vol. 29, pp. 399-411.

[10] Lindblom, C.E., "The Science of Muddling Through", *Public Administrative Review*, 1959, Vol. 19, pp. 79-99.

[11] Etzioni, A., "Mixed Scanning: a Third Approach to Decision-making", *Public Administration Review*, 1967, Vol. 27, pp. 385-392.

[12] Cohen, D.K., March, J.G., and Olsen, J.P., "A Garbage Can Model of Organizational Choice", *Administrative Science Quarterly*, 1972, Vol. 17, pp. 1-25.

[13] Sherpereel, C.M., "Decision orders: a decision taxonomy", *Management Decision*, 2006, Vol. 44(1), pp.123-136.

[14] Tarter, C.J. and Hoy, W.K., "Toward a contingency theory of decision-making", *Journal of Educational Adminstration*, 1998, Vol. 36 No.3, pp. 212-228.

[15] Duffy, A.H.B., "Designing design", *Proc. Of 3rd International Seminar and Workshop in Engineering Design in Integrated Product Development*, Zielona Gora - Lagow, Poland: Design Society, 2002.

[16] Maher, M. L. and Garza, A. G. S., "Case-based reasoning in design", *IEEE Expert*, 1997, Vol. 12(2), pp. 34-41.

[17] Kolodner, J., *Case-Based Reasoning,* Morgan Kaufmann, San Mateo, USA, 1993.

[18] Aamodt, A. and Plaza, E., "Case-based reasoning: Foundational issues, methodological variations, and system approaches", *Artificial Intelligence Communications*, 1994, Vol. 7(1), pp. 39-59.

[19] Whitfield, R.I., Duffy, A.H.B., Coates, G., and Hills, W., "Efficient process optimisations", *Concurrent Engineering: Research and Applications*, 2003, Vol. 11(2).