

Ovidiu Noran

Griffith University, AUSTRALIA

O.Noran@griffith.edu.au

Collaborative Networks (CNs) enhance the preparedness of their participants to promptly form Virtual Organisations (VOs) that are able to successfully tender for large scale and distributed projects. However, the CN efficiency essentially depends on the ability of its managers to match and customise available reference models but often, also to create new project activities. Thus, given a particular VO creation project, the CN managers must promptly infer 'what needs to be done' (discover the project processes) and how to best communicate their 'justified beliefs' to the CN members involved. This paper proposes a framework for a decision support system that can help managers and enterprise architects discover / update the main activities and aspects that need to be modelled for various enterprise task types, with special emphasis on the creation of VOs. The framework content is also explained 'by example', in the context of a real-world scenario.

1 INTRODUCTION

Collaborative Networks (CNs) allow their members to promptly create virtual organisations (VOs) able to bid for projects that exceed the competencies of the individual CN participants. Although most CNs maintain pools of reference models, VO projects often require their customisation and sometimes, the creation of new specific project processes altogether. This involves understanding the current situation, choosing the right alternative from a set of plausible scenarios, planning the transition from present to future states, knowledge and choice of the available technologies and other useful artefacts and importantly, communicating and justifying the decisions to the rest of the CN organisation(s) involved. Many modern support systems such as Executive Dashboards, although based on Decision Support Systems (DSS) principles (Volonino and Watson, 1990-91), focus on presenting information, rather than on actively assisting in the decision-making process.

This paper describes a decision support system framework based on analysing the interactions between CN members involved in a (VO or not) project in the context of their lifecycles, using elements of main architectural frameworks (AFs).

2 A META-METHODOLOGY FOR COLLABORATIVE NETWORKS

Previous research (Noran, 2004a, 2004b, 2005) has attempted to find a set of steps ('meta-methodology') that would assist in the creation of process models customised for various CN projects. This forms the theoretical basis of the DSS framework.

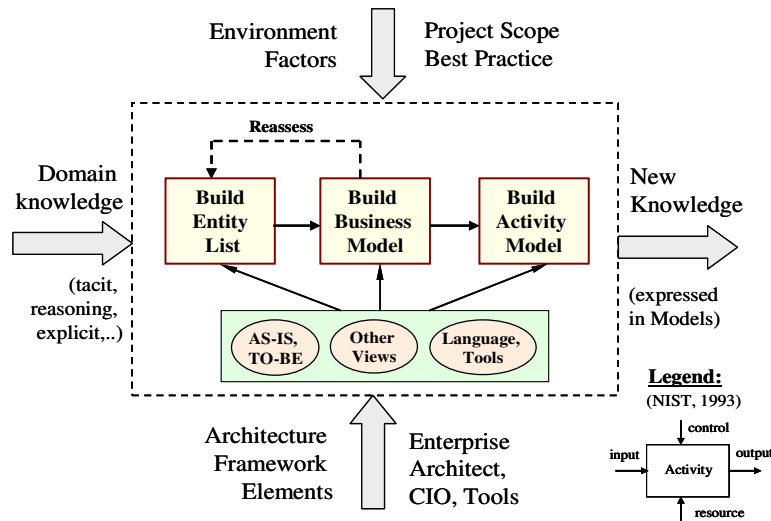


Figure 1. Simplified meta-methodology concept

The main deliverable of the meta-methodology is a model of the tasks performed in order to accomplish the VO project. However, in order to be able to obtain this model, the user has to represent several other aspects choosing appropriate tools, reference models and modelling frameworks (MFs) containing views, languages, etc. The meta-methodology requires existing domain knowledge (see **Figure 1** left) that is *transformed* into new, explicit knowledge and is eventually internalised by other stakeholders (Kalpic and Bernus, 2006) thus closing the knowledge lifecycle.

This paper describes a way to harness the meta-methodology concept, rather than to provide a full description of the meta-methodology research and development. The interested reader can find all these details in (Noran, 2004a, 2004b, 2005).

3 THE DECISION SUPPORT SYSTEM FRAMEWORK

Knowledge is continuously *produced* (or 'converted' (Nonaka and Takeuchi, 1995)) in the organisations as a consequence of decision-making processes (Holsapple and Whinston, 1996). The proposed system intends to promote this conversion and help management make 'semi-programmed' decisions (Simon, 1977) difficult to encode in a program but possible to *facilitate* by a DSS (Keen and Scott_Morton, 1978).

3.1 Requirements

A successful executive decision support system is likely to become widely used (Wheeler et al., 1993). Therefore, the system should be *scalable* from desktop to enterprise-wide level (Power, 2002). The system should also be *interactive*, allowing the decision maker to use own insights to modify the solutions provided by the system (Turban, 1995). This will promote user buy-in, ownership and the use of natural knowledge management skills and talents (Holsapple and Whinston, 1996).

The system should help detect existing problems, be able to model them for clarification, provide the means to consider options (by simulation of various scenarios until a suitable solution is obtained) and help with implementation of change (Finlay, 1994; Hättenschwiler, 1999).

Importantly, the proposed support system should deliver *guidance* in defining the change processes needed to migrate from the present to the selected future state and in *uttering* those processes and aspects in an intelligible form for the target audience.

3.2 Proposed Architecture of the Support System

The close link between decision-making and knowledge management has been manifest throughout the research pertaining to the support system requirements. This has suggested a rule-based knowledge base approach for the system repository and an expert system-type paradigm for the entire support system architecture.

The adopted concept has been assessed against previous research in the area. Thus, most knowledge-based DSS framework elements described by Sprague (1980), Sprague and Carlson (1982) and Marakas (1999) are present in the design of this system. Among other components, this system comprises a database (in the form of a knowledge base, if the components are represented as facts and rules), a model base (containing reusable reference models, which are extracted and classified from AFs and previous projects¹) and a dialog generation mechanism (provided by an inference or similar engine type see **Figure 2**).

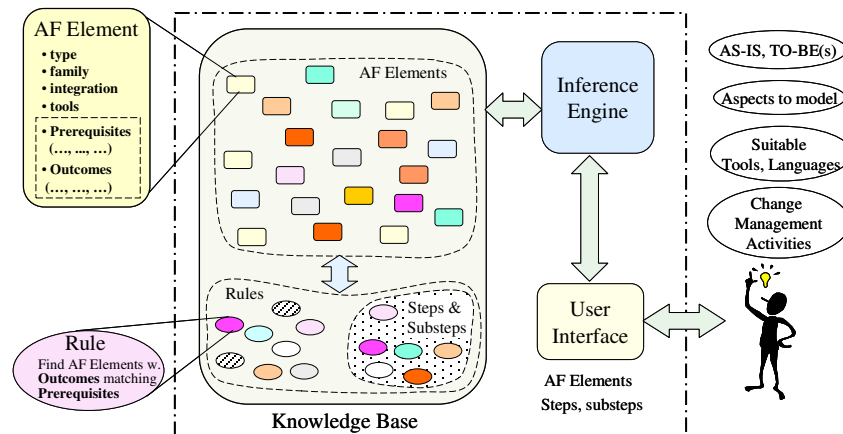


Figure 2. Decision support system architecture and outcomes of a consultation

Figure 2, left presents the rule-based knowledge base approach applied to the repository, where the rules for element selection, ordering (for ranked lists), etc share the knowledge base with the facts. The rules could take the form of typical IF/THEN-type statements specific to the inference engine involved (as exemplified in (Noran, 2005)). Simon's (1977) so-called *programmable*, *semi-programmed* and

¹ e.g. using ISO15704:2000 Annex A (ISO/TC184, 2000) as described in (Noran, 2006)

non-programmable decisions are reflected in this DSS by static / dynamic facts and by the user choices (e.g. by accepting or overriding of the highest ranked elements).

The proposed system assists in all decision-making phases identified by Simon (1977), i.e. intelligence gathering (via AS-IS modelling), design (via TO-BE modelling of several scenarios), choice (via WHAT-IF simulation) and review (through analysis of the various scenario results). The decision maker should be able to understand the alternatives, make the choice and explain the decisions to other echelons by selecting suitable modelling formalisms (assisted by the system).

The inference engine can be selected off-the-shelf (e.g. from a 'shell') as long as it meets the specifications (rule-based, platform independent, etc) and performance requirements resulting from the architectural design of the decision support system.

3.3 Main Rules of the Support System

Main Rule 1: Identify a list of entities relevant to the EA project. If projects are set up to build the target entity (entities), include them in the list;

Main Rule 2: Create a business model showing the relations between the life cycles of the listed entities. Re-assess the need for each entity in the diagram, and the *extent* of the life cycle set to be represented for each entity;

Main Rule 3: Using the life cycle diagram of each target entity, infer a set of activities describing the creation of each entity and the roles played in it by other entities. Detail the activities to the necessary level.

The following rules apply to each of the main rules:

Rule a: Identify a suitable MF if applicable/mandated. Choose the aspects to model, using the chosen MF and other aspects in the repository as checklist. Resolve aspect dependencies.

Rule b: Choose whether to represent the present (AS-IS) state. Choose whether to represent AS-IS and TO-BE states separately or combined. Model AS-IS when not fully understood or when the TO-BE will be derived from it (no radical change)

Rule c: Choose modelling formalism(s) depending on the aspect(s) selected, intended audience and modelling best-practice criteria, such as: previously used in the modelling, specialisation, prerequisites, potential multiple uses, part of a set, language set integration. Choose modelling tool depending on formalism and best practice, such as rigorously defined (preferably using a metamodel), belonging to an integrated suite, feasibility, availability, staff proficiency in that tool, etc.

3.4 Other Rules and Features

Rules are provided to resolve dependencies, seek solutions, order the AF elements (for the ranked lists) using the best practice criteria previously described, etc. Other rules provide default choices and generic elements if no preferences are entered. Examples: GERA² MF, plain English text, 'dumb' (i.e. language unaware) graphical editors, Rich Pictures. Solutions are found and dependencies are resolved using techniques specific to the inference engine selected (e.g. pattern matching, etc). The user can reject system suggestions and mark the override points for later reference.

² Generalised Enterprise Reference Architecture (ISO/TC184, 2000)

3.5 Implementation

Small-scale pilot implementations of the proposed DSS rules have been attempted using expert system shells such as JESS (Friedman-Hill, 2003). These efforts have typically involved a (thin) client-server paradigm and platform-independent applet / servlet technologies. Results have been encouraging thus far and are to be published.

4 TESTING THE FRAMEWORK: A CASE STUDY

4.1 Background and Specific Features of VO Formation

Schools A and B within the Faculty FAC wish to form a VO, called *Unified School* (US) in the TO-BE state shown in **Figure 3**. This would ensure a unique corporate image and consistency in the product delivered and the policies governing the future VO campuses at locations L1, L2 and L3. The individual campuses are set to retain much of their internal decisional and organisational structure except for the highest layer, which will be replaced by the VO governance structure. The function of CN is performed by the Faculty FAC. The lead partner for the VO project is school A. The VO is *on-going* and importantly, the partners cease to operate independently during the life of US. Since the support system is still being tested, a facilitator with knowledge of AF elements will *assist* the use of the system by the stakeholders and will note all decisions made by the system and the stakeholders. The audience of the support system deliverables is the management of U, FAC, A and B.

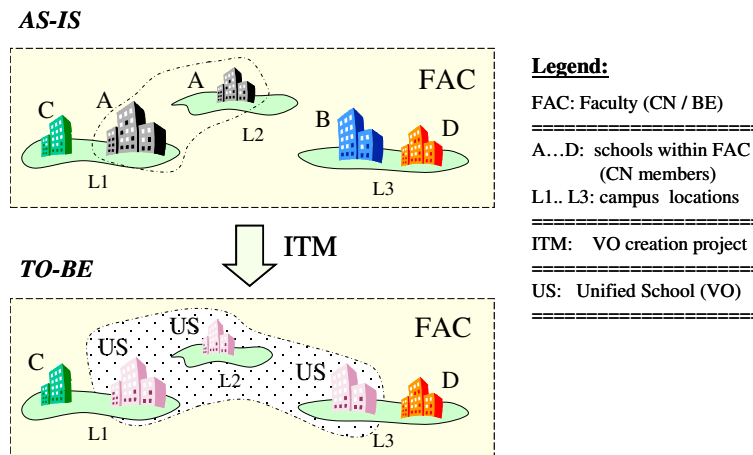


Figure 3. AS-IS and TO-BE states of the proposed VO creation project

4.2 Framework Application

Main Rule One: Identify the Entities Relevant to the VO formation

Rule a: Using stakeholders' domain knowledge, elicit relevant entities participating in the VO task; decide whether to represent a full or restricted set of life cycle phases. **Motivation:** Other aspects not relevant for this early stage. **Rule b:** Choose

to represent both AS-IS and TO-BE states in a unified representation. **Motivation:** there is no obvious gain in having two lists with most list members identical. **Rule c:** Choose text representation as the modelling formalism. Choose a plain text editor or whiteboard as 'tool'. **Motivation:** formalism and tool must preserve simplicity. The list of entities constructed in this step is shown in the legend of **Figure 4**.

Main Rule Two: A Business Model of Entity Roles in a Life Cycle Context

Rule a: adopt the GERA MF. Model the management / service, decisional and organisational aspects for the entities participating in the project. **Motivation:** no user preference for a particular framework, hence default MF chosen. The aspects selected were elicited from stakeholders from the checklist of aspects in the repository. *Life cycle representation context is mandatory.* **Rule b:** represent both AS-IS and TO-BE states. Represent management / service states combined and decisional / organisational states separately. **Motivation:** The stakeholders felt they did not fully understand the present state. No tangible advantage was seen in showing separate AS-IS / TO-BE business models; more interest shown in the decisional structure. Organisational structure was the *only representation able to discern between several TO-BE states* - hence show separate AS-IS and TO-BE. **Rule c:** choose modelling formalisms ranking highest in efficiency for the aspects selected by rule a: for life cycle and management / service, choose a GERA MF-based formalism. Choose the GRAI³-Grid formalism for decisional and organisational aspects and a plain graphical editor for modelling. **Motivation:** GRAI-Grid ranks high in respect to other languages due to its specialisation, potential multiple use (e.g. organisational aspect) and lack of prerequisites. Recommended modelling tool (IMAGIM (GRAISoft, 2002) overridden by user with 'plain graphical editor' due to tool complexity and lack of skills.

The business model (**Figure 4**) is now constructed using stakeholder knowledge. This illustrates entity roles in fulfilling the VO project, in the context of the life cycle and management / production (M / P divisions in **Figure 4**) aspects. Several entities influence various life cycle phases of US directly or via other entities - notably ITM, the project set up specifically to build US.

The AS-IS and TO-BE decisional and organisational models are then built to enable a better understanding of the problems that triggered the VO project (e.g. narrow/paternalistic management, improper information flows) and allow discerning between various TO-BE scenarios. These models are presented in (Noran, 2007).

Main Rule Three: The Set of Activities describing the VO Creation

This rule is performed by 'reading' the life cycle diagram of the US and ITM in the context of their relations with other entities (**Figure 4**). The set of activities obtained is decomposed using aspects selected from the chosen MF and repository, ranked and suggested by the system and approved by the user during the consultation.

Rule a: choose functional and life cycle aspects; use other views to *detail* the activities. **Motivation:** the main deliverable is an activity model, hence functional. However, to be understood and enacted, the activities must be detailed using other necessary aspects and views – here, management / service, human / machine and software / hardware aspects. **Rule b:** choose to represent AS-IS and TO-BE states, in a unified representation. **Motivation:** the activity model is expressing a *transition* from AS-IS to TO-BE, thus both should be represented. Separate views did not justify the consistency-maintaining overhead. **Rule c:** choose the IDEF language set

³ Graphs with Results and Methods Interrelated (Domeingts, 1984)

(NIST, 1993); select the AIOWin tool (KBSI, 2007). **Motivation:** The Unified Modelling Language (Rumbaugh et al., 1999) ranks higher than IDEF0 due to an underlying metamodel and wider tool support; however, availability of the AIOWin tool and IDEF0 skills have motivated the user to override the proposed ranking.

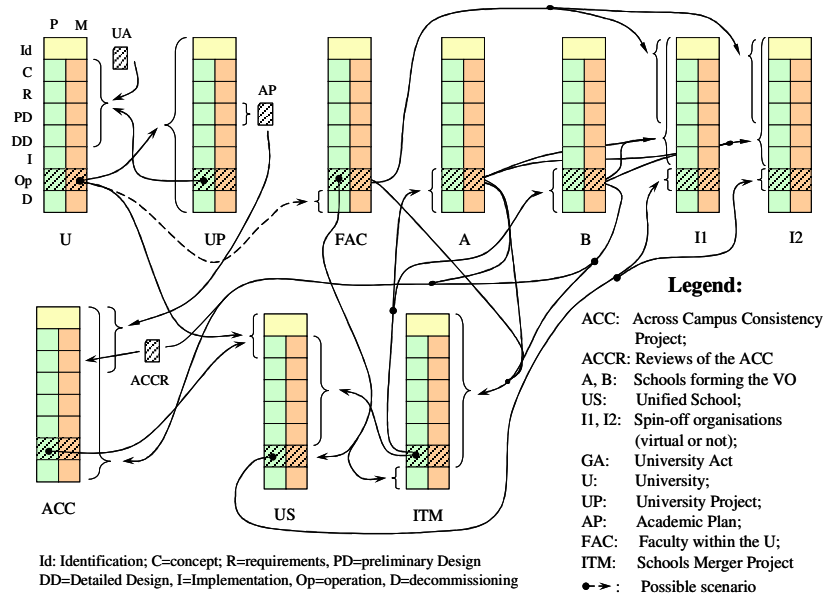


Figure 4 Business model expressing entity roles in accomplishing the EE project

4.3 Notes on the Creation of the Activity Model

The functional model can be initiated by creating one main activity for each VO (and VO setup project if applicable) life cycle phase. The modelling formalism chosen will also assist in developing the model. For example, IDEF0 requires inputs, controls, outputs and mechanisms to be defined for these activities (see **Figure 1** lower right). Therefore, the system will prompt the IDEF0 user to represent what is used in each activity, what controls it, what is produced and who / what executes it.

Each activity should then be decomposed to a level understood by the envisaged audience, showing aspects used in previous steps and / or present in the chosen MF and contained in the repository, such as human / machine, hardware / software.

Not all aspects are relevant to all life cycle phases. For example, early life cycles (e.g. Identification, Concept) require few or no aspects, and the human / automation boundaries may only be relevant in the Preliminary / Detailed Design phases.

Detailed descriptions of the activity model creation and structure, which is beyond the purpose of this paper and available space are contained in (Noran, 2007).

5 CONCLUSIONS

CN management involves complexity, politics, tacit knowledge and uncertainty. A DSS can help managers understand and isolate problems, examine scenarios, predict

outcomes and thus make *informed decisions*. The proposed support system framework has several additional distinctive features: it is supported by an original and tested theoretical concept; it is based on a life cycle paradigm, appropriate for the dynamic nature of organisations; it uses mainstream AF elements (while neutral in respect to any AF); and finally, it suggests areas that need to be represented in the change process via suitable models that enable management to communicate their 'justified true beliefs' (Nonaka and Takeuchi, 1995) to the rest of the organisation.

6 REFERENCES

1. Doumeings, G. La Methode GRAI. Doctoral Thesis, University of Bordeaux, Bordeaux, France, 1984
2. Finlay, P. N. *Introducing Decision Support Systems*. Manchester: Blackwell publishing, 1994
3. Friedman-Hill, E. *Jess in Action: Java Rule-based Systems*. Cherokee Station, New York, NY: Manning Publications Co., 2003
4. GRAISoft. IMAGIM software product, (PDF document). GRAISoft. Available: <http://www.graisoft.com>, 2002
5. Hättenschwiler, P. *Neues anwenderfreundliches Konzept der Entscheidungsunterstützung. Gutes Entscheiden in Wirtschaft, Politik und Gesellschaft*. Zurich: vdf Hochschulverlag AG, 1999
6. Holsapple, C. W., Whinston, A. B. *Decision Support Systems: A Knowledge-Based Approach*. Minneapolis / St. Paul: West Publishing, 1996
7. ISO/TC184. "Annex A: GERAM", ISO/IS 15704: Industrial automation systems - Requirements for enterprise-reference architectures and methodologies, 2000
8. Kalpic, B., Bernus, P. Business process modeling through the knowledge management perspective. *Journal of Knowledge Management*, 10(3), 2006
9. KBSI. AI0Win Software Product, (White Paper). Knowledge Based Systems, Inc. Available: <http://www.kbsi.com/Software/KBSI/AI0WIN.htm>, 2007
10. Keen, P. G. W., Scott Morton, M. S. *Decision Support Systems: An Organizational Perspective*. Reading, MA: Addison-Wesley, Inc, 1978
11. Marakas, G. M. *Decision support systems in the twenty-first century*. Upper Saddle River, N.J.: Prentice Hall., 1999
12. NIST. Integration Definition for Function Modelling (IDEF0) (183: Federal Information Processing Publication): Computer Systems Laboratory, National Institute of Standards and Technology. 1993
13. Nonaka, I., Takeuchi, H. *The Knowledge-Creating Company. How Japanese companies create the dynamics of innovation*. Oxford: Oxford University Press, 1995
14. Noran, O. *A Meta-methodology for Collaborative Networked Organisations*. Doctoral Thesis, School of CIT, Griffith University, 2004a
15. Noran, O. "A Meta-methodology for Collaborative Networked Organisations: 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
16. Noran, O. "A Meta-methodology Prototype for Collaborative Networked Organisations". In L. Camarinha-Matos (Ed.), *Collaborative Networks and Breeding Environments (Proceedings of the 6th IFIP Working Conference on Virtual Enterprises - PROVE 05)* (pp. 339-346). Toulouse / France: Springer Verlag, 2005
17. 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, 2006
18. Noran, O. "Discovering and modelling Enterprise Engineering Project Processes". In P. Saha (Ed.), *Enterprise Systems Architecture in Practice* (pp. 39-61) Hershey, USA: IDEA Group, 2007
19. Power, D. *Decision support systems: concepts and resources for managers* (Vol. 2007). Westport, Conn.: Quorum Books, 2002
20. Rumbaugh, J., et al. *The Unified Modelling Language Reference Manual*. Reading, MA: Addison-Wesley, 1999
21. Simon, H. *The New Science of Management Decision* (3rd ed.). Englewood Cliffs, NJ.: Prentice-Hall, 1977
22. Sprague, R. H. A Framework for the Development of Decision Support Systems. *Management Information Systems Quarterly*, 4(4), 1-26, 1980
23. Sprague, R. H., Carlson, E. D. *Building Effective Decision Support Systems*. Englewood Cliffs, NJ: Prentice-Hall, Inc, 1982
24. Turban, E. *Decision support and expert systems: management support systems*. Englewood Cliffs, N.J: Prentice Hall, 1995
25. 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