Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

State of the Art Report on Patterns in Systems Engineering and Capability Engineering

Attilio Bagnulo
Tim Addison

Prepared By:
CGI Group
275 Slater Ottawa, ON
Contract Project Manager: Richard d'Anjou, (418) 623-5051 x 2513
PWGSC Contract Number: PWGSC Contract Number: W7701-083926
CSA: Christophe Nécaille, (418) 844-4000 x 4490

Defence R&D Canada – Valcartier
Contract Report
DRDC Valcartier CR 2010-012
March 2010

Canada

State of the Art Report on Patterns in Systems Engineering and Capability Engineering

Attilio Bagnulo
Tim Addison

Prepared By:
CGI Group
275 Slater
Ottawa, ON
Contractor's Document Number:
Contract Project Manager: Richard d'Anjou, (418) 623-5051 x 2513
PWGSC Contract Number: PWGSC Contract Number: W7701-083926
CSA: Christophe Nécaille, (418) 844-4000 x 4490

Defence R&D Canada – Valcartier

Contract Report
DRDC Valcartier CR 2010-012
March 2010

Principal Author

Attilio Bagnulo, CGI Group

Approved by

Christophe Necaille

Defence Scientist, DRDC Valcartier

Approved for release by

Guy Turcotte

SdS Section Head, DRDC Valcartier

Abstract

This document presents the results of a survey of open literature on the following subjects:

- Patterns in general;
- Patterns for Systems Engineering (SE), Systems of Systems Engineering (SoSE), and Capability Engineering (CE);
- Process Patterns;
- Organizational Patterns Pattern Languages;
- Pattern Modeling Main Academic active Players in the domain;
- Main Conferences and Publications within the domain;
- Main Research Streams and Results;
- Main Patterns Documentation Forms & Standards.

This page intentionally left blank.

Table of contents

List of figures

1 Introduction

This document presents the results of a survey of open literature on the following subjects:

- Patterns in General;
- Patterns for Systems Engineering (SE), Systems of Systems Engineering (SoSE), and Capability Engineering (CE);
- Process Patterns;
- Organizational Patterns[1];
- Pattern Languages;
- Pattern Modeling[2];
- Main Academic active Players in the domain;
- Main Conferences and Publications within the domain;
- Main Research Streams and Results;
- Main Patterns Documentation Forms & Standards.

1.1 Document Objective

The intent of this document is to provide the PACEM-Re Project Team with a sound knowledge base that will serve the purpose of verifying the applicability of the concept of "Patterns" to:

- The current and future versions of the Capability Engineering Process (CEP);
- The Work Products (e.g., artefacts) resulting of the application of the CEP within the current and future DND/CF CE Initiatives.

[1] This subject was not identified in the original PACEM-Re Statement of Work but it appears to be relevant to both CET Organization Model and DoDAF Organizational Relationship Maps (OV-4).
[2] This subject was also not identified in the original PACEM-Re Statement of Work but it appears to be relevant and worthy of further examination.

2 Survey and Update of Open Literature

The survey results described hereafter were essentially obtained from the Internet where they are available in the forms of white papers, articles, books and websites. In order to facilitate retrieval, reference information is provided.

2.1 Patterns in General

There are several definitions of a pattern in the open literature. To begin with, the Oxford English Reference Dictionary describes a pattern as "a model or design from which copies can be made". "Patterns might be so specific as to name particular objects, interface elements or implementation structures in a solution, or so general as to describe high-level architectures or the structures of the organizations that create them. They could describe configurations of hardware, software or even people" (ref[1]).

If we look at some definitions of pattern as applied to systems engineering and systems development we see that the Oxford definition is quite generic and simple but nonetheless applicable. In Ref [2], Barter defined a pattern as "a solution of a specific problem placed in a specific context", where the solution is a model or design. This model or design can differ depending on the type of pattern. For example, according to Ambler, process patterns are "a collection of general techniques, actions, and/or tasks (activities)" (Ref [3]). In this case the models are a collection of general techniques, actions and/or tasks. This is conceptually similar to Cloutier, who saw a pattern "as a model or facsimile of an actual thing or action, which provides a degree of representation (an abstraction) enabling the repeated recreation of that entity." (Ref [4]).

A design pattern in architecture and computer science is a formal way of documenting a solution to a design problem in a particular field of expertise. This idea/definition was introduced by Christopher Alexander in the field of architecture and has been adapted for various other disciplines, including computer science (Ref [14]). From this context the pattern was considered to be a three-part rule, which expresses a relation between a certain context, a problem, and a solution (Ref [15]). Design patterns have moved into the mainstream of professional software development as a highly effective means of improving the quality of software engineering, system design, and development, as well as the communication among the people building them. Patterns capture many of the best practices of software design, making them available to all software engineers.

The goal of patterns is to re-use solutions to solve similar problems to save time and costs. There are different types of patterns to address various problems and context. These patterns can address patterns for Systems Engineering, System of Systems Engineering and Capability Engineering (referred to as only Capability Engineering in the rest of this paper), patterns for processes and patterns for organization structure.

Patterns for Capability Engineering are patterns of solutions to Capability Engineering problems which would include models and other artefacts of operational and system architectures including requirements and system processes. These patterns are analogous to design patterns as described

in the literature. These architectures depict the inter-relationships of systems and system components both from an operational and systems perspective documented in such a way that they can be re-used to solve similar capability engineering problems. These patterns can address the business or organizational processes of the Department of National Defence.

Process patterns, as described by Ambler (Ref. [3]), are a collection of general techniques, actions, and/or tasks (activities) from which an organization can develop a tailored process that meets its exact needs. Even though he describes process patterns within the context of software development, the concepts can be applied to capability engineering.
Organizational Patterns are closely related to Process Patterns in that they deal with actions and/or tasks specific to the organizational construct of a group, team, company, corporation, department, etc. The organizational patterns often determine the output of the organization and its efficiencies, or lack thereof.
The Capability Engineering Processes (CEP) can be viewed as a collection of process and organizational patterns to perform capability engineering within the context of DND. The work products of the CEP become the Capability Engineering patterns for future work products in support of future capability engineering initiatives.
The following sections provide more detail on these types of patterns.

2.2 Capability Engineering Patterns

In Ref [4], a pattern hierarchy is proposed that provides a contextual picture to position the various types of patterns. This hierarchy identifies the context within which solution patterns would be applied. That is, the organization, business and mission patterns define the problem space and context. The subsequent system architecture patterns then define the collection of solutions that can be applied. As described in Figure 1, the system architecture patterns are composed of five sub-patterns: structural, engineering roles (organization), requirements, activities (process) and system process. These various types of patterns all describe various aspects of the solution that can be re-used in subsequent capability engineering initiatives. Organization and activities (process) patterns are described in more detail in separate sections below.

- A structural pattern is a physical pattern to follow when designing a part of the architecture to describe how to build the system;
- System requirements patterns prescribe the format of a properly formed requirement, or a collection of requirements that can be reused to describe desired functionality; and
- System Process patterns define the processes performed by the system.

This hierarchy is very interesting and should be further explored and refined to better reflect the nature of capability engineering and the dynamics of a capability engineering initiative.

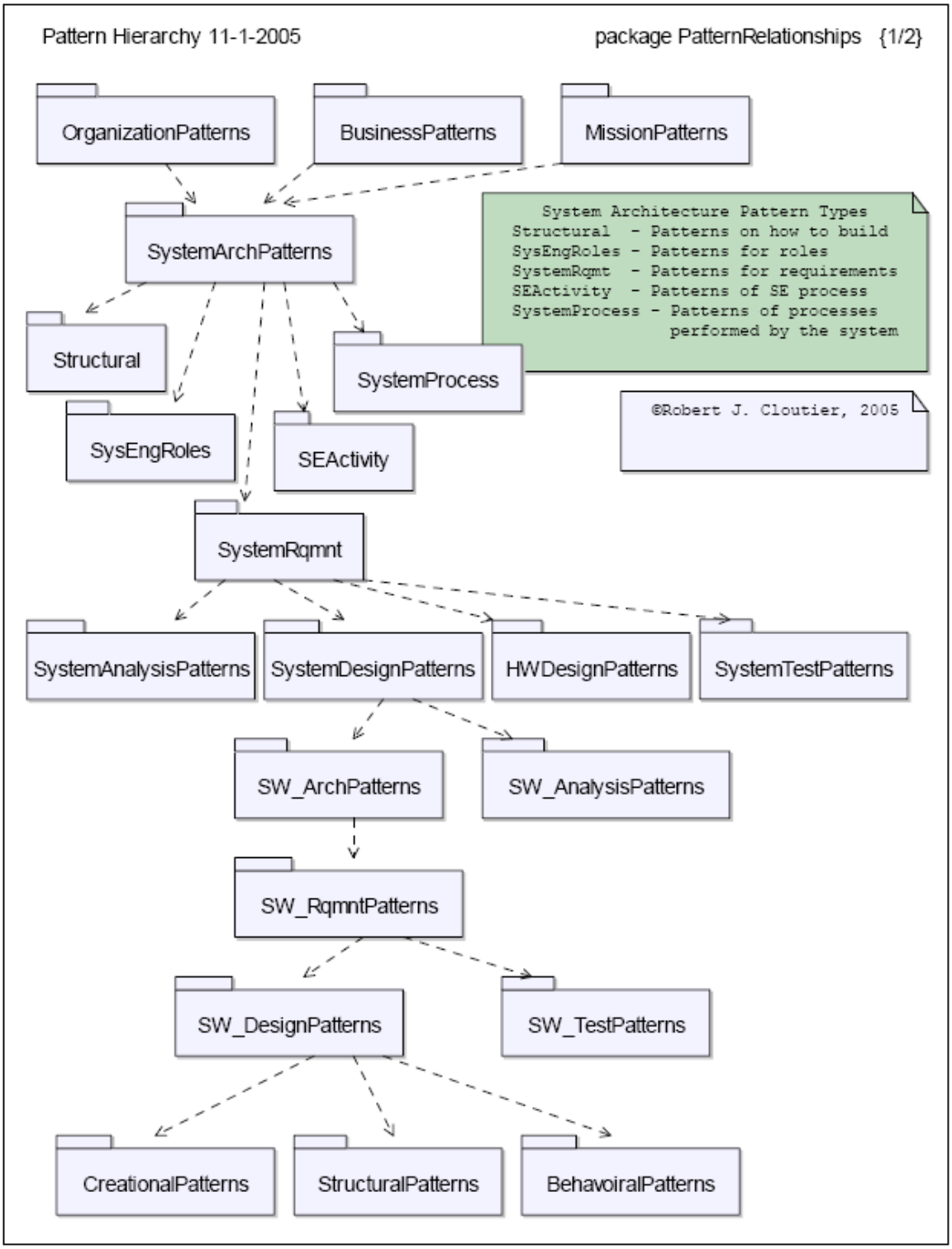

Figure 1 - Proposed System Pattern Hierarchy (Ref. [4])

2.3 Process Patterns

As described by Ambler (Ref. [3]), process patterns are a collection of general techniques, actions, and/or tasks/activities for developing object oriented software. This definition can be applied to capability engineering in the development of systems and system of system architectures. According to Ambler there are three types of process patterns: Task; Stage; and Phase. Although the majority of Ambler's focus on patterns is related to his interest in using them to support software development processes, these concepts can be applied to Capability Engineering and the CEP. The CEP is a set of process patterns for architecting complex system of system architectures. By extension, a CEP Task is a Task Process Pattern, as discussed further below.

Ambler defines Task process patterns as those process patterns that depict the detailed steps to perform a specific task, such as the Technical Review and Reuse First process patterns. In comparison, the CEP describes task process patterns where the inputs describe the context, the steps describing each task is the process pattern that generate associated deliverables/output of the task. For example, in the CEP, Initiate Requirements Analysis is a task process pattern.

Ambler next considers Stage process patterns as those process patterns that relate to the steps, which are often performed iteratively, of a single project stage. A project stage is a higher-level form of process pattern, one that is often composed of several task process patterns. Similarly, the CEP Process Analyse Capability Gap defines a stage process pattern according to Cloutier's definition above. This process pattern describes all the activities and tasks to be performed during this stage.

Finally, Ambler considers Phase process patterns as those that demonstrate the interactions between the stage process patterns for a single project phase, such as the Initiate and Delivery phases. Project phases are performed in a serial manner while stage process patterns are performed iteratively. In the CEP lifecycle, stages are equivalent to the Phase process patterns described above: Inception, Comprehension, Elaboration, and Completion. For example, the Inception phase of a capability engineering initiative can consist of an Analyse Capability Gap stage process and a Define Operational Architecture stage process. Each one of these includes one or more task process patterns.

2.4 Organizational Patterns

As mentioned earlier, there is a general consensus that Process Patterns and Organizational Patterns are closely related (Ref [5]). Ambler states that in Coplien's (1995) paper "A Generative Development-Process Pattern Language," he hints at the definition for the term "process pattern" in his statement that "the patterns of activity within an organization (and hence within its project) are called a process" (Ref [3]). This relationship is further exemplified by what is known as Conway's Law, which states organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations (Ref [6]). As Coplien put it, there is, "a close tie between the technical issues of architecture and the human issues of an organizational structure" (Ref [7]).

Coplien spent a great deal of his time applying patterns to organizations, with a focus on software development organizations and projects. He made the point that organizations that do not use patterns (i.e. processes) are not as efficient as those that do and he states, “Patterns support emerging techniques in the software design community, where they are finding a new home as a way of understanding and creating computer programs. There is an increasing awareness that new program structuring techniques must be supported by suitable management techniques, and by appropriate organization structures; organizational patterns are one powerful way to capture these” (Ref [8]).

There are two aspects of organizational patterns as they apply to the CEP: the organizational structure of the CEP Team developing the capability architectures as part of a capability engineering initiative; and the organizational structure of the solution that describes the structure of the defense organizations that are integral to the system or systems as part of the solution. The CEP provides an organizational pattern that can be tailored to address the nature of the capability engineering initiative. Defense organizational structures provide set patterns that can be used and/or tailored to be part of the solutions.

2.5 Pattern Languages

According to Robert H. Barter, “A pattern language is a collection of patterns” (Ref [9], just as the English language is a collection of English words. Ref [10], written by Joseph J. Simpson and Mary J. Simpson, states that Barter was the first to propose the creation and use of a systems engineering pattern language in his paper, "A Systems Engineering Pattern Language" (Ref [2]), which appeared in the 1998 INCOSE Symposium Proceedings. Simpson and Simpson’s 2006 paper included an expanded definition from that of Barter. Their definition was, “A pattern language is a collection of interrelated patterns, with specific relationships binding individual patterns together.” This expanded definition allowed them to explore the relationships between patterns by accurately and completely defining them. Their methodology was to use the form expressed by Barter, where patterns are described as having an Evocative name and 5 contexts: problem statement, forces, context, related patterns and solution (see Figure 2).

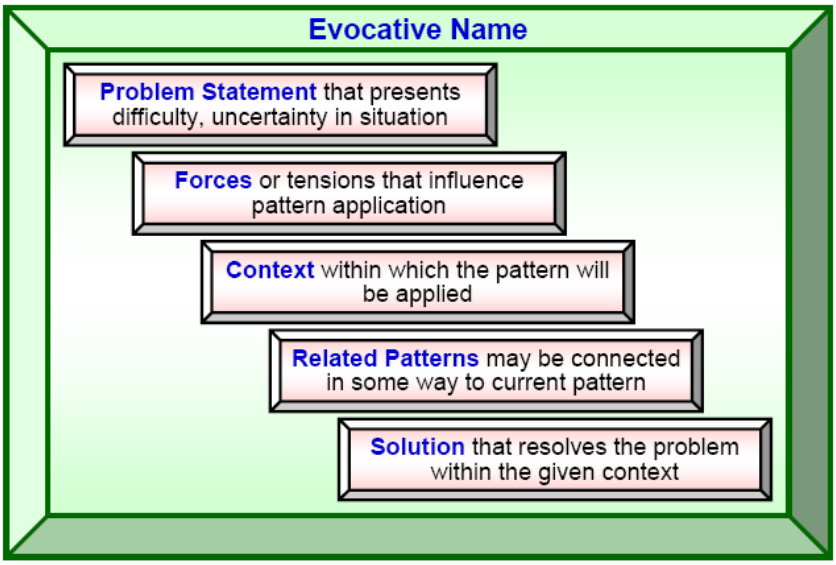

Figure 2 - Pattern description format (Ref. [10])

The following is an example which applies the CEP to DRDC applied research projects, demonstrating how one can document a pattern:

Name: Management Hammer
Forces: Actual DRDC applied research projects do not currently use a formal management procedure in the conduct of their projects. In order to provide greater rigour and management oversight of these projects and still allows them the flexibility and agility required of research projects a minimal management process is required.
Context: A tailored CEP process is applied within the context of a DRDC applied research project.
Solution: The process pattern to follow for DRDC applied research projects in the inception phase is shown in Figure 3. In the inception phase the Provide Support task is not performed. The Define Management Procedure and Define Workplan tasks are merged. Also, the Validate Requirements and Validate Operational Options tasks are merged.

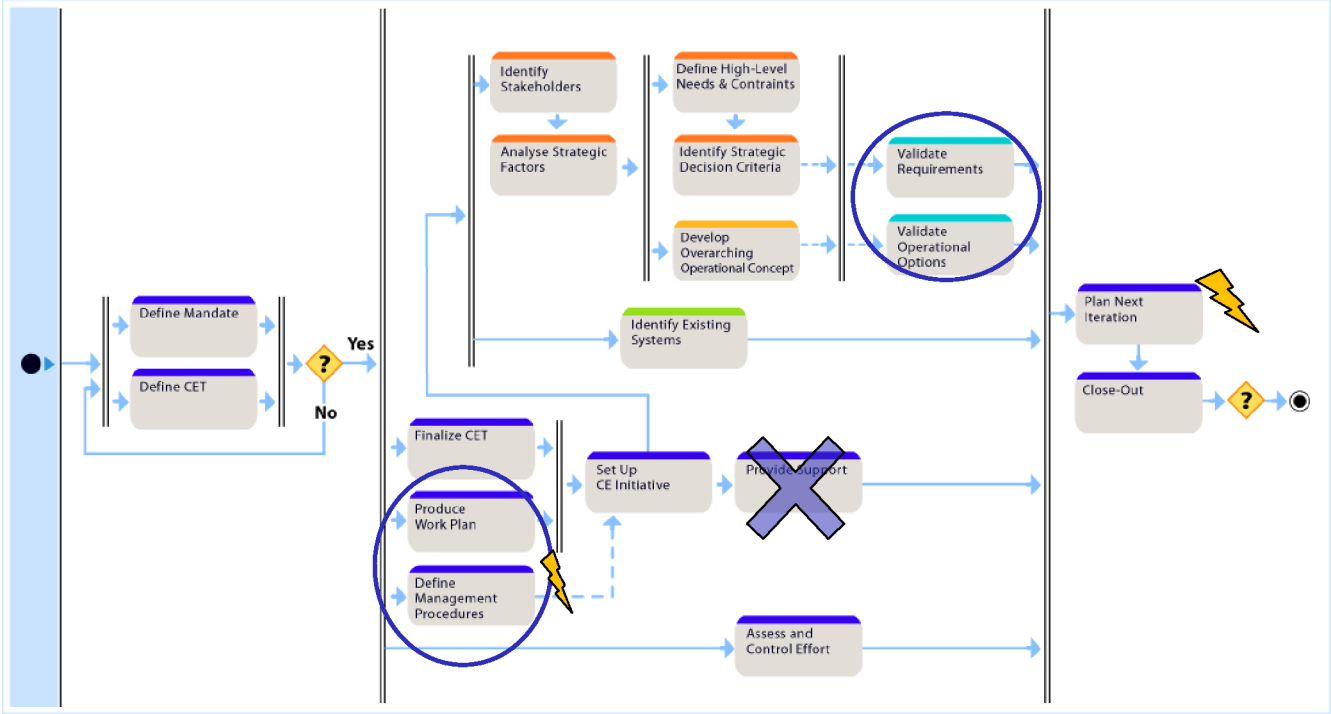

Figure 3: A process pattern example applied to DRDC applied research projects.

Simpson and Simpson also described 3 high level, global systems engineering patterns that can be used as a means of organizing patterns. Those high level systems were (1) Anything can be described as a system. (2) The problem system is always separate from the solution system. (3) Three systems, at a minimum, are always involved in any system activity: the environmental system, the product system, and the process (that produces the product) system. They recognized that extensive descriptions exist that augment these particular patterns. Their intent was to create an initial 'ontology' of high-level pattern descriptions that provide a framework within which pattern relationships can be discussed and proposed as a part of a systems pattern language.

Simpson and Simpson continued with a proposed foundational set of SE patterns for use in SE pattern language and the introduction of a collection of SE pattern relationship types. The basic SE patterns included: the Triadic Compatibility Pattern, the Requisite Parsimony Pattern, the Limits Pattern, the Gradation Pattern, the Functions, Requirements, Architectures and Tests (FRAT) Pattern, and the Context, Concept, Functions, Requirements, Architectures and Tests (CCFRAT) Pattern. The majority of these patterns were based on earlier work by John N. Warfield, who wrote extensively on systems science and related subjects (Ref [21]. The pattern relationship types were described as anywhere on a range from formal to informal. They were further discussed in terms of styles such as the Concept Map Style developed by Barter, through the use of concept map notation; the Sequential Concept Map Style, which uses concept map relationship notation and includes basic relationship types such as: top down, center out and

sequential relationship tags which were developed by McCartor and Simpson; the Hierarchical Style; the Sequential Framework Style; and the Abstraction Stack Style.

Simpson and Simpson also outlined the current state of systems engineering pattern literature. Their paper suggests that SE patterns and pattern languages should be based on the capture of SE best-practices, and formal SE processes as well as the Systems Engineering Book of Knowledge (SEBOK). The paper proposes using patterns and pattern languages to capture and manage the SEBOK information.

Also according to Simpson and Simpson, Cecilia Haskins also proposed the use of SE patterns later in 2003 to capture the information in the SEBOK. There was a difference between the definition and development of pattern languages between her proposal and that of Barter. Barter suggested the use of 'Concept Map'-like relationships between the individual patterns in a SE pattern language, while Haskins referred to a pattern language as a collection of patterns, and does not address the specific relationship types between patterns in any detail. In a 2005 paper, "Application of Patterns and Pattern Languages to Systems Engineering," Haskins addressed the relationship component of pattern languages in a different manner by addressing links between patterns at different levels and appealing to the ideas of networks and hierarchy to describe these links (Haskins 2005) (Ref [10]).

Given that there are virtually an unlimited number of patterns, pattern interrelationships and relationship types, Simpson and Simpson's discussion of types is not all inclusive and they conclude that, "More research is needed to further develop the core set of systems engineering patterns and pattern relationship types. This research should also cover groups of related patterns or pattern languages." (Ref [10])

At Ref [11] "A Pattern Language", the second of three books published by the Center for Environmental Structure to provide a "working alternative to our present ideas about architecture, building, and planning", Christopher Alexander offers a practical language for building and planning based on natural considerations. He gives an overview of some 250 patterns that are the units of this language, each consisting of a design problem, discussion, illustration, and solution. By understanding recurrent design problems in our environment, readers can identify existent patterns in their own design projects and use these patterns to create a language of their own. This is an extraordinarily thorough, coherent, and accessible book useful for homebuilders, contractors, and developers who care about creating healthy, high-level design.

From a software architecture perspective, "Pattern Languages of Program Design" Ref. [12], Coplien et al look at frameworks and components for engineering solutions to particular types of problems at a higher level, such as looking at patterns as "tools and materials" that can be used to solve problems effectively. Coplien went further in Ref [13], to illustrate pattern use in the real world using examples from C++ including the Command Processor for executing commands, and the Observer Pattern for building efficient components. There is also some extensive research on the Proxy pattern, which presents many extensions for a variety of network and distributed solutions. Some of these patterns could be applicable to address specific capability engineering problems to deal with network centricity.

2.6 Pattern Modeling

Pattern modeling provides a formal approach to define patterns, their behaviour and interrelationships. To date Unified Modeling Language (UML) has been primarily used to design software. Ref [16] appears to take the idea of patterns beyond software modeling and into the world of modeling business processes as well. The authors put forward the idea of using UML to model entire organizations. The argument presented is that with models, an organization can provide better software, define and implement new goals, and even decide whether to outsource certain operations. The Erickson-Penker Business Extensions for UML, invented by the authors and presented within the text, permit UML to document an entire business enterprise. Included are instructions on how to model businesses, from business architecture to processes, business rules, and goals and short case studies--for Web-centric and more traditional companies--are used to illustrate key concepts here. This approach is worthy of further investigation in supporting capability engineering design patterns.

3 Players, Conference & Publications, Research Streams, Documentation Forms and Standards

3.1 Main Academic Active Players

Discussion concerning the use of patterns to support engineering processes has been on-going for many years. Christopher Alexander who authored *A Pattern Language: Towns, Buildings, Construction* in 1977 was one of the first proponents of using patterns for house construction and renovation. In itself this seems remarkable given that the construction industry has been using blueprints as patterns for buildings and houses for decades. Wikipedia credits him as being "the father of the Pattern Language movement in computer science", (Ref [15].

Over time patterns became associated with software development. One of the most widely known individuals involved in this activity was James Coplien who, along with his contemporaries at the Cutter Consortium established the software pattern discipline from discussions at a workshop in 1994 on an earlier Advanced Research Project at Bell Laboratories called the Pasteur Project. By the fall of 1994, a collection of about 40 organizational patterns existed. Coplien presented these patterns at the first pattern conference in Allerton Park, Illinois.

Another Cutter Consortium Senior Consultant, Kent Beck, was involved in Coplien's work and he would later cite Coplien as one of three influences in his work on Extreme Programming (XP). Other organizational pattern works soon followed, including papers presented at the same conference by Norman Kerth and Bruce Whitenack that went beyond software architecture to explore the realm of human behavior. In 1995 Cutter Consortium Senior Consultant Alistair Cockburn published "Prioritizing Forces in Software Design", and Ward Cunningham published his EPISODES pattern language. The former would become one of the foundations of the AgileAlliance, and the latter lent major structure to XP. All these works had a pattern focus and emanated from the pattern community.

Robert H. Barter entered the scene in the late 1990s with his work on Systems Engineering patterns which appeared in the 1998 INCOSE Symposium Proceedings. He proposes the creation and use of a systems engineering pattern language and suggested that SE patterns and pattern languages should be based on the capture of SE best-practices, and formal SE processes as well as the Systems Engineering Book of Knowledge (SEBOK). According to Barter, Eric Gamma and Michael A. Beedle were also noted as some of the earliest investigators of using patterns for object oriented software and Enterprise Architectures in the mid to late 1990s. Additionally in the late 1990's conferences began to become the hotbeds of discussion on patterns and numerous other contributors engaged in information sharing on patterns. A group, known as the "Gang of Four", namely Eric Gamma, Richard Helm, Ralph Johnson and John Vlissides emerged in this timeframe as a group dedicated to patterns research (Ref [17]).

Robert J. Cloutier emerged as a follower of the "Gang of Four" in the late 1990s. He is associated with the Stevens Institute of Technology in New Jersey and he continues to do research and publish his ideas through the Centre for Patterns and System Engineering Website, (Ref [22]).

In 2003, Cecilia Haskins again proposed the use of System Engineering patterns to capture the information in the SEBOK. There is a difference between the definition and development of pattern languages between these two first INCOSE pattern papers. Barter suggested the use of 'Concept Map'-like relationships between the individual patterns in a SE pattern language, while Haskins referred to a pattern language as a collection of patterns, and does not address the specific relationship types between patterns in any detail. In 2004 James Coplien and Neil Harrison wrote a book on Organizational Patterns. The collection was a cohesive and broad work built on the original core patterns but that drew heavily on patterns — whether in "pattern form" or not — from the branches of the agile movement (Ref [7]). In a 2005 paper, "Application of Patterns and Pattern Languages to Systems Engineering," Haskins addressed the relationships component of pattern languages in a different manner by addressing links between patterns at different levels and appealing to the ideas of networks and hierarchy to describe these links (Haskins 2005) (Ref [10]).

A relative latecomer, Ambler has written extensively recently using all his predecessors are references. He has started a cottage industry around Patterns for Engineering applications and has compiled the most complete set of references available through the internet as White Papers or available for purchase through Amazon.Com. The emergence of design patterns and the explosion of information available on the Internet have given rise to a number of individuals interested in patterns and their uses, particularly for software development. In July 2006 Ambler joined IBM Canada as the Chief Methodologist/Agile for IBM Rational. He works out of the Toronto Labs with IBM clients, in particular for executive coaching, management briefings, and software process improvement efforts (Ref [23]).

Simpson and Simpson also emerge in the 2004-5 timeframe as investigators of various pattern styles and types. Their work, building on that of Barter, Haskins and Warfield and published in 2006, is some of the more complex, but interesting literature on patterns and pattern languages and a possible relationship to what could be referred to as capability engineering process patterns.

3.2 Main Conferences and Publications

Conferences

INCOSE (Ref [18]) Symposiums which have been taking place for 20 years are the most often quoted conferences on patterns. The overall focus of the symposiums is on systems engineering, however presentations and discussions on patterns, pattern languages, etc have taken place numerous time over the years.

EuroPLoP (Ref [19]) is another often quoted conference event, which is in its 14th year. Michael Weiss (weiss@sce.carleton.ca) of Carleton University was listed as a chair of the recent July 2009 Conference in Bavaria, Germany and an oranizer of the next event scheduled for July 2010. There is also an annual Scandinavian pattern conference known as VikingPLoP which last took place in September 2008 (Ref [1]). The Hillside Group, who sponsors EuroPLoP and VikingPLoP also sponsors a number of other annual conferences all dealing with various themes related to patterns (Ref [24]).

Publications

As stated throughout the paper, there are a number of publications available on patterns in general, process patterns, organizational patterns, pattern languages and pattern modeling. The most accessible and up to date set of documents on patterns appears to be at the Hillside website (Ref [24]). It contains links to books, articles, papers and research on patterns as well as a catalogue with links to other repositories.

There are also a number of books on Patterns, Pattern Languages, Pattern Design, etc all available through Amazon.com.

3.3 Main Research Streams & Results

There is no main stream of research on patterns. For example, the papers submitted to EuroPLoP (Ref [19]) over the years have covered a wide range of subjects, from technical issues, like Java coding techniques, to social and organizational issues such as environmental damage. In previous years, papers have clustered around subjects such as:

- Software development: design, architecture, management and processes
- Human computer interaction (user-interface aspects and novel modes of interaction)
- Pedagogy and Education (both professional training and classroom teaching)
- Business and organizational questions

The majority of research on patterns appears to be focused on patterns as they relate to software development. There is a large body of earlier research done on patterns related to construction and manufacturing, pattern languages and organizational patterns, whereas more recent research is on pattern modeling and the use of UML to document patterns and pattern languages.

3.4 Main Pattern Documentation Forms & Standards

There is no set format or convention for a pattern or a pattern language in the system engineering context. The forms used in the presentation of patterns for system engineering vary from simple textual documents with tables, to spreadsheets and ultimately to the UML approach discussed above. The form at Ref [20] developed by Cloutier is one example of a form associated with Patterns and System Engineering.

Also, Ambler discusses how to document a process pattern at Ref [5]. He states, “I believe that there is a need to combine the existing work in process patterns, well defined (yet still evolving) within the patterns community, with that of the existing process improvement community. The implication is the need for a format/template that both communities understand and agree to. In that light, immediately below is one way that a process pattern can be documented.

Process Pattern Format:

- Name. Provide a concise, strong name for the pattern, such as Program or Reuse First.
- Intent. Describe the process pattern in one or two paragraphs, providing if applicable a graphical description of the process pattern. Common graphical notations applied to process patterns include, but is not limited to, flow charts, process diagrams, UML activity diagrams, and data-flow diagrams.
- Type. Indicate if it is a Task, Stage, or Phase process pattern.
- Initial Context. Indicate the situation to which the pattern solution applies, and if applicable the entry conditions that must be true before the process may begin.
- Solution. Describe in detail how to perform the steps/activities of the process pattern. You may also choose, particularly for phase and stage process patterns, to describe management, quality assurance, and risk management issues, as well as indicate potential metrics to collect when working the process.
- Resulting Context. Indicate the situation/context which will result from performing the process pattern solution, including if applicable the conditions that must be true for the process to be considered complete.
- Related Patterns. Indicate the other patterns that this pattern is composed of, is a part of, or is associated to.
- Known Uses/Examples. Indicate where/how the process pattern has been applied in use. For example, the Technical Review task process pattern can be applied to the management of peer reviews, code reviews, model reviews, and management reviews."

In a presentation made by Dr. Cloutier at INCOSE – Orlando Chapter in February 2009, there is a proposed high-level process to document, retrieve and adapt Process Patterns to specific Engineering problems Ref. [25]. This process is illustrated in Appendix A.

4 Conclusions

Patterns have been examined, used and reused by mankind for various functions since the beginning of time. Recently patterns have emerged as a means of assisting software development processes to reduce time, cost and presumably to increase profit margins. Patterns are also now used to model organizations and gain insight into how organizational structures affect outcomes. In this sense patterns are already an intrinsic part of the Capability Engineering Process. Since the number and types of patterns is limitless, the opportunity for study and discussion of pattern languages and pattern relationship types also knows no bounds. There are emerging uses of patterns, such as the use of UML to document organizations that are worthy of further examination and possible application in a CEP environment.

The CEP is a set of process patterns for architecting complex system of system architectures. By extension, a CEP Task is a Task Process Pattern, as discussed in this document. Depending on the nature of the capability initiative to be addressed, the budget, the timeframe and resources available will determine the set of process tasks of the CEP that will be performed. Identifying the characteristics of these variables and how they affect or are affected by each CEP Task becomes a critical function. Once these relationships are understood then decisions can be made on the combination of CEP tasks to be applied. The challenge is to identify the relevant characteristics of each variable that can be used to decide on the CEP process pattern to be used.

References

[1] Seventh Nordic Pattern Languages of Programs Conference Web Page, http://vikingplop.org/

[2] Robert H. Barter, "A Systems Engineering Pattern Language." Proceedings of the 8th Annual International Symposium of the International Council on Systems Engineering (Vancouver, BC, July, 1998).

[3] Scott W. Ambler, An Introduction To Process Patterns, AmbySoft Inc, June 1998, http://www.ambysoft.com/essays/processPatternsPaper.html;

[4] Robert J. Cloutier, "Application of Patterns to Systems Architecting", Telelogic - Lockheed Martin & Stevens Institute of Technology, December 2005, http://www.calimar.com/papers/wp_application_of_patterns.pdf

[5] Scott W. Ambler, The Process Patterns Resource Page, http://www.ambysoft.com/processPatternsPage.html

[6] Conway's Law, http://en.wikipedia.org/wiki/Conways_Law

[7] James O. Coplien et al, Organizational Patterns: Building on the Agile Pattern Foundations, Cutter Consortium, 2005, http://793481125792299531-a-gertrudandcope-com-s-sites.googlegroups.com/a/gertrudandcope.com/info/Publications/Patterns/Process/OrganizationalPatterns.pdf

[8] James O. Coplien et al, Organizational and Process Patterns, AT&T, March 1995, http://sites.google.com/a/gertrudandcope.com/info/Publications/Patterns/Process/PLoPD1/all-patterns

[9] Bob Barter, A Systems Engineering Pattern Language, August 1998, http://www.incose.org/sfbac/1990sevents/980811-sepl-presentation.pdf

[10] Joseph J. Simpson & Mary J. Simpson, "Foundational Systems Engineering (SE) Patterns for a SE Pattern Language", System Concepts, 2006, http://www.eskimo.com/~jjs-sbw/SE_Pat_IS06.pdf

[11] Christopher Alexander et al, "A Pattern Language" , Oxford University Press, 1977, ISBN 0-195-01919-9, http://www.amazon.com/Pattern-Language-Buildings-Construction-Environmental/dp/0195019199 ;

[12] James O. Coplien et al, "Pattern Languages of Program Design", Addison-Wesley, May 1995, ISBN 0-201-60734-4, http://www.amazon.ca/Pattern-Languages-Program-Design-Coplien/dp/0201607344 ;

[13] James O. Coplien et al, "Pattern Languages of Program Design 2", Addison-Wesley, June 1996, ISBN 0-201-89527-7, http://www.amazon.com/Pattern-Languages-Program-Software-Patterns/dp/0201895277/ ;

[14] Design Pattern, http://en.wikipedia.org/wiki/Design_pattern

[15] Design Patterns, http://wiki.fluidproject.org/display/fluid/Design+Patterns

[16] Magnus Penker and Hans-Erik Eriksson, Business Modeling With UML: Business Patterns at Work, http://www.amazon.com/Business-Modeling-UML-Patterns-Work/dp/0471295515/ref=sid_dp_dp

[17] Erich Gamma et al, Design Patterns: Elements of Reusable Object-Oriented Software, 1994, http://en.wikipedia.org/wiki/Design_Patterns_(book)

[18] International Council on Systems Engineering (INCOSE) Website, http://www.incose.org/

[19] European Conference on Pattern Languages of Programs, http://hillside.net/europlop/

[20] Robert J. Cloutier, Sample Form for Documenting Patterns, Center for Patterns and Systems Engineering, Stevens Institute of Technology. [http://www.patterns4se.com/pattern_form.doc];

[21] John N. Warfield, An Introduction to Systems Science, World Scientific Publishing Company Ltd, http://books.google.ca/books?id=MmX0TnVBLKAC&pg=PA256&dq=warfield+design&ei=l3m_So6bFo_-ywTB4JTGDw#v=onepage&q=warfield%20design&f=false

[22] Robert J. Cloutier, Sample Form for Documenting Patterns, Center for Patterns and Systems Engineering, Stevens Institute of Technology. [http://www.patterns4se.com/];

[23] Scott W. Ambler, [http://www.ambysoft.com/services/contactUs.html]

[24] Hillside Conference Site, http://hillside.net/europlop/HillsideEurope/index.htm/

[25] Robert J. Cloutier, Patterns and System Architecture, INCOSE Orlando Chapter, February 2009. http://www.incose.org/orlando/Attach/200902/Cloutier%20Patterns%20Orlando%20INCOSE%20-%20no%20backup.pdf

This page intentionally left blank.

Annex A Pattern Form

On his *Center for Patterns & Systems Engineering*, Dr. Robert J. Cloutier of STEVENS Institute of Technology provides a template to document Process Patterns.(shown below) [http://www.patterns4se.com/pattern_form.doc];

Pattern Name	The name of the pattern should be descriptive to enable the pattern user to understand the usage	
Aliases	Other names by which the pattern may be known	
Keywords	Keywords which assist in locating appropriate patterns when needed	
Problem Description	Brief discussion of the constraints the pattern may impose	
Problem Context	What is the problem this pattern can be used to solve	
Forces	Challenges that exist in the problem being addressed by the pattern, and the problems in applying the pattern	
Pattern Solution	Discussion on how the pattern solves the problem being addressed	
Sketch	This can be one or more diagrams necessary to represent the pattern. Please attach files to email.	
Interfaces	Discussion of the critical interfaces or information flows necessary in implementing the pattern	
Resulting Context	What are the unaddressed issues remaining when the pattern is applied/used	
Example	An example of how the pattern may be applied	
Pattern Rationale	Why the pattern works	
Known Uses	Where else is the pattern being used in other places or applications	
Related Patterns	Other patterns what may work in conjunction or in association with this pattern	
References	Other information that may be useful in understanding or applying the pattern	
Authors	Who identified and documented the pattern	

This page intentionally left blank.

Distribution list

Document No.: DRDC Valcartier CR 2010-012

LIST PART 1: Internal Distribution by Centre

1 Christophe Necaille
2 Bibliothèque

3 TOTAL LIST PART 1

LIST PART 2: External Distribution by DRDKIM

1 Library and Archives Canada

1 TOTAL LIST PART 2

4 TOTAL COPIES REQUIRED

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

CGI Group
275 Slater
Ottawa, ON

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

State of the Art Report on Patterns in Systems Engineering and Capability Engineering:

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Bagnulo, A.; Addison T.

5. DATE OF PUBLICATION (Month and year of publication of document.)	6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)	6b. NO. OF REFS (Total cited in document.)
March 2010	30	25

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contract Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – Valcartier
2459 Pie-XI Blvd North
Quebec (Quebec)
G3J 1X5 Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)
	W7701-083296

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
	DRDC Valcartier CR 2010-012

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

This document presents the results of a survey of open literature on the following subjects:

- Patterns in general;
- Patterns for Systems Engineering (SE), Systems of Systems Engineering (SoSE), and Capability Engineering (CE);
- Process Patterns;
- Organizational Patterns Pattern Languages;
- Pattern Modeling Main Academic active Players in the domain;
- Main Conferences and Publications within the domain;
- Main Research Streams and Results;
- Main Patterns Documentation Forms & Standards.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Patterns, Systems Engineering, Capability Engineering

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE R&D DÉFENSE

www.drdc-rddc.gc.ca