DISCOVERING AND MODELLING ENTERPRISE ENGINEERING PROJECT PROCESSES

Ovidiu Noran

School of Information and Communication Technology, Griffith University,

N44_1.23 Nathan (Brisbane) QLD 4111 Australia.

Phone: +61-7-3375-5382

Fax: +61-7-3425-3175

E-Mail: O.Noran@griffith.edu.au.
Abstract: Often, in an Enterprise Engineering (EE) project it is quite difficult to figure out what exactly needs to be done due to the rather generic (and often proprietary) character of the EE methods available. In addition, selecting appropriate elements from the multitude of available and emerging Architecture Frameworks (AFs) in order to model and manage the given EE undertaking is a non-trivial task. This chapter proposes a way to assist the inference of processes and to facilitate the selection and use of AF elements needed to accomplish EE projects. This is accomplished by assessing and organising AF elements into a Structured Repository (SR) using a generalised Architecture Framework (ISO15704:2000 Annex A) and by providing a ‘method to create methods’ (a meta-methodology) for specific EE tasks that also guides the selection of AF elements from the SR. A brief introduction outlining the above-mentioned EE problems is followed by the description of the meta-methodology principle and of the assessment reference used. Next, a case study presents a sample application of the meta-methodology for a real EE project. The chapter closes with conclusions on the presented approach and a description of further work to refine and enrich the meta-methodology.

Keywords: architecture frameworks, meta-models, GERAM, virtual enterprises

INTRODUCTION

Typically, the scope of Enterprise Engineering (EE) projects requires significant resources and involves large turn-around periods. Therefore, such projects should be approached using suitable and mature methods, modelling constructs and tools. EE practice in the Virtual
Enterprise (VE) domain (Globemen, 2000-2002) has shown that often, the initial problem in an EE project is the lack of a clear image of the activities that need to be performed to manage and execute that project. The currently available public and proprietary EE methods are quite generic, resembling reference models that need to be customised for specific projects; this typically requires knowledge of those methods (Noran, 2003a).

EE artefacts typically required by EE projects, such as modelling frameworks (MFs), reference models (RMs), modelling constructs (languages) and tools, etc can be provided in an integrated manner by Architecture Frameworks (AFs). Note that the term ‘Architecture Framework’ is understood in this chapter as an artefact defining the types of elements needed to support the creation of an object from the identification of the need to create that object through to its decommissioning. However, often the artefacts composing a single AF do not provide sufficient coverage for a specific EE project and thus, a combination of elements from several AFs is necessary. The complexity involved in most EE tasks makes the selection of AF elements a non-trivial task, usually requiring knowledge of the elements’ outcomes, prerequisites and dependencies on other AF elements.

This chapter proposes a basic method to guide the creation of a set of activity type descriptions expressing what needs to be done in a particular EE project, based on domain knowledge – i.e., based on project stakeholder / champion knowledge about the participating entities and their relations. The proposed method also assists in the selection of suitable AF elements for the specific needs of the particular EE project, based on their capabilities assessed in relation to a reference AF.

Note that the method and the reference AF used in this example do not prescribe the use of any specific AF or AF elements; they provide a way to assess AF element capabilities, to present a set of steps that specify types of activities needed to accomplish specific EE tasks and to recommend sets of AF elements suitable for those tasks; it is then up to the user to
select specific AF elements out of ranked lists, or override the recommendations. Therefore, this approach could be reused by EA practitioners to evaluate and select their preferred AF elements and to assess other methods for applicability to their specific EE project(s).

BACKGROUND: ISSUES IN ENTERPRISE ENGINEERING PRACTICE

The critical review of several mainstream AFs described in (Noran, 2003a, 2004a) has identified some of the problems associated with the use of AF elements in EE practice. The reviewed AFs were PERA (Purdue Enterprise Reference Architecture) (Williams, 1994), GRAI (Graphs with Results and Activities Inter-related) (Doumeingts, 1984), Computer Integrated manufacturing – Open System (CIMOSA) (CIMOSA Association, 1996), the Zachman Framework (Zachman, 1987), Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) (C4ISR Architectures Working Group, 1997) and ARchitecture for Information Systems (ARIS) (Scheer, 1992).

Note that subsequently, several other AFs such as TOGAF (The Open Group Architecture Framework) (The Open Group, 2006), FEAF (Federal Enterprise Architecture Framework) (US Federal CIO Council), Department of Defence Architecture Framework (DODAF) (Department of Defence Architecture Framework Working Group, 2003) and TEAF (Treasury Enterprise Architecture Framework) (Department of the Treasury CIO Council, 2000) have also been examined and found to be related to, and / or display similar main problems to the first six AFs. These problems are summarized below.

Methods: Generality

Several methodologies associated with, or relating to AFs have been reviewed in (Noran, 2003a), such as the Purdue Methodology, (published as a 'Handbook for Master Planning' (Williams, Rathwell, & Li, 2001)), the Structured Approach of the GRAI Integrated Methodology (Doumeingts, Vallespir, Zanettin, & Chen, 1992), the CIMOSA methodology

The review has found that all methods display various degrees of generality, reflecting the parent AFs’ aim to have broad applicability. However, for this reason the available methods are rather reference models of methods that need to be customised for each particular EE project. It has also been found that knowledge of these methods is required in order to decide on their selection and to subsequently customise them for the specific EE task.

**Models: Consistency**

A given EE task typically requires several enterprise models focusing on specific aspects. This creates the issue of consistency between the models created. For example, if the functionality of the enterprise is expressed in an activity / process model, the inputs / outputs, resources and controls of the activities represented in that model could also be described in information, resources and organisational models. In that case it is imperative that the various descriptions of the same artefact are kept in agreement. Typically this is no easy task and can be enforced by the users via policies (weak), triggers and procedures (medium), or through more formal means such as meta-models underlying modelling constructs (strong). Some of the reviewed AFs aim to ensure such consistency by integrating the views contained in their MFs, e.g. C4ISR’s Core Architecture Data Model, Zachman’s metamodel (Sowa & Zachman, 1992) and ARIS high level metamodel (Scheer, 1999).

**Languages: Integration, Complexity Control**

Some reviewed AFs such as PERA and Zachman aim not to prescribe a complete set of languages, instead arguing that the given engineering domain already has adequate means to describe any deliverable, and the architecture is only there to create a checklist of deliverables
and to make their relationships explicit. Such AFs only give examples of typical modelling languages. However, the choice of modelling languages is paramount for the user / stakeholder support and thus ultimately, for the success of an EE task. Thus, modelling languages can provide a degree of inter-model consistency by being part of a set / ‘family’ (e.g. IDEF (Menzel & Mayer, 1998) or UML (Rumbaugh, Jacobson, & Booch, 1999)), ideally integrated through a set of metamodels (e.g. UML, but not IDEF (Noran, 2003b)). Layering and complexity hiding abilities of a language allow parts of the same model to be developed and used simultaneously by several user groups while maintaining overall model consistency. It also allows presenting the model to various audiences, from technical personnel to upper management / CEO.

**Reference Models: Generality vs. Specialisation**

The review undertaken has found that RMs provided by some AFs, such as e.g. the Purdue CIM Reference Model (Williams, 1988), the GRAI Grid Reference Model (Doumeingts, Vallespir, & Chen, 1998), the CIMOSA Integration Infrastructure Services, the C4ISR Universal Reference Resources and the ARIS Y-CIM model (Scheer, 1994) come in various degrees of specialisation. A specialised RM has a narrow area of application, while a more generic RM needs to be customised before being used for a specific EE task. Knowledge of the RM is needed, either to be able to use a specialised RM, or to customise a generic one. RMs also vary greatly in their ability and efficiency to cover various life cycle phases.

**Tools: Awareness, Integration**

Effective enterprise modelling tools have to be ‘aware’ of the languages they support (i.e., support their syntax at a minimum). Noran (2004a) has reviewed several tools, including FirstStep (Levi & Klapsis, 1999), IMAGIM (GRAISoft, 2002) and System Architect (Popkin Software, 2001). It has been found that tools belonging to a ‘suite’ and / or based on a shared
repository and supporting languages belonging to a set can provide a degree of integration of
the models produced. Existing / legacy modelling tools and staff skills may constitute
deciding factors in modelling tools selection for an EE task.

**Proposed Solution**

The above-described issues can be addressed in two main steps. First, evaluate and organise
AF elements in respect to a *generalised reference* AF. Second, develop a method guiding the
discovery of the processes involved in accomplishing EE tasks and supporting the user in
selecting suitable AF elements for those tasks. These two steps have been addressed in the
*meta-methodology* (method to create methods) research described in (Noran, 2004a). Thus,
one of the main meta-methodology components is a *structured repository* containing elements
of several AFs assessed in respect to ISO15704:2000 Annex A (ISO/TC184, 2000) as a
common reference. The other major component is a set of steps assisting the inference of
activity types that need to be undertaken for specific EE tasks. The meta-methodology steps
use the repository content to build ranked lists of AF elements recommended to the user for
each step, taking into account the applicable aspects and life cycle phases.

**EVALUATION AND SELECTION OF ARCHITECTURE FRAMEWORK ELEMENTS**

**The Architecture Framework Assessment Reference**

The [ISO15704:2000 standard](#), 'Requirements for Enterprise Reference Architectures and
Methodologies', has been developed to supply a generic set of criteria to test the ability of
existing AFs to provide relevant assistance for EA tasks. Annex A of the standard, the
Generalised Enterprise Reference Architecture and Methodology (GERAM, see Figure 1) is a
generic AF compliant with ISO15704, obtained by generalising concepts present in several
mainstream AFs and in EA best-practice. Note that GERAM only specifies *requirements* for
tools, methods and models; it does not enforce any particular choices to satisfy these requirements.

**GERAM Components**

A main component of GERAM is its reference architecture (GERA), whose MF features a three-dimensional structure containing several views that may be used to structure the knowledge in various *interrelated* models (refer Figure 2). GERA does not enforce the presence of all views; also, other view types can be added if necessary. The condition however is that the views that *are* present must together cover the necessary aspects of the specific EE task. Thus, the user is presented with a list of possible views and must make informed decisions as to which views are indeed necessary for the task at hand. GERA is further described in the next section.

ISO15704:2000 specifies a requirement for compliant AFs to have associated methods
exemplified in GERAM by Enterprise Engineering Methodologies (EEMs)) guiding the user in utilising their elements. The level of specialisation of such methods can be assessed by using GERA’s MF Instantiation dimension (see next section).

Enterprise modelling languages (EMLs) provide the necessary constructs to describe the various artefacts present within GERAM. For example, EEMs represent the models of engineering processes by means of EMLs. Typically, a combination of EMLs has to be used to describe all necessary modelling aspects (information, function, organisation, behaviour, etc) of an EE project.

GERAM calls RMs partial enterprise models (PEMs) - reusable templates for human roles (organisational), processes (common functionality) or technology (e.g. IT resources).

Enterprise models (EMs) are the main vehicle for structuring the knowledge existent in the enterprise (e.g. by modelling the AS-IS state) but are also essential enablers of the change processes that may have prompted their creation (by modelling TO-BE states). A typical set of EMs as required by ISO15704:2000 and described by GERAM should include enterprise operations, organisation, IS and resources, and clearly show the human role within the control and production systems. Further detail, such as hardware / software may also be required depending on models’ intended use (see the GERA views in Figure 2).

Enterprise modules (EMOs) are implemented RMs, usable as trusted ‘plug-and-play’ components. For example, if a design has used certain RMs, the EMOs corresponding to those RMs can be directly used in the implementation of that design.

**GERA Structure**

GERA is a life cycle reference architecture, i.e. an architecture that can specify types of activities involved in the implementation of a project spanning over part, or the entire life of an entity (in contrast, a system architecture only models the structure of an entity (system) at a certain point in time). GERA contains an MF (represented in Figure 2) and other concepts
such as life history, entity recursiveness, etc.

Generally, an MF is a structure containing placeholders for artefacts needed in the modelling process. Depending on the structure of the framework, the type of these artefacts may be limited to models, or may extend to other construct types such as RMs, metamodels, glossaries, etc. For example, the GERA MF contains placeholders for artefacts of the types shown in the GERAM metamodel (see Figure 1). The GERA MF allows the practitioner to focus on certain aspects of complex EMs by using views defined by several criteria, as further described.

The model content criterion provides four views describing the functional, information, resource and organisation aspects of an enterprise. It is to be noted that the organisation view may be obtained by mapping part of the resource view (human side) onto a subset of the functional view (the human-implemented functions).

The purpose criterion divides the EE artefacts into production (or customer service) and control (or management). Typically (although not mandatory), this division is present when representing the relation between the life cycle phases of various enterprise entities in the business models used in meta-methodology application.

The implementation criterion provides a way to distinguish between the production / service and the management / control aspect of an enterprise, while at the same time allowing to represent the extent of the human-accomplished tasks (‘humanisability’) in both aspects. The human role is an essential success factor in EA, although it is often either overlooked or has only limited coverage in the AFs and RMs currently available.

Finally, the physical manifestation criterion provides a finer subdivision, setting apart hardware and software aspects of EA artefacts. It is mostly used to scope views derived from other criteria.

The second dimension of the GERA MF contains the concept of specialisation, which
allows to represent all of the above aspects at the generic, partial and particular levels (see Figure 2). For example, the partial level of GERA can be used to assess the coverage and degree of specialisation of RMs (PEMs) or methods (EEMs).

Figure 2. The modelling framework of GERA (ISO/TC184, 2000).

Life cycle represents the third dimension of the GERA MF, allowing to represent (and assess) the applicability of AF elements to various life cycle phases of the modelled entity. For example, this dimension enables the creation of business models in the context of the life cycles of the participating entities, which is an essential meta-methodology requirement.

Several other GERA artefacts reside outside its MF. For example, the concept of life history can be used to model process concurrency. In GERA, life ‘history’ implies a time dimension and represents the collection of life cycle phases that the entity has gone (or will go) through during its life. In contrast, life cycle abstracts from time and is a collection of life cycle phases that the entity could go through during its life.

In conclusion, GERAM provides a suitable reference for assessing the pool of AF
elements needed to construct a repository and support the meta-methodology application.

**Architecture Framework Element Assessment and Organisation into a Knowledge Base**

In addition to identifying problems in EE practice, the critical literature review described in (Noran, 2003a) has also mapped the reviewed AFs against GERAM, attempting to identify relevant components such as MFs, RMs, generic modelling concepts (such as metamodels, ontologies, etc), modelling methods, languages, tools, etc. The MF of GERA has then been used to *scope* the identified AF elements in order to check their intended coverage and determine their *potential* applicability domain(s) in terms of views, life cycle and specialisation. The assessed and mapped AF elements have then been stored in a knowledge base (KB) organising AF elements by name, type, family, integration, tools, and importantly, associating *prerequisites* and *outcomes* to each AF element. The KB has then been added rules for the selection of the AF elements and their ordering in ranked lists, resulting in the rule-based KB shown in **Figure 3**. The subsequent addition of an inference engine to the KB has resulted in the Structured Repository (SR) currently used by the meta-methodology concept (see **Figure 5**, left).
For example, the representation of a GRAI Grid decisional RM (explained elsewhere in this chapter) includes its name (=‘GRAI Grid’), type (=‘Reference Model’), whether it belongs to a set of reference models (YES, =‘GRAI’), if the set is integrated (NO – not formally) and whether the RM is supported by any available tools (YES, =‘IMAGIM’). The main outcome of the GRAI-Grid RM representation in the KB would be ‘decisional modelling’. However, GRAI Grid RMs can be also used for organisational modelling if the human resources have been previously mapped. Therefore, this GRAI-Grid RM could also have an ‘organisational modelling’ outcome, however associated with a ‘resource modelling’ prerequisite.

The mapping of the AF elements on GERAM and GERA has revealed that typically, elements of several AFs can cover a particular area of an EE project. However, their efficiency in modelling the area in question may vary greatly in respect to various aspects and life cycle phases. For example, GRAI-Grid would be quite efficient in modelling the decisional / organisational structure for the concept, requirements and architectural design of an entity, but
less efficient for the detailed design, implementation or operation of that entity. Therefore, the inference engine would create *ranked lists* of matching AF elements for the specific task; the user can then accept or override recommended choices and thus test various scenarios.

**GUIDANCE FOR PROCESS DISCOVERY**

In response to the methodological problem identified in the Background section, a basic procedure (‘meta-methodology’) has been developed to guide the development of methods (sets of activity / process types) for specific EE tasks. This method is based on the concept of 'reading' the life cycle phases of the target entity (here, one or more enterprises) in the context of its relations with other entities, using a life cycle-based business model, and, based on domain knowledge, inferring the necessary activities for each life cycle phase. The meta-methodology contains the following **main steps** (see Figure 4):

![Figure 4](image_url)

**Figure 4.** The meta-methodology concept (Noran, 2004a)

1) Identify a list of entities relevant to the EE project. If one or more projects are set up to build the target entity (entities), consider including them as target entities as well;

2) Create a business model showing the relations between the life cycles of the identified
entities, while re-assessing the need for the presence of each entity in the diagram, and the extent of life cycle set to be represented for each entity (e.g. full set, single phase, etc).

3) Reading the life cycle diagram of the target entity, phase by phase, infer a set of activities describing the creation of the entity and the roles played by other entities. For several target entities, determine the order in which they are influenced (from the relations in the diagram) and create the activity model in that order. Detail the activities to the necessary level using the aspects adopted and, if applicable, the views of a suitable MF.

The sub-steps relating to all the main steps are as follows:

a) Identify a suitable MF if applicable. Choose the aspects to model, depending on the meta-methodology step. Resolve any aspect dependencies;

b) Choose whether to represent only the present (AS-IS), future (TO-BE), or both states. If both, choose whether to represent them separately, or combined;

c) Choose modelling formalism(s) depending on the aspect(s) selected and on modelling best-practice criteria, such as: previously used in the modelling, specialisation in those aspects, potential multiple use, part of a family, family integration. Choose modelling tool depending on formalism and other factors (such as belonging to a suite, availability, etc).

The application of these steps is illustrated in a case study presented further in this Chapter.

In building the activity model (step 3), a ‘sufficient’ level of detail is reached when the processes can be directly executed by the envisaged resources.

Note that the main aim of this chapter is to incite reader interest in the potential meta-methodology reuse and application to other EE tasks, rather than to provide a full description of the meta-methodology development and use. The interested reader is invited to get further meta-methodology details from (Noran, 2004a).
Note on the Meta-methodology Capabilities

The meta-methodology provides guidance and assistance, but it does not infer the activities for the user. This is because as shown in Figure 4, the domain knowledge required to deduce such activities is quite specific and owned by the specific project’s stakeholders. The meta-methodology user (typically the enterprise architect) has to elicit this knowledge from the stakeholders or otherwise acquire it by immersing in the participating organisation(s).

For example, step 3) tells the user how to elicit the activities from the business model. The sub-steps applied to step three assist and prompt the user to select appropriate aspects and modelling formalisms, in order to enforce a degree of coherence to the activity inference process. However, specific conclusions and insights gained by the user in steps 1) and 2) should be also used to select and (if warranted) even override step three recommendations.

In conclusion, the meta-methodology concept implies constructive assistance, rather than blindly following a set of fixed steps that automatically lead to a unique, ‘best’ solution.

INTEGRATION OF THE META-METHODOLOGY STEPS AND THE SR

As can be seen in Figure 4, there is a close relationship between the SR and the meta-methodology steps. This connection is further detailed in Figure 5: the SR (containing the KB) is integrated with the meta-methodology steps in a meta-methodology environment. Thus, the search / inference engine (or an artefact with similar functionality, depending on the implementation) within the SR selects AF elements at the request of the meta-methodology steps and sub-steps and arranges them in ranked lists using the KB matching and ordering rules. The engine also attempts to resolve dependencies and prerequisites – i.e., if the chosen AF elements require other elements in the KB, they are also added as ranked lists to the proposed solution. The user is notified of any unsatisfied prerequisites.
Another aspect of the above-mentioned integration is the implementation of meta-methodology step logic in the KB rules. For example, the main aspect recommended for modelling in sub-step a) can be supplied by a rule in the KB of the SR, e.g. on the basis of the step number; subsequently, the engine would also resolve any aspect dependencies by recommending additional aspects if applicable.

Dynamic facts (specific to the particular setting) that cannot be stored in the KB, such as staff proficiency in particular modelling languages and modelling tools, available / legacy systems and other relevant constraints within the participating organisations are gathered from the user at runtime and are used in ranking the AF element lists.

**CASE STUDY: THE FORMATION OF A VIRTUAL ORGANISATION**

This section provides an example of using the previously described meta-methodology environment to infer the types of activities needed to accomplish a concrete EE task, namely the creation of a virtual organisation in the higher education sector. AF elements needed to model and manage the virtual organisation creation process are also suggested. Note that the
presented example is applicable to the creation of other VO types in other areas.

Collaborative Networks and Virtual Organisations in the Industry

In the current global market, enterprises worldwide must often come together for the purpose of tendering, executing and servicing large scale projects requiring competencies and resources beyond their individual capabilities. The resulting organisations, although composed of various participants, appear as an indivisible entity to the outside environment (service providers, clients, etc) - hence their name of virtual organisations (VOs). VOs exist as separate entities for legal and commercial purposes although in fact, most of their resources belong to the participant organisations.

![Figure 6. Company Network and Virtual Organisation concepts](image)

(Globemen, 2000-2002; Tølle & Vesterager, 2002)

Usually, the formation of VOs takes time, due to potentially lengthy processes such as trust building, obtaining commitment from management, agreement on a stable common ICT infrastructure, establishment of commonly understood RMs, etc. Typically however, the time available to react to a business opportunity (such as a large project) is limited. Therefore, enterprises aiming to successfully tender for such projects often become Collaborative
Networked Organisations (CNO) forming a collaborative network (Camarinha-Matos, 2004) or a company network (CN) (Globemen, 2000-2002) (see Figure 6). Such a CN provides a ‘breeding environment’ where the participants can accomplish the previously described processes in advance and thus improve their preparedness to quickly form VOs as required.

**A Virtual Organisation for the Tertiary Education Sector**

Faculty FAC within university U contains several schools (A to D), with schools A and B having the same profile. School A is based in two campuses, situated at locations L1 and L2, while school B is based on single campus, situated at location L3 (AS-IS state in Figure 7). Although of the same profile, and belonging to the same FAC and U, schools A and B are confronted with a lack of consistency in their products and resources, such as the programs and courses offered, allocated budget, academic profile and availability of teaching staff, etc. This situation causes negative effects, such as additional costs in student administration and in course / program design / maintenance and staff perceptions of unequal academic and professional standing between campuses, all of which are detrimental to the entire faculty. The problems previously described could be resolved by schools A and B forming a VO (called merged school (MS) in the TO-BE state in Figure 7) with cross-campus management and policies ensuring inherent consistency in the strategy regarding the products delivered and the resources allocated to future VO campuses at L1, L2 and L3. The individual campuses are set to retain most of their internal decisional and organisational structure except for the highest layer, which will be replaced by the VO governance structure.
In view of the CN / VO model previously explained, in this case study the CN function is performed by FAC, which contains several schools forming VOs as necessary. The ‘partners’ (i.e. schools) A and B within the CN come together at the initiative of U to form an on-going VO project. Importantly, the partners cease to operate independently during the life of MS.

**Meta-methodology Application**

*Step One: Identification of the Entities Relevant to the EE Project*

The list of entities has been built according to the domain knowledge elicited from, and checked with the stakeholders and the project champions (Pro-Vice-Chancellor (PVC), Dean).

**Sub-step a):** Choose life cycle representation, however no details: only stating to be full set of phases vs. one phase only. Others not necessary. **Motivation:** the representation must be kept simple. In this early stage, details are not relevant and can be detrimental to creative output;
**Sub-step 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;

**Sub-step c):** Choose text representation as modelling formalism. Choose a plain text editor or whiteboard as ‘tool’. **Motivation:** formalism and tool must preserve simplicity (see a) above).

The list of entities constructed in this step is shown in the legend of **Figure 10**.

**Step Two: A Business Model expressing Entity Roles in a Life Cycle Context**

**Sub-step a):** adopt GERA as a MF. Using GERA as a checklist, it has been established that management vs. product / service aspects and also decisional and organisational models of the entities participating in the VO formation may be necessary. These representations must be shown in the context of life cycle. **Motivation:** the scope of this EE project includes management, product / service, decisional structure and organisation. Life cycle representation is necessary for the second and third main meta-methodology steps.

**Sub-step b):** represent both AS-IS and TO-BE states. For management / service and life cycle, represent AS-IS and TO-BE in a combined view. Represent decisional and organisational views in separate models. **Motivation:** Since the present situation was not fully understood by the stakeholders, AS-IS needed to be modelled. Both separate and combined AS-IS / TO-BE representations have been considered. There was no tangible advantage in showing separate AS-IS / TO-BE states in the business model showing management / service in the context of life cycle. However the interest shown by the stakeholders in the decisional structure and the fact that the organisational structure was the only representation able to discern between several TO-BE states have determined the choice of separate AS-IS and TO-BE representations.

**Sub-step c):** choose modelling formalisms ranking highest in efficiency for the aspects selected in sub-step a). For life cycle and management / service, choose a ‘chocolate bar’ formalism derived from GERA as shown in **Figure 8**.
Figure 8. Modelling formalism used for the life cycle and management vs. service/product views of the business model.

Choose the GRAI–Grid formalism (Figure 9) for decisional and organisational aspects.

GRAI-Grid ranks high in respect to other candidate languages due to its specialisation for the decisional aspect, potential multiple use (e.g. for organisational aspect) and no prerequisites.

Legend:  
= decisional centre (DC);  
= decisional framework (DF);  
= information flow (IF)
Next, construct the business model (Figure 10) illustrating entity roles in accomplishing the EE project in the context of their life cycles and management vs. production / service aspects.

The business model in Figure 10 allows a bird’s eye view on the entities involved in the target entities’ life cycle phases and has been constructed using the domain knowledge of the stakeholders as elicited by the user. Several entities influence various life cycle phases of MS directly, or through other entities. The main influence is exercised by ITM, which is in fact a project set up to build the VO and ceases to exist after MS starts operating. Note that this is a common occurrence in EE projects, reflected in the body of step 1) of the meta-methodology.
The AS-IS decisional model (Figure 11) has shown in more detail some problems that have triggered the EE project, such as a high degree of turbulence and a lack of clear and effective strategy within the organisation. Thus, the Head of School (HOS) in the role of Planner had to put out 'fires' (product / resource discrepancies requiring immediate reconciliation) on a short-term basis, rather than having strategies in place to avoid the cause of such problems. This can be seen in Figure 11 in the form of decisional frameworks (DFs) flowing from Planning towards Products and Resources at all horizons. The AS-IS decisional model has also shown a lack of sufficient financial and decisional independence of the schools A and B and a shortage of information crucial to long-term strategy making (shown in Figure 11 by several DFs coming from outside the School and by limited external information flow).

The TO-BE decisional model (see Figure 12) has attempted to address these problems by confining the authoritarian role of the HOS to the strategic level, by increasing the financial and decisional independence of the target VO (MS), and by providing the necessary external information to MS for the purpose of strategy making and self-governance. This is reflected in a lesser number of horizontal DFs originating from Planning, less DFs coming from U and increased number of information flows in Figure 12. Also, some existing external DFs have been relocated so as to minimise the negative impact on MS.

Figure 11 and Figure 12 show the AS-IS and TO-BE decisional and organisational models.
Interestingly, the TO-BE decisional model has been found to match all potential scenarios, with differences only obvious in the organisational structure derived from the decisional model, i.e. the allocation of the various human resource groups to the decision centres (e.g. see areas defined as ‘committees’ in Figure 12). Therefore, modelling the organisational structure was instrumental in distinguishing between the available TO-BE options.

**Step Three: the Set of Activities describing the EE Project Accomplishment**

This step represents the main purpose of the meta-methodology, resulting in a set of activities describing how to accomplish the EE project (here, the VO creation and partly operation). The step is accomplished by 'reading' the life cycle diagram of the entity to be designed (MS) and its relations with the other entities shown in Figure 10. The set of activities obtained is then refined and decomposed into sub-activities, using views of the selected MF (GERA).

**Sub-step a):** aspects: functional and life cycle, but use other categories to detail activities.

**Motivation:** the main deliverable is an activity model, hence functional. However, the activities must be detailed using aspect from the SR and views from the selected AF, in this case management / service with human vs. machine and software vs. hardware aspects.

**Sub-step b):** choose to represent both AS-IS and TO-BE states in a unified representation.

**Motivation:** the activity model is expressing the transition from AS-IS to TO-BE, thus both states should be represented; here, separate views did not justify the consistency overhead.

**Sub-step c):** choose IDEF0 (NIST, 1993) functional modelling formalism (see Figure 13 for a basic explanation of its elements). Select AI0Win tool by Knowledge-Based Systems, Inc.

**Motivations:** UML and IDEF0 are both suitable; UML ranks higher than IDEF0, due to integrated metamodel present and wider tool support. However, the available (AI0Win) modelling tool and necessary skills were an overriding factor. AI0Win is also semi-integrated with information / resources modelling (‘SmartER’) and behaviour modelling (‘ProSIM’) tools, similar to a ‘suite’, thus ensuring some degree of model consistency. Therefore, overall
IDEF0 and AI0Win were considered a suitable choice in this particular case.

![Diagram of IDEF0 modelling formalism](image)

**Figure 13** IDEF0 modelling formalism (1st, or ‘context’ level shown)

**Creating the Activity Types in Step Three**

A set of activities must be inferred for each life cycle phase represented in the business model. Thus, the functional model can be initiated by creating one main activity for each life cycle phase identified in the business model (such as shown in **Figure 14**). The modelling formalism chosen to represent the model will assist in developing the model. For example, IDEF0 requires that inputs, controls, outputs and mechanisms (ICOMs) are defined for these activities (see **Figure 13**, where inputs can be omitted if justified). Therefore, the user has to give consideration to what (if anything) is used in the activity, what controls it, what is produced by the activity and who / what executes it. This information can be construed from the roles shown in the business model and further domain knowledge.
Figure 14 IDEF0 Activity model for the VO creation and operation (2nd level shown)

Each activity must be recursively decomposed down to a level where it can be directly...
executed; this can be assisted by a ‘checklist’ of views used in previous steps and / or the chosen MF (if any) and / or available in the MFs contained in the SR. For example, if GERA’s MF is chosen, the checklist contains Function, Information, Resources, Organisation, Management vs. Production / Service, Human Roles and Hardware / Software. For example, in Figure 16 Management vs. Production / Service and Human Role views have been used to decompose the Detailed Design activity.

Note that 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 aspect may only be visible / relevant in the Preliminary / Detailed Design phases, such as illustrated in Figure 2.

Example: Creating the Second Level of the Functional Model

Figure 10 shows that the MS entity is influenced by the university U, Across Campus Consistency (ACC) project and by the project that builds it, namely ITM.

The first phase in the life cycle of MS is the identification of the need for this artefact. However, in this case, domain knowledge obtained from stakeholders has established that this phase has already been accomplished by U, i.e. the PVC has determined the need for an MS and mandated its creation; thus, Identification does not need to be represented in the activity model in this case.

The Concept phase is accomplished under the control of the mandate from PVC, ACC reviews that show what needs to be addressed and U’s policies. The work to define the concept of MS is carried out by a working party composed of U staff, the majority of which are in fact also staffing the ITM project. No inputs are present. The outputs reflect the creation and refinement of the MS concept, such as goals, objectives and responsibilities of the MS.

The main changes brought by this EE project are in the management area. Thus, the Requirements definition process needs to investigate suitable management structures and in addition, make needed corrections to the MS concept. For this purpose this activity needed to
contain the creation and use of decisional / organisational frameworks (such as shown in Figure 11 and Figure 12). These sub-activities are shown in Figure 15.

**Figure 15** Decomposition of the Requirements activity (3rd level IDEF0 model)

In this case study, only organisational models were able to discern between various TO-BE states. Therefore, the Preliminary Design phase must develop organisational structures reflecting various scenarios and recommend / seek stakeholders’ approval for the optimal one.

The Detailed Design Activity was further decomposed according to the GERA Implementation criterion, thus showing Management vs. Production / Service and Human Roles, as shown in Figure 16. Further decompositions then discerned between hardware and software-oriented activities. (Noran, 2004a) contains the relevant diagrams.
The VO Implementation process must set up the transition process, and then implement the detailed specifications produced by the Detailed Design sub-activities (thus some of its decomposition was similar to Detailed Design activity).

In this case, the Operation process of the School can be decomposed in operation and monitoring (producing internal information for the management and requests for change that cannot be handled at operational level). The EE project did not include detailed operation of the VO, thus no decompositions below level three have been produced for this activity.

A full description of the activity model creation and structure (which is beyond the purpose of this paper) is contained in (Noran, 2004b). Note that the activity model created represents only one of the many possible suitable solutions to this EE problem.

**CONCLUSIONS AND FURTHER WORK**

A practical problem in EE projects is not knowing what *exactly* needs to be done since
potentially usable methods, reference models, modelling frameworks, etc are rather general in nature. Furthermore, several such AF elements are often suitable for any given area of the project in question, although their modelling efficiency varies greatly depending on aspect, life cycle phase and specialisation required. Selecting and customising suitable AF elements requires knowledge of such artefacts, which is not normally available. However, domain knowledge describing the problems that triggered the project, specific features and relations between project participants, etc is typically available within stakeholders and/or project champion(s), even if often in a tacit form needing to be elicited.

In conclusion, help is needed to discover and model what needs to be done for as specific EE project and to select AF elements that suit the particular project but also the host organisation’s specific proficiencies, legacies and culture.

This chapter has described a method aiming to assist the discovery of activity types involved in accomplishing specific EE tasks and to provide guidance for the selection of AF elements needed to model and manage those tasks. The proposed meta-methodology achieves these goals through a meta-methodology environment comprising a set of steps and an SR containing AF elements assessed and organised in respect to a suitable reference (GERAM).

Research currently in progress aims to implement the KB and SR in a web-based, open source expert system for possible use as a decision support system for upper / middle management. Future research efforts in this area will be focused towards formalising the meta-methodology environment and enriching the content of the KB by mapping additional AF elements against the chosen reference. Further field testing and refinement of the KB rules and meta-methodology steps is also needed in order to improve its accuracy and usefulness.

REFERENCES


Levi, M. H., & Klapsis, M. P. (1999). FirstSTEP process modeler — a CIMOSA-


