

A Decision Support System for Enterprise Engineering

Ovidiu Noran

Griffith University Australia, School of ICT

O.Noran@griffith.edu.au

Abstract. This paper describes the design principles of a system aimed to support decision-making in Enterprise Engineering (EE) projects. The system proposes a novel approach based on the analysis of the interactions between project participants in the context of their life cycles. An outline of the theoretical foundation is followed by the requirements, architectural and detailed design descriptions of the proposed support system.

1 Introduction

The business environment is undergoing a fundamental change. Underpinned by the fast evolving capabilities of the communications infrastructure, business opportunities are becoming accessible to worldwide tenders. Cooperation and competition have gathered a global dimension; more than ever before, businesses must become agile and ready to cooperate irrespective of location (such as by forming Virtual Organizations) and in a timely manner, in order to meet project bidding scope and deadlines. Typically however, the tasks involved in creating or evolving organizations (an endeavor often called Enterprise Engineering (EE)) are complex, potentially lengthy and often ill-defined. Therefore, EE project managers must promptly understand the current situation, clarify and choose the future alternative, grasp the scope and content of the change process required, gather knowledge of useful artifacts, and importantly, adequately communicate and justify their way forward to the organization(s) involved. Current systems such as Executive Dashboards, based on the Executive Information Systems principles as described by Volonino and Watson (1990; 1991), focus more on presenting existing data rather than on actively assisting the decisional effort.

This paper describes the principles of a system aimed to support decision-making in EE projects by using the analysis of the interactions between EE project participants in the context of their lifecycles. The system elicits and transforms the tacit domain know-how of the project stakeholders into new knowledge. Mainstream Architecture

Framework (AF) elements and supporting tools are then used by the system to model this knowledge and help the participant organization(s) understand and internalize it.

2 The Theoretical Base of the Support System

The foundation of the proposed support system originates in research attempting to assess the feasibility and define the principles of a method to create methods (meta-methodology) guiding specific projects in the Collaborative Network (CN) and Virtual Organization (VO) areas (Noran 2004a; 2004b; 2005b; 2006a). The main result of the meta-methodology application to a specific project is a method expressed in an activity model depicting tasks necessary to accomplish that project.

Importantly, the research has found that in order to be able to describe the necessary tasks, the project stakeholders must first understand the problems of the current state, reason about possible ways to solve them and select the optimal solutions. This requires modeling of additional aspects using appropriate views, formalisms, reference models (RMs) and modeling frameworks (MFs). Thus, in fact the meta-methodology assists stakeholders in using their (often tacit) domain knowledge (Figure 1 left) to infer new facts, within a guided process of business modeling (Kalpic and Bernus 2006). This new knowledge is explicit (expressed in the models created) and used primarily to accomplish the project. In addition, it is internalized by other stakeholders and reflected in future decision-making, thus completing the knowledge lifecycle within the participant organization(s).

2.1 Stages and Sub-steps

As can be seen from Fig. 1, currently the meta-methodology comprises three major stages and a set of sub-steps. In the first stage, the user is prompted to create a list containing entities of interest to the project in question, making sure to include project participants, target entities (organizations, other projects) and importantly, the EE project itself. The second stage comprises the creation of business models showing the relations between the previously listed entities in the context of their lifecycles, i.e. illustrating how entities influence each other within each life cycle phase. The third stage assists the user in inferring the set of project activities by reading and interpreting the previously represented relations for each life cycle phase of the target entities. The activities must be detailed to a level deemed as comprehensible (and thus usable) by the intended audience. This task is assisted by other models and artifacts built and adopted during the second stage (as further exemplified).

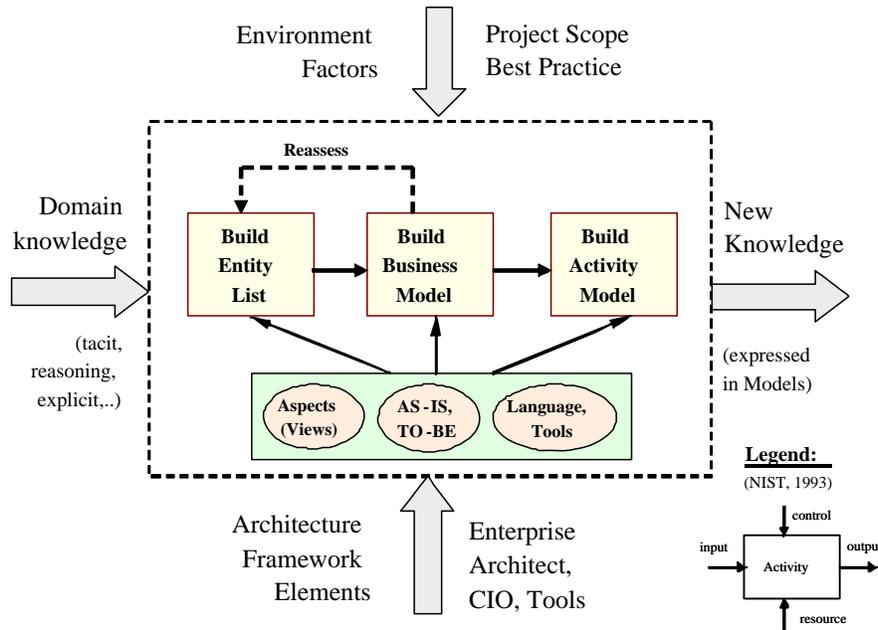


Fig. 1. Simplified meta-methodology concept

The first meta-methodology sub-step calls for the selection of suitable aspects (or views) to be modeled in each stage; the life cycle aspect must be present since it is essential to the meta-methodology. The selection of a MF is also recommended, as MFs typically feature structured collections of views that can be used as checklists of candidate aspects and their intended coverage. This sub-step also calls for the identification and resolution of any aspect dependencies.

The second sub-step asks the user to determine if the present (AS-IS) state of the views adopted previously needs to be shown and whether the AS-IS and TO-BE (future) states should be represented in separate or combined models. The third sub-step requires the selection of suitable modeling formalisms and modeling tools for the chosen aspects.

Note that all the above-described stages and sub-steps have underlying logic guiding the user in the decision-making process. A sample subset of this logic (refined and implemented as rules and facts in the system detailed design) is presented in Section 3 in a simplified format.

2.2 A Structured Repository of AF elements

The meta-methodology concept relies on creating and interpreting models depicting various aspects of the EE project participants. This modeling effort requires the selection of methods, languages, tools, MFs and RMs according to specific project requirements and best practice. As such artifacts are typically structured in AFs, the question is in fact how to find suitable sets of AF components for given projects. A possible answer is to construct ranked lists of suitable AF elements and assist the stakeholders in selecting the most appropriate ones for the EE task at hand, based on their domain knowledge.

For this to be possible, AFs must first be decomposed in elements assessed on their scope, integration, dependencies, etc and organized in a coherent collection using a set of consistent criteria.

In this case, the organized pool of AF elements (hereafter called a Structured Repository, or SR), has been obtained by assessing and decomposing mainstream AFs in respect to ISO15704 Annex A - the Generalized Enterprise Reference Architecture and Methodology (GERAM) (ISO/TC184 2000), which is neutral in respect to all assessed AFs and contains most criteria necessary for AF element classification.

PERA¹, GRAI-GIM², CIMOSA³, ARIS⁴, Zachman⁵ and DoDAF⁶ have thus been decomposed using GERAM's elements (Noran 2003; 2005a) (see Fig. 2.). Other AFs, such as TEAF⁷, FEAF⁸ and TOGAF⁹ are also being analysed and mapped onto GERAM with the resulting components being added to the repository.

Previous research and testing has determined that, although originating in the CN / VO area, the meta-methodology in fact applicable to *any* EE project type, due to its open character and neutral approach towards all major AFs (Noran 2007).

¹ The Purdue Enterprise Architecture Framework (Williams, 1994)

² Graphs with Results and Activities Interrelated (Doumeings, 1998)

³ Open System Architecture for Comp. Integrated Manufacturing (CIMOSA Assoc., 1996)

⁴ ARchitecture for Information Systems (Scheer, 1992, 1999)

⁵ the Zachman Architecture Framework (Zachman, 1987)

⁶ Department of Defence Architecture Framework (DoD AF Working Group, 2004)

⁷ Treasury Enterprise Architecture Framework (Treasury CIO Council, 2000)

⁸ Federal Enterprise Architecture Framework (The CIO Council, 1999)

⁹ The Open Group Enterprise Architecture Framework (The Open Group, 2006)

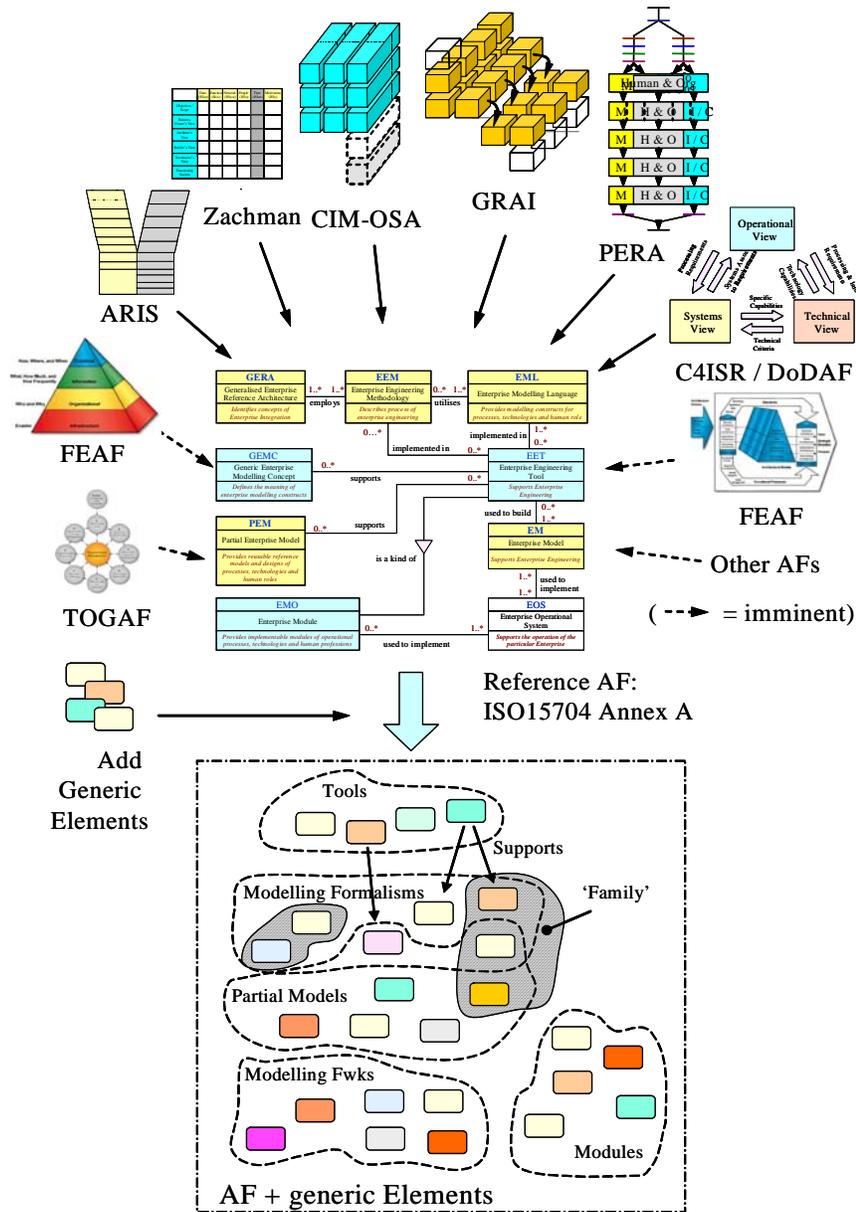


Fig. 2. Building the Structured Repository of Architecture Framework elements

3 Design of the Decision Support Framework

3.1 Identifying the Need for Support: Knowledge and Decision-making

Knowledge is present in all organizations in various forms (explicit, tacit, descriptive, procedural, reasoning) but also continuously produced (or converted (Nonaka and Takeuchi 1995)) as a consequence of knowledge-intensive decision-making processes (Holsapple and Whinston 1996). Knowledge is an enabler of organizational agility, ensuring survival and competitiveness; thus, its capturing, representation, processing and the selection of supporting technology and infrastructure (Davenport and Prusak 1998) is a major concern of management. The use of enterprise-wide management and executive information systems, expert systems, decision support systems (DSS) and their business performance-related derivatives (e.g. balanced scorecards and executive dashboards) promotes organizational learning; this results in improved decision-making that will enrich and refine the existing body of knowledge (Klein and Methlie 1995; Power and Karpathi 1998).

EE tasks (activities) often present a semi-structured character due to insufficient information and the inherent complexity of organizational change processes. This requires EE project managers to make 'semi-programmed' decisions (Simon 1977) that are difficult to encode in a program can be facilitated by a DSS (Keen and Scott Morton 1978). This obvious need for decision-making guidance and support in EE has led to the idea of materializing the knowledge acquisition, structuring and transformation capabilities of the meta-methodology in the form of a decision support system for EE that would help stakeholders such as enterprise architects, project managers and Chief Information Officers identify problems, find suitable solutions and define change processes to implement them.

3.2 Defining System and User Requirements

Successful support systems initially aimed at the executive level are likely to gain increasing acceptance and gradually expand into an enterprise-wide system embraced at several organizational levels (Wheeler et al. 1993). This scenario suggested designing scalability into the proposed system (Power 2002) and considering the possibility of future (semi-)automated learning to alleviate the anticipated knowledge acquisition bottleneck (Fogel et al. 1993).

EE decision-making involves problem solving in addition to using well-defined procedures. This mandates a cooperative and interactive support system, allowing the users to address the part of the problem that cannot be structured and to use their own insights to modify and/or refine the solutions proposed by the system (Turban, 1995). In this specific case, the users must be able to review and accept / override the ordering in the ranked lists of modeling elements suitable for the EE problem at hand. Interactivity and cooperation promote acceptance of the system; equally im-

portant, they put to use natural knowledge management skills and talents that cannot be programmed into a machine (Holsapple and Whinston 1996).

Finlay (1994) has defined some basic requirements for DSS-es, which are reflected in the proposed system's functionality: help detect existing problems, model a problem situation to clarify it, provide the means to consider various options (e.g. via simulation) and help with implementation of change (by assisting in the modeling of the TO-BE states). The system must also allow the user to bookmark important decision points so as to be able to backtrack if the decisions made have lead to unsatisfactory outcomes. This would allow the analysis of various scenarios and cycling until an acceptable solution is obtained (Hättenschwiler 1999).

Other typical DSS requirements such as robustness and ease of use and control (Little 1970) can be satisfied here by using tested off-the shelf (but highly configurable) components providing most necessary services, such as web-based shells. This allows focusing the effort towards the knowledge repository development and user interface improvement, rather than on the intricacies of implementation. The knowledge captured by the system should be easily accessible for debugging and upgrading – thus preferably stored in plain text format.

3.2 The Concept and Architectural Design of the Proposed System

The close connection between EE decision-making and knowledge management has suggested a rule-oriented knowledge-based paradigm (Holsapple and Whinston 1996) for the proposed support system. The adopted structure builds on previous mainstream research in this area. Thus, essential elements of knowledge-based DSS frameworks described by Sprague (1980), Sprague and Carlson (1982) and Marakas (1999) are present in the design of this system – such as a data base (a rule-based knowledge base (KB), if database elements are represented as facts and rules), a model base (RMs of the AFs or abstracted from previous projects) and a dialog generation mechanism (possibly provided by the inference engine).

Figure 3 presents the KB approach adopted for the SR described in Section 2.2, containing rules for element selection (e.g. via pattern matching) and ordering (for ranked lists), fixed facts (e.g. AF element representations) and other rules containing the logic necessary for system operation. Typically, rules take a form similar to IF / THEN statements (implemented depending on the specific inference engine used).

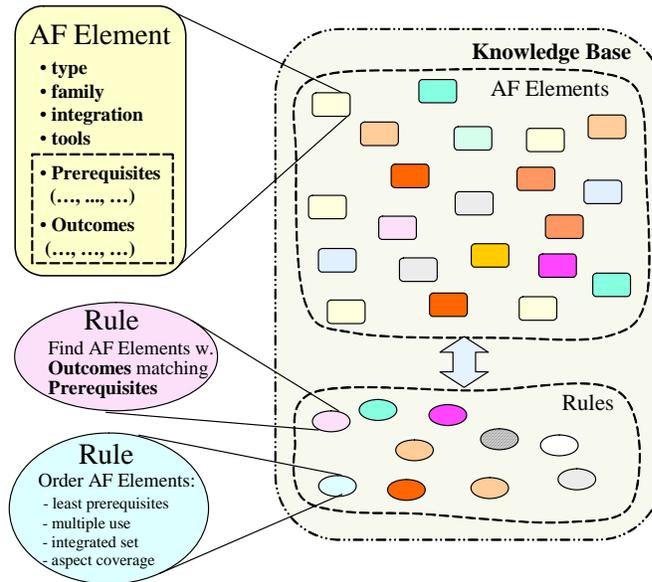


Fig 3. Structured Repository in Knowledge Base format

For example, the rule to decide whether to model the present state (AS-IS) in an EE project could take the following form:

```
IF ((TO-BE_obtained_from_AS-IS) OR (AS-IS_not_understood))
    THEN (model_AS-IS) (1)
```

Thus, the ‘model the present state’ rule will fire if the user asserts the fact that the AS-IS is not understood or the fact that the TO-BE will be obtained from the AS-IS. Another example:

```
IF (undecided_TO-BE) THEN (several_TO-BE) (2)
```

and as a likely consequence:

```
IF (several_TO-BE) THEN (separate_AS-IS_TO-BE) (3)
```

Thus, if the TO-BE will be derived from the AS-IS (e.g. no radical changes are to be made) or if stakeholders do not have a clear and common view of the future state of the enterprise and assert this at run-time (1), then several TO-BE states will be required (2) and it is likely that they will need to be shown separately from the AS-IS state (3). Note that although (3) is a recommended (‘best-practice’) consequence of

(2), the user can override (3) if there is a compelling reason to do so. Generally, various levels of interactivity and control can be made available so as to match the needs and skill level of the user.

The system will provide assistance in all essential decision-making phases (Simon 1977): intelligence gathering (via AS-IS modeling), design (via TO-BE modeling of several scenarios), choice (via simulation) and review (via the analysis of scenario results). The system will seek to apply best practice (using rules and ranking the facts) and consider user insights (using run-time asserted facts) to construct a set of solutions, ordered by their suitability and presented in a form that will facilitate decision-making (Mallach 1994).

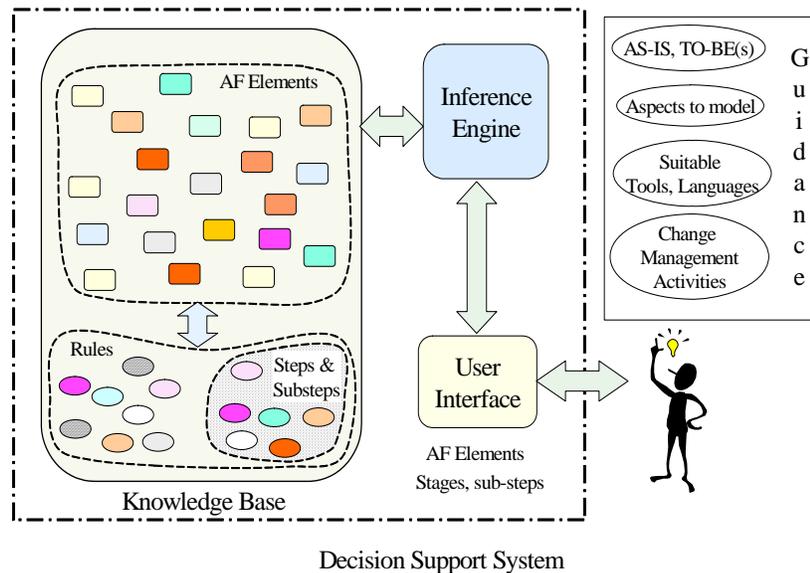


Fig. 4. Decision support system architecture and possible outcomes of a consultation

The stages and sub-steps of the underlying meta-methodology (see Fig. 1) have been incorporated as rules in the KB, as shown in Fig. 4, left. Thus, in the current form, the meta-methodology stages, sub-steps and logic required for their proper operation are modeled by rules, the AF elements are represented by fixed facts, while user decisions (accept / override system recommendations) and motivations and other input are modeled via run-time asserted facts. As expected, the solutions are to be built by a) rules implementing stage and sub-step logic, firing according to run-time asserted facts and b) pattern-matching routines, ensuring that all dependencies of the fixed facts representing recommended and/or selected AF elements are satisfied.

3.3 Detailed Design: Main Steps of the Proposed Support System

Due to space constraints, detailed design descriptions have been limited to the components reflecting the original meta-methodology stages and sub-steps described in Section 2.1. In order to facilitate the understanding of the system operation, the level of detail has been kept low and plain text has been used to describe rule components.

Stage One: Create a List of Entities relevant to the EE Project

System Explanation / Recommendation (E/R): At this early stage, type and granularity consistency of the entities relevant to the project is not important. Thus, the set may contain documents, entire organizations, individuals, policies, etc. The target organization(s) (i.e. that are being created or evolved) and the EE project itself must be included in the list.

Stage Two: Business Models of Entity Relations in a Life Cycle Context

(E/R): Represent the interactions between the entities listed, in the context of their life cycles (i.e. for each life cycle phase of the target entities). Focus on the target organization(s) and on the EE project. Ideally, restrict the representation of each entity to the extent of life cycle phases that is relevant to the project in question (to keep model complexity low). In view of the knowledge gained while building the models, re-assess the relevance of each entity to the EE project and thus the actual need for its presence in the models.

Stage Three: The Set of Activities describing the EE Project Accomplishment

(E/R): Create a set of activities for each life cycle phase of the EE project and of the entities to be created / evolved by it. Describe the inputs, outputs, controls, resources, etc needed for each activity using the interactions detailed in the business models created in Stage Two. Decompose the set of activities using views chosen in previous Stages, contained in the selected MF and in the KB (in that order).

Sub-step A: Select relevant aspects and an MF:

(E/R): Unless explicitly mandated, this sub-step need not apply to Stage One. Re-use of aspects selected in previous Stages is recommended (build and display list).

Rules: Life cycle aspect mandatory. Functional aspect is mandatory in Stage Three. Select an MF a) as mandated, b) according to existing skills or c) containing most of the aspects required by the project scope. Some typical aspects: management (or control) / service (or production, depending on enterprise profile), human extent, decisional, organizational, functional, resources, informational, stakeholder, concern, hardware / software. Attempt to recommend MF according to project scope; default on the MF used to organize the original SR (the MF of GERA- Generalized Enter-

prise Reference Architecture (ISO/TC184, 2000), a component of GERAM) as most comprehensive; recommend complementary views if selected MF lacks them.

<Stakeholder input: accept / override recommendations>

Rules: Dependency / overlap check of selected aspects; warn and ask user if overlaps detected or additional aspects required. Record decision, bookmark decision points.

Sub-Step B: AS-IS required? Type of AS-IS and TO-BE representation?:

(*E/R*): Unified representation is recommended in Stage One. In Stage Three, use the same representation type as in Stage Two unless otherwise mandated.

Rules: Apply this sub-step for all aspects selected in Sub-step A. TO-BE representation is *mandatory*. AS-IS must be represented if there is no clear common understanding of the present state by stakeholders, or if the TO-BE state must be evolved from the AS-IS. Represent AS-IS and TO-BE separately if there are obvious benefits (allow discerning between various scenarios, clarity, etc).

<Stakeholder input: accept / override recommendations, make decision>

Rules: Record decision, set bookmarks.

Sub-Step C: Modeling formalisms, tools:

(*E/R*): Choose formalisms (modeling languages) that best support the aspects selected in Sub-step A and choose tools supporting these formalisms (ranked lists are provided). Within Stage One, keep things as simple as possible (e.g. text editor).

Rules: Create ranked lists of suitable formalisms and tools for aspects selected in Sub-step A, using best practice criteria – e.g. used in previous Stages (priority criterion), potential multiple use, aspect coverage (or fit) factor, part of an *integrated* set (e.g. via underlying metamodels), etc.

<Stakeholder input: accept / override recommendations>

Rules: Perform dependency check for chosen elements, record decision; bookmark.

Implementation

The main research is directed towards extending the SR and developing and testing the support system rules in real case studies until reaching a maturity level suitable for full-scale implementation. Nevertheless, prototype implementations of the system have been attempted using the expert system shells, e.g. JESS (Friedman-Hill, 2007).

Testing of the Proposed Support System

The behaviour of the proposed system can be tested before reaching the full implementation phase since most of the essential rules and facts are known at this stage. Thus, a facilitator familiar with the rules and facts present in the KB can perform the functions of the user interface and the inference engine, applying rules, asserting facts to reflect stakeholders input, building ranked lists and assisting in the creation, presentation, and interpretation of the results.

A test of the rules composing the proposed support system has been accomplished within a case study that has also validated the theoretical framework (Noran, 2006b, 2007). Space limitations do not allow presenting the practical rule application results; they will be published elsewhere.

Conclusions and Further Work

In setting up and managing EE projects, stakeholders are sometimes confronted with a complex present, an incomplete picture of the future and with quandaries of ‘what needs to be done next’, what artefacts to use, and how to clearly communicate their vision to the organisation(s) involved. This paper describes the design principles of a system aiming to guide and assist the stakeholders in using their domain knowledge to address these essential issues.

The proposed system has several distinct features in respect to other approaches. Thus, the system is underpinned by a sound and original theoretical basis, tested in several case studies. Secondly, the paradigm used by the system is based on a *life cycle*, rather than ‘snapshot’ approach, covering all relevant life cycle phases of the analysed entities rather than just a particular one. This approach is suitable for the dynamic nature of organisations. Finally, the system uses a collection of elements belonging to well-known AFs and tools, structured using a *neutral* framework. Thus, the system can be extended with elements of other AFs in order to maintain its relevance in the future.

Substantial work lays ahead for the proposed system. The system may be implemented in a myriad of ways, e.g. local or web-based, stand-alone or as a decision support add-on to existing project management systems. Irrespective of this however, the repository and associated knowledge base need to be enriched with elements of other major AFs as previously described. In addition, more case studies are necessary in order to test and refine the rules – before and after full implementation. In fact, the knowledge acquisition, rule base update and testing processes must be *on-going* for the system to reflect the state-of-the-art in EE and thus to remain an effective management tool.

References

- CIMOSA Association. CIMOSA - Open System Architecture for CIM,. Technical Baseline, ver 3.2. Private Publication, 1996
- Davenport, T., Prusak, L. Working Knowledge. Boston: Harvard Business School Press, 1998
- DoD Architecture Framework Working Group. DoD Architecture Framework Version 1.0 Available: http://www.dod.mil/cio-nii/docs/DoDAF_v1_Volume_II.pdf, 2004
- Doumeings, G. GIM, Grai Integrated Methodology. In A. Molina, et al. (Eds.), Handbook of Life Cycle Engineering - Concepts, Models and Methodologies (pp. 227-288). Dordrecht: Kluwer Academic Publishers, 1998
- Finlay, P. N. Introducing Decision Support Systems. Manchester: Blackwell publishing, 1994
- Fogel, D., et al. The impact of machine learning on expert systems, Proceedings of the 1993 Conference of the ACM on Computer Science (pp. 522-527). Indianapolis, Indiana, US: ACM Press, 1993
- Friedman-Hill, E. JESS - The Rule Engine for the Java Platform. Distributed Computing Systems. Available: <http://herzberg.ca.sandia.gov/jess/>, 2007
- Hättenschwiler, P. Neues anwenderfreundliches Konzept der Entscheidungsunterstützung. Gutes Entscheiden in Wirtschaft, Politik und Gesellschaft. Zurich: vdf Hochschulverlag AG, 1999
- Holsapple, C. W., Whinston, A. B. Decision Support Systems: A Knowledge-Based Approach. Minneapolis / St. Paul: West Publishing, 1996
- ISO/TC184. Annex A: GERAM, ISO/IS 15704: Industrial automation systems - Requirements for enterprise-reference architectures and methodologies, 2000
- Kalpic, B., Bernus, P. Business process modeling through the knowledge management perspective. Journal of Knowledge Management, 10(3), 2006
- Keen, P. G. W., Scott_Morton, M. S. Decision Support Systems: An Organizational Perspective. Reading, MA: Addison-Wesley, Inc, 1978
- Klein, M., Methlie, L. B. Knowledge-based Decision Support Systems with Applications in Business. Chichester, UK: John Wiley & Sons, 1995
- Little, J. D. C. Models and Managers: The Concept of a Decision Calculus. Management Science, 16(8), B466-485, 1970
- Mallach, E. G. Understanding Decision Support and Expert Systems. Illinois: Richard D. Irwin, Inc., 1994
- Marakas, G. M. Decision support systems in the twenty-first century. Upper Saddle River, N.J.: Prentice Hall., 1999
- NIST. Integration Definition for Function Modelling (IDEF0) (183: Federal Information Processing Standards Publication): Computer Systems Laboratory, National Institute of Standards and Technology.1993
- Nonaka, I., Takeuchi, H. The Knowledge-Creating Company. How Japanese companies create the dynamics of innovation. Oxford: Oxford University Press, 1995
- Noran, O. A Mapping of Individual Architecture Frameworks (GRAI, PERA, C4ISR, CIMOSA, Zachman, ARIS) onto GERAM. In P. Bernus, et al. (Eds.), Handbook of Enterprise Architecture (pp. 65-210). Heidelberg: Springer Verlag, 2003
- Noran, O. A Meta-methodology for Collaborative Networked Organizations. Doctoral Thesis, School of CIT, Griffith University, 2004a

- Noran, O. A Meta-methodology for Collaborative Networked Organizations: A Case Study and Reflections. In P. Bernus, et al. (Eds.), *Knowledge Sharing in the Integrated Enterprise: Interoperability Strategies for the Enterprise Architect*. Toronto / Canada: Kluwer Academic Publishers, 2004b
- Noran, O. An Analytical Mapping of the C4ISR Architecture Framework onto ISO15704 Annex A (GERAM). *Computers in Industry*, 56(5), 407-427, 2005a
- Noran, O. Managing the Collaborative Networks Lifecycle: a Meta-methodology. In A. G. Nilsson, et al. (Eds.), *Advances in Information Systems Development - Bridging the Gap between Academia and Industry (Proceedings of the 14th International Conference on Information Systems Development (ISD 2005))* (Vol. 1, pp. 289-300). Karlstad, Sweden: Kluwer Academic Publishers, 2005b
- Noran, O. Refining a meta-methodology for collaborative networked organizations: a case study. *Int. J. Networking and Virtual Organizations*, 3(4), 359-377, 2006a
- Noran, O. Using Reference Models in Enterprise Architecture: An Example. In P. Fettke, P. Loos (Eds.), *Reference Modeling for Business Systems Analysis*. Hershey, USA: Idea Group, 2006b
- Noran, O. Discovering and modelling Enterprise Engineering Project Processes. In P. Saha (Ed.), *Enterprise Systems Architecture in Practice*. Hershey, USA: IDEA Group, 2007
- Power, D. *Decision support systems: concepts and resources for managers* (Vol. 2007). Westport, Conn.: Quorum Books, 2002
- Power, D. J., Karpathi, S. The Changing Technological Context of Decision Support Systems In D. Berkeley, et al. (Eds.), *Context-Sensitive Decision Support Systems*. London: Chapman & Hall, 1998
- Scheer, A.-W. *Architecture for Integrated Information Systems*. Berlin: Springer-Verlag, 1992
- Scheer, A.-W. *ARIS-Business Process Frameworks* (3rd ed.). Berlin: Springer-Verlag, 1999
- Simon, H. *The New Science of Management Decision* (3rd ed.). Englewood Cliffs, NJ.: Prentice-Hall, 1977
- Sprague, R. H. A Framework for the Development of Decision Support Systems. *Management Information Systems Quarterly*, 4(4), 1-26, 1980
- Sprague, R. H., Carlson, E. D. *Building Effective Decision Support Systems*. Englewood Cliffs, NJ: Prentice-Hall, Inc, 1982
- The CIO Council. *Federal Enterprise Architecture Framework (v1.1)* Available: https://secure.cio.noaa.gov/hpcc/docita/files/federal_enterprise_arch_framework.pdf, 1999
- The Open Group. *The Open Group Architecture Framework, TOGAF v8.1.1*, 2006
- Treasury CIO Council. *Treasury Enterprise Architecture Framework v.1* Department of the Treasury Chief Information Officer Council. Available: www.eaframeworks.com/TEAF/teaf.doc, 2000
- Turban, E. *Decision support and expert systems: management support systems*. Englewood Cliffs, N.J: Prentice Hall, 1995
- Volonino, L., Watson, H. J. The strategic business objectives method for guiding executive information systems development. *Journal of Mgmt Info Systems*, 7(3), 27-39, 1990-91
- Wheeler, F. P., et al. Moving from an executive information system to everyone's information system: lessons from a case study. *Journal of Information Technology*, 8(3), 177-183 1993
- Williams, T. J. The Purdue Enterprise Reference Architecture. *Computers in Industry*, 24(2-3), 141-158, 1994
- Zachman, J. A. A Framework for Information Systems Architecture. *IBM Systems Journal*, 26(3), 276-292, 1987