A Framework to Define a Generic Enterprise Reference Architecture and Methodology

Peter Bernus
School of Computing and Information
Technology , Griffith University
Nathan Qld 4111 Australia
bernus@cit.gu.edu.au

Laszlo Nemes
CSIRO
Division of Manufacturing Technology
Preston VIC., 3072 Australia
lnm@mlb.dmt.csiro.au

DRAFT

Abstract

This paper presents a framework suitable for the definition of a Generic Enterprise Reference Architecture and Methodology to readers interested in architectures- and methodology development.

The Generic Enterprise Reference Architecture and Methodology is about those methods, models and tools which are needed to build the integrated enterprise. The architecture is generic because it applies to most, potentially all types of enterprise.

The coverage of the framework spans Products, Enterprises, Enterprise Integration and Strategic Enterprise Management, with the emphasis being on the middle two. The proposal for the architecture follows the architecture itself improving the quality of the presentation and of the outcome.

Definitions of Generic Enterprise Reference Architecture, Enterprise Engineering/ Integration Methodology, Enterprise Modelling Languages, Enterprise Models, and Enterprise Modules are given. It is proposed how the above could be developed on the basis of previously analysed architectures (and other results too), such as the Purdue Enterprise Reference Architecture[3], the GRAI Integrated Methodology[8], CIM-OSA[7], and TOVIE[5].

Keywords: Enterprise Integration, Reference Architecture, Enterprise Engineering Methodology, Enterprise Modelling

Introduction: Identification of what is a Generic Enterprise Reference Architecture and Methodology

This paper presents a framework suitable for the definition of a Generic Enterprise Reference Architecture and Methodology.

The Generic Enterprise Reference Architecture and Methodology is about those methods, models and tools which are needed to build the integrated enterprise. The architecture is generic because it applies to most, potentially all types of enterprise.

The scope of the architecture is therefore the union of domains which need the attention of enterprise engineering and development. Thus the scope is defined through a pragmatic need, the need to design and redesign as well as continuously improve the functioning of enterprises.

The Concept of a Generic Enterprise Reference Architecture and Methodology

The functional components of a Generic Enterprise Reference Architecture and Methodology are the following:

- Generic Enterprise Reference Architecture (GERA);

  This is the definition of enterprise related concepts, with the primary focus on the life-cycle of the enterprise. Since the life-cycle can be considered as a design process the architecture will also have to identify the results and the intermediary components of this design process.

- Generic Enterprise Engineering Methodology (GEEM).

  This is the description, on a generic level, of the processes of enterprise integration. In other words the methodology is a detailed process-model, with instructions for each step[1] of the integration project.

- Generic Enterprise Modelling Tools and Languages (GEMT&L);

  The engineering of the integrated enterprise is a highly sophisticated, multidisciplinary management, design and implementation exercise during which various forms of descriptions and models of the target enterprise need to be created. To express these models potentially more then one modelling language is needed.

[1] "Step" is to be understood as a functional component not as an element of a temporal sequence of actions.

Further, based on the above three components, define:

- Generic Enterprise Models (GEMs)

  Generic enterprise models capture concepts which are common to all enterprises. Therefore the enterprise engineering process can use them as tested components for building any specific enterprise model. It is possible to distinguish two levels of models

  – Ontological theories

    These theories describe the most generic aspects of enterprise-related concepts. (They can also be considered to be "meta-models" because the facts and rules in them are *about* facts and rules of enterprise models). Ontological theories play similar role that "data models" play in database design. Ontological theories capture the most basic properties of enterprise-related concepts (function, structure, dynamics, cost, etc).

  – Reusable Enterprise Models which are of the following types:

    ∗ Models which capture some common part of a class of enterprises. This type of reusable enterprise model can be used as a building block of a complete set of models.

    ∗ Paradigmatic (prototypical) models which describe a typical enterprise of a class. Prototype models can be subsequently modified to fit a particular case.

    ∗ Generic (abstract) model of a part of a class of enterprise which captures the commonalities but leaves out specific details. This type of model is of the "fill-in-the-blank" type.

- Generic Enterprise Modules (GMs)[2]

  Modules are *products*, which are standard implementations of components that are likely to be used in enterprise integration – either by the enterprise integration project or by the enterprise itself. Generic modules can be configured to form more complex modules for the use of an individual enterprise. Two typical generic modules are:

  – Enterprise Engineering Tool (EET)

  – Enterprise Integration Platform or Integration Infrastructure (IIS)

[2] Modules are generic in the sense that they are generally applicable.

Requirements of the Generic Enterprise Reference Architecture and Methodology

These are requirements that any enterprise reference architecture and methodology should satisfy. References to strong features of example architectures are also given. The overview is primarily based on the analysis which was carried out by the IFIP/IFAC Task Force on Enterprise Reference Architectures[1].[3]

It is proposed that a published definition of the Generic Reference Architecture and Methodology should contain its own detailed requirements definition and design decisions. The following list can form the basis of this requirements definition.

- The Best Treatment of the Enterprise Scope From the System Theoretic Point of View

  It is necessary that all activities which are involved either directly or indirectly in designing and operating or improving the enterprise should be covered by the architecture. Given that the goal is to provide methodologies (not an architecture alone) the architecture should be the backbone of the methodology. Thus the architecture should be based on the modelling of the enterprise engineering process, or life-cycle.

  A prominent example of this is the PERA [3] architecture which most fully captures the enterprise life-cycle.

- The Provision of a Consistent Modelling Environment Leading to Executable Code

  The modelling views offered should cover a minimal set (such as in CIM-OSA or TOVIE), but this set should be expandable with new related views. Modelling views should be based on a common theory, or meta-model, through which views can be related.

  The ideal modelling environment should be modular so that alternative methodologies could be based on it. I.e., there should not be any prejudice built into the modelling languages as to what the methodology will be.

  The modelling environment should be extensible rather then a closed set of models, and permissive – leaving space for alternative modelling methods [4].

  CIM-OSA (after adopting GRAI's modelling of the decision aspect) has created a consistent set of modelling tools (with a common meta-model) which can support an enterprise engineering methodology. Since the architecture also covers the operation of the enterprise the requirement is strongly connected with the functionality of information integration infrastructures.

  Other suites of modelling methods also exist (e.g. the IDEFX set of languages [9]). The TOVIE [5] system of generic enterprise models also define a consistent set of modelling

[3] Formerly called "IFIP/IFAC Task Force on Integrating Manufacturing Entities and Enterprises."

classes (as an enterprise ontology). The Generic Enterprise Modelling Language and Tool set should amalgamate the advantages of these[4].

- The Existence of a Detailed Methodology Which Enterprises Can Follow

  In addition to the methodology being technically correct it must be understandable and usable by the communities targeted. Thus the methodology should be executable by real (as opposed to hypothetical) teams, guaranteeing a high quality result within acceptable cost, time, and resource constraints.

  The methodology should identify the application circumstances which must hold, with particular attention to the size and maturity of the enterprise wishing to apply it. ISO 9000 series of relevant quality standards should be treated as a guideline for the enterprise engineering process resulting from the methodology.

  The GIM methodology [8] has been demonstrated on numerous industrial projects as effective while the Purdue Implementation Procedures Manual [10] has the widest coverage.

  The ideal Enterprise Engineering Methodology should also be expandable, since new engineering methods will always come into existence and the framework needs to anticipate and accommodate those developments.

  It should be possible for the methodology to be presented both in a generic fashion and in specialised forms. These specialised forms could better serve particular areas of industry. The Generic Enterprise Engineering Methodology should allow an entire family of methodologies to be defined on its basis.

- The Adoption of Good Engineering Practice for Building Reusable, Tested, and Standard Models

  It is important that the apparent complexity of the enterprise engineering process should be kept low. Intricate details of models should be encapsulated in reusable building blocks.

  Enterprise integration and enterprise engineering is a complex design process carried out by a group of people. As such, the methodology should ideally be based on a sound design theory treating design as a collaborative activity of multiple agents. Such theoretical basis is meant to ensure that the methodology is not only practical but also is formulated in a durable way.

- The Provision of a Unifying Perspective for Products, Processes, Management, Enterprise Development, and Strategic Management

  The development of the enterprise can also be considered as only one of the activities in the enterprise. Thus the architecture should tie and relate enterprise integration and enterprise engineering to the rest of activities in the enterprise. Especially important is the wish to support evolutionary paths to enterprise integration, although revolutionary processes can (and should) not be excluded.

  The Generic Enterprise Reference Architecture and Methodology must clearly define how other efforts to integration relate to it. The architecture should make it simpler to grasp integration then it was before.

[4] See design considerations as to how this can be achieved.

> It is permissible, indeed desirable, to have multiple presentations of the architecture to facilitate understanding and acceptance by a variety of user and developer groups and to ease identification with it.
>
> The SATT methodology [4] developed a recursive view of enterprise architecture where the end product of one architecture is the process creating the second etc. Using this principle the complexity of the methodology can be tackled by applying the reference architecture and methodology to its own development! The present article attempts to start this by a presentation that follows the anticipated architectural design.

The above list of requirements can be completed and organised as functional, information, organisational, resource requirements, etc. [5].

Design Decisions on the Generic Enterprise Reference Architecture and Methodology

Proposed Design Decisions To Satisfy the Requirements

From the requirements follow that design decisions on the Generic Enterprise Reference Architecture and Methodology must address all components and their combination: architecture, methodology, modelling language and generic models/modules. It also follows that it should be permitted for a multitude of methodologies to exist – some competing – while the reference framework allows this competition to occur on common grounds.

Through the combination of the dimensions along which the PERA, GIM, and CIM-OSA architectures divide concerns of the enterprise life-cycle the authors developed and contributed to [2] a *matrix representation* of the enterprise life-cycle model; this model was further improved[6], by expanding it with the identification- and concept- as well as build- and operate- phases of the enterprise life-cycle.

The matrix there developed described the life-cycle of the enterprise and mapped on it the three investigated architectures (PERA, GIM, and CIM-OSA) to compare their coverage. This matrix can be thought of as a generalisation of the CIM-OSA cube or as a CIM-OSA-cube-like presentation of the PERA two-tiered diagram or of GIM.

However, as shown here, the matrix is capable of describing the life-cycle of any other *entity* covering generic, partial and particular models/descriptions of that entity. These models/descriptions are the ones that should (or may) be produced during the entity's life-cycle.

[5] Using a bootstrapping method

[6] The improvement was suggested by T.J.Williams the chairman of the Task Force.

Before presenting how the idea of the matrix can be utilised for the design of the Generic Enterprise Reference Architecture and Methodology a brief presentation of some technical details is necessary.

The Life-Cycle Matrix

The matrix (see Fig.1), first of all, is a specification of the Generic Architecture of an Entity, such as the enterprise. Each area in the matrix has

- Subject Matter Specification –

  – collating these specifications one can produce the terms of reference for the entity, the life-cycle of which is under consideration.

- Modelling Language Specification –

  – such that the language is suitable for expressing descriptions/models that belong to the subject matter.

The various modelling views (as derived from CIM-OSA and GIM) and the corresponding classes of models may have to be expanded with additional ones, such as cost, economy, or time. However, these views should be (as they are appropriately called) only projections of an underlying model. In other words, adding for example a view of temporal behaviour can only be done by incorporating time in the underlying model and then expressing the temporal aspect through a view mechanism.

- Candidate Architecture

  An "Candidate Architecture" is an alternative presentation of a generic architecture. Such candidates are the PERA two tiered diagram, the CIM-OSA cube or GIM.

Candidate Architectures can be mapped onto the matrix by graphically presenting them as a collection of ellipsoids drawn on the matrix diagram. Each ellipsoid corresponds to a description category of the candidate architecture and an ellipsoid normally covers more then one Subject Matter of the matrix. If not all Subject Matters are covered by ellipsoids representing a Candidate Architecture then that candidate architecture is not yet complete.

- Methodology

Given a matrix for the life-cycle of an entity it is then possible to define a Methodology. A methodology is a set of procedures and methods which enable the user to produce the descriptions of a set of areas (Subject Matters). The graphical representation on the matrix of a

Figure 1: Matrix Representation of the Life-cycle of any Entity

methodology's coverage is by shading the ellipsoids that correspond to the components of the Candidate Architecture.

If not all ellipsoids are shaded for a methodology that means that the methodology is not yet complete relative to its own architecture definition.

- Generic Models

For some areas of the matrix there can be Generic Models available. Notably, for the most generic (left hand) column, these "models" are *ontological theories* that describe the meaning of common concepts of the subject matter.

If such model is available, then the graphical representation on the matrix of this fact is a filled triangle in the corresponding area.

Proposed Four Matrices of the Generic Enterprise Reference Architecture

Adopting the recursive view of integration developed in SATT [4] altogether *four* life-cycle matrices can be identified. We define four entity types and corresponding four life-cycles:

1 The Strategic Enterprise Management Process (Life-Cycle)

2 The Enterprise Engineering/ Integration Process Life-Cycle

3 The Enterprise Life-Cycle

4 The Product Life-Cycle

One such matrix (#3) was presented in detail in [2], and described the life-cycle of the enterprise.

However, it is also possible to define the life-cycle of the Enterprise Engineering / Integration Processes . This definition would then consist of the concept, identification, requirements definition, design, implementation (detailed design) etc. of enterprise engineering projects. The present document is a first attempt to do this.

The Methodology Life-Cycle Matrix

The customer service of the Enterprise Engineering Process is that it implements the life-cycle of the enterprise. Matrix #2 can be divided into two broad areas, customer service and control, i.e.:

Figure 2: Individual Projects Relying on Generic Methodology

- The definition of the ”customer service” side of enterprise engineering projects

  These areas of the matrix together cover the implementation, the building, and operation of an enterprise integration *project*. In fact the descriptions/models of this area should collectively make up a complete Implementation Procedures Manual!

  A methodology handbook (such as the Purdue Implementation Procedures Manual [10]) is, in this matrix, a generic functional model of the Enterprise Integration Processes. (See “PERA IPM” on Fig.2.) The representation of this fact on matrix #2 is a triangle in the ”generic requirements” area. Note, however, that a detailed investigation of the Purdue Implementation Procedures Manual and the corresponding GIM documents should be undertaken to establish which other areas of the matrix are also covered and which one could be further detailed to the benefit of the generic methodology.

- Definitions of generic project management concepts, requirements, and designs – along with generic models (and modules such as project management software).

  These generic models would then be specialised for the purposes of enterprise engineering projects and even further specialised for any particular enterprise integration project.

  The generic models of this level would formulate a *project management ontology* and specialised models would be available for different types of projects. The best practice of project management in general and enterprise-engineering-project management in particular would be documented as models of this matrix, including the definition of these on all levels from the concept to the implementation and operation. (See Fig.3).

Note that the “operation” phase of this matrix is the actual enterprise engineering activity, while the “build” phase is preoccupied by creating the enterprise engineering project team (including education). Phases before that are carried out by management (as part of the Strategic Enterprise Management operations) – defining the mandate of the enterprise integration project, organising its project management, planning etc.

Figure 3: Management Requirements of Integration Projects in the Methodology Matrix

It immediately follows that a Generic Enterprise Engineering / Integration Methodology is a pattern (in the left hand side) of matrix #2; and there can be specialisations of the methodology. (See Fig.4)

A few useful specialisations would be:

- Methodology for integrating enterprises of a given sort, (e.g. a "Methodology for the Integration of Government Departments" etc.)
- "Methodology for Innovative Re-engineering of an Enterprise" versus "Methodology for Evolutionary Enterprise Engineering" projects.

Specialised methodologies have the advantage that they better suit industry (filling in the gap between abstract methodological instructions and practical steps). Furthermore specialised methodologies are more efficient because there is less need to spend time on adopting them to the special circumstances of the individual project.

The practical advantage of defining, in detail, the subject matters of matrix #2 would then be that the connection between enterprise integration / enterprise engineering and project management would be apparent in such an extended reference architecture providing common ground for two separate – although in the practice mutually reliant – disciplines (enterprise engineering and project management).

Figure 4: Specialising the Generic Enterprise Engineering Methodology

The Strategic Enterprise Management Life-Cycle Matrix

As the process of matrix #2 (enterprise engineering) results in all deliverables of matrix #3 (the enterprise) it is relevant to ask: what is the process that brings about the enterprise engineering process and is it of interest to the enterprise integration community?

It is proposed that the subject matter of matrix #1 is the life-cycle of strategic enterprise management (SEM). As SEM operates it brings about (among other things) enterprise engineering projects.

Although not all subject matters in that matrix are directly relevant to enterprise engineering, some parts are. Namely, the methods for the identification of the necessity and the starting of enterprise engineering efforts belong to the operation of Strategic Enterprise Management and these methods should also be part of the generic enterprise integration methodology. Similarly, the selection of suitable methodology (and the methods to create a specialised methodology) belong to this matrix.

For the completeness sake matrix #1 should be defined to the extent that the above functions can be identified. The requisite methods need to be specified and (if possible) paradigmatic models should be incorporated.

The Product Life-Cycle Matrix

Similarly to the way matrix #1 was discovered it is possible to ask: which entity's life-cycle is implemented by the enterprise? The obvious answer is that it is the products (services) produced by the enterprise.

In fact many attempts to implement the integrated information flow in the enterprise have been based on *Integrated Product Models*. The question arises, then, how does enterprise engineering – as an integrating factor – relate to product-model-based integration?

With the introduction of the Product Life-Cycle Matrix it is now clear that the enterprise model should be such (including the requirements, design, detailed design and building and operating the enterprise) that it implements the full product life-cycle.

By looking at the product life-cycle matrix the enterprise engineer can derive a full set of requirements for the enterprise designed. In fact the knowledge of the product life-cycle is a prerequisite of the successful enterprise engineering process.

The Product Life-Cycle Matrix (matrix #4) should be defined and disseminated together with the Generic Enterprise Reference Architecture and Methodology. Furthermore, matrix #4 would allow to compare product modelling languages and their scopes (e.g. STEP [11]), showing the connection with the product life-cycle, and with the enterprise life-cycle.

As an important result, this approach unifies product model-based and enterprise engineering-based integration.

Are All the Matrices Identified?

To find out whether there is a matrix below matrix #4 we would need to know the outcome of the "operation" of the product. Since matrix #4 describes all types of products there is no common *engineering* concept that grasps the outcome. In this sense the product model is fundamental to enterprise modelling.

To find out if there is a matrix above #1 we need to identify what operation brings about strategic enterprise management (SEM). We believe that processes which bring about SEM are beyond the realm of engineering and the enterprise – they belong to politics and education. Therefore the four matrices do cover the entire scope of enterprise integration.

Further Connections Between the Life-Cycles Identified

When developing the present form of the matrix representation it was the target of considerable debate whether to include the "build" phase in the life-cycle models or not? CIM-OSA did not include this in the CIM-OSA cube, nor did the first version of the matrix presentation as contributed to [2]. This controversy can be resolved now:

- The life-cycle model includes all major phases of building the enterprise thus building it can not be left out (“build” should be in the life-cycle matrix).
- The descriptions of each phase are about the entity under consideration. E.g., if looking at matrix #4 then in each phase there should be some description of the product, because the matrix describes the product’s life cycle.

  This is not true of the build phase. The description given there is *not what the product is* but *how the product is built.*

  However, the ”build” phase is actually a part of the ”operate” phase of the matrix above. That is: by operating the enterprise the products are built.

It therefore does no harm incorporating the ”build” models in the life cycle of any entity with the understanding that the build process is part of the operation one level above.

Identification of Deliverables in the Development of the Generic Reference Architecture and Methodology

As identified above the fully developed architecture and methodology would have to contain the following *deliverables*:

- Generic Enterprise Reference Architecture (GERA);
- Generic Enterprise Engineering Methodology (GEEM).
- Generic Enterprise Modelling Tools and Languages (GEMT&L);
- Generic Enterprise Models (GEMs)

In its report and subsequent article [1] the IFIP/IFAC Task Force identified two possibilities to develop the required complete architecture and methodology:

- develop a new generic architecture and methodology, or
- combine existing ones.

Both technically and strategically the combination of existing architectures and methodologies has great advantages and this line is followed from here on. However, the strategy to develop GERA is not necessarily the same as the strategy to develop GEEM or the other deliverables.

The respective strategies should be acceptable to the community of users and developers which in turn depends on the following criteria:

- the quality and adequacy of the proposed deliverables for the task at hand
- the apparent complexity: the proposed development must clarify and not obscure the area.
- the wish to preserve the value of investment in the respective (partially incomplete) reference architectures and to contain the cost of this further development.
- the need to allow to preserve the advantages which result from having accumulated knowledge and intellectual property in the respective teams
- the need to respect the intellectual contributions of those who were the main engineers of the respective methodologies and abstin from trying to create a Procrustean bed.
- the need to progress with the maximum benefit to the developers and users.

It is possible to give separate consideration to each of the *deliverables* respecting in each case the above strategic requirements.

The development of the Generic Enterprise Reference Architecture (GERA)

The building of GERA should be similar to building a car-race track and defining the rules of the game rather then building a Formula-1 T-model of enterprise engineering. GERA is an open system framework.

GERA must be a public domain document (possibly considered for standardisation) to allow meaningful and comparable competition to occur in the area of enterprise engineering. Similar goals were adopted by projects as CIM-OSA and the existing links to standards bodies (CEN-CENELEC, ISO TC184 etc) must be built upon as well as links to other groups involved in enterprise modelling (ICEMT SIG).

Examples for candidate competitive products which could arise out of GERA, or could use GERA as a leverage: Computer-aided Enterprise Engineering, Enterprise Modelling and Project Management tools; Integration Platforms; Enterprise (re)Engineering Know-how; Educational Material etc.

Milestones to develop GERA:

- Take the Purdue Enterprise Reference Architecture as the most complete in scope and add to it the areas discovered in the four matrices. This will include the development of the complete specification of GERA[7] (i.e. including the concept, identification and requirements and design of GERA). The specification will also have to include the definitions of the other deliverables.

[7] The present article can be built upon.

- Develop two presentations:

  Presentation of GERA for the user community:

  For the user community, preferably use the two tiered diagram format (now presenting four such diagrams according to the four matrices). The two tiered diagram format will *group* together those "subject matter" areas which will belong together in the methodology. This presentation can be prepared on the basis of the PERA document by adding further chapters.

- **Presentation of GERA for the developer community:**

  The second presentation is for the methodology-, modelling-, and standardisation community, preferably in the form of the four matrices. The matrix presentation will be seen as a further development of the CIM-OSA cube. The matrix presentation will use a refined subject matter presentation drawing on the one hand on the corresponding CIM-OSA definitions and on the above subject matter groups (as developed for the user community).

The Development of a Generic Enterprise Modelling Technology (GEMT) and Generic Enterprise Models (GEM)

The development of GEMT is not independent of the most generic enterprise models (GEM) because these are the meta-models for the language design (the ontological theories define the semantics of the language). For this reason the two are dealt with together.

The GEMT is a collection of modelling tools (languages / templates) which cover the needs of GERA. The definition can be done on two levels.

The first level is the definition of suitable languages for each area separately. This is expected to be done for each area in GERA, at least for those subject matters which require model development in the application phase of the generic methodology (minimum requirement).

The second level is where the common meta-model of the modelling sub-languages is expressed as views of a common meta-model. If this is done then far more possibilities exist for enterprise engineering tools to analyse, execute (e.g. simulate), cross check, and validate models then in the first case. This integration of languages is also necessary for a smooth transition from the design to the building and operating of enterprise models.

The second case *also* allows for the adoption of the General Modelling System and Permissive Modelling concept [4], which are based on the recognition that designers need to be able to use a wide variety of languages to express themselves. Once a core set of languages (CIM-OSA) is expressed as views of the underlying meta-model, it will be possible for other views to be developed as add-ons. Information contributed to the enterprise design database using either the core languages or the add-on languages will all be added to the underlying design database and thus becomes visible through the defined views.

A major part of necessary languages was developed in the CIM-OSA project and in GIM (organisational/decisional modelling), and some additions will have to be considered (e.g. temporal behaviour, cost/economy). In other words, the modelling technology of GERA should be an open system as well.

The granularity of modelling is a problem at the moment. CIM-OSA strives at model executability down to the level of enterprise operation. This requirement limits CIM-OSA's ability to incorporate in the modelling language anything more then a simple ontological theory of events/actions and data (for efficiency reasons).

TOVIE also strives at executability but is less concerned with the efficiency of execution. For this reason a complex ontological theory of actions, time, cost etc. is part of the meta-model.

The tension between the two approaches is apparent and a possible solution is proposed here. However, further investigation is necessary in this regard. It is proposed to express CIM-OSA modelling languages[8] as views of a more complex ontology (TOVIE) then the CIM-OSA meta-model. In this way a design system based on CIM-OSA models could take advantage of the added semantics of CIM-OSA concepts, and the improved analysis capability in the *design* phase. However, for efficiency, the simpler, compatible, CIM-OSA meta-model could form the basis for building and operating enterprise control software.

For the user of GEMT there would be no difference seen, but for the developer of enterprise engineering tools and integration platforms the differentiation would be visible.

Milestones to develop GEMT and GEM:

Given the above uncertainties the milestones need to be re-evaluated in light of the feasibility of the first few steps.

- Determine which modelling tasks are not currently addressed by CIM-OSA or GIM.
- Identify the complementing set of languages as needed. Draw on the IDEFX set of languages.
- Feed the result into the definition of the Generic Enterprise Engineering Methodology (GEEM) project. This ensures that the existing set of modelling constructs will be met by a compatible presentation of the Enterprise Engineering Methodology.

  Up to this point this is a feasible engineering project. The following tasks, however, need to be addressed by research.
- Integrate these languages on the meta level – possibly using ontological theories developed in TOVIE or IDEFX. Since the core CIM-OSA languages are already based on one integrated meta-model this is not a project from scratch.
- Express the languages as views of an underlying ontology.

[8] And the additional ones which might not at the moment be part of CIM-OSA but required by GERA.

One suitable expression of the modelling concepts of the languages so defined is the definition of the basic language concepts as *modelling classes* in a single object-oriented language[6].

This step need not necessarily lead to the design of any new language (although some extensions may have to be done both on the language and the meta-model level.)

- Extend the underlying ontology as required.

The Development of a Generic Enterprise Integration and Engineering Methodology (GEEM)

It is acknowledged that more then one methodology may exist to cover the GERA framework. The present proposal points out how one such methodology can be developed with relatively small expenditure, drawing upon the Purdue and GIM methodologies. Other, equally viable proposals could be presented on the basis of only one of them or based on other combinations with proprietary methodologies.[9]

The first goal of developing GEEM is to provide one complete methodology which covers the entire GERA architecture. The coverage of the methodology may be achieved by a./ adding more information to the Purdue Implementation Procedures Manual b./ adopting compatible methodologies which complement the Purdue Implementation Procedures Manual with methods in the areas not addressed (or not addressed in sufficient detail).

The second goal developing GEEM is to *specialise* the methodology such that a consistent set of modelling languages is proposed along with the presentation of procedures. This will allow the addition of examples more easily. The second goal supposes that the GEMT project provides the necessary input. However, because the basis of that is the CIM-OSA modelling languages, this is not a heavy constraint.

Milestones to develop GEEM:

- Determine, given GERA, the areas which should but are not yet covered by the Purdue Implementation Procedures Manual (IPM) [10]
- Extend the methodology regarding the specification and design of the decision system of the enterprise, based on GIM where these details are highly developed and tried.
- Based on the extensions to the CIM-OSA modelling framework (GEMT) illustrate, with examples and specific instructions, the steps of the methodology.

[9] It may even be that a particular enterprise integration project will not use a single generic methodology to follow, but develops an in-house, individual solution by mixing and matching of more limited methodologies.

References

[1] T.J. Williams P. Bernus, J. Brosvic, D. Cheng, G. Doumeingts, L. Nemes, J.L. Nevins, B. Vallespir, J. Vlietstra, D. Zoetekouw, "Architectures for Integrating Manufacturing Activities and Enterprises," Prepr. IFAC World Congress 1993, Sydney, Vol.X. pp273-283 (1993) also in *Design of Information Infrastaructure Systems for Manufacturing*, Prepr. IFIP WG 5.3 Working Conference, Tokyo (November 1993)pp1-17.

[2] T.J. Williams P. Bernus, J. Brosvic, D. Cheng, G. Doumeingts, L. Nemes, J.L. Nevins, B. Vallespir, J. Vlietstra, D. Zoetekouw, "Architectures for Integrating Manufacturing Activities and Enterprises," Technical Report of the IFIP/IFAC Task Force, International Federation of Automatic Control and International Federation of Information Processing,Technical Report, Theodore J. Williams, Editor, Purdue University, West Lafayette, Indiana, USA (March 1993) approx 500 pages.

[3] Williams, T. J., "The Purdue Enterprise Reference Architecture," Instrument Society of America, Research Triangle Park, NC (1992).

[4] Bernus, P., Rigour and Permissiveness in the Design of CAD/CAM Systems, **in***Integration of CAD/CAM*, D.Kochan (ed), North Holland, Amsterdam (1983)pp161-178.

[5] Fox, M.S., "The TOVIE Project, Towards a Common Sense Model of the Enterprise," in *Enterprise Integration*, Ch. Petrie (Ed), Cambridge, MA, MIT Press (1992)

[6] K. Mertins, W. Sassenguth, R. Jochem. An object- oriented method for integrated enterprise modeling as a basis for enterprise coordination. In Enterprise Integration Modeling (C. Petrie, Ed.). The MIT Press, Cambridge, MA. 1992. pp. 249-258.

[7] F. Vernadat. CIMOSA - A European development for enterprise integration. Part 2: Enterprise modelling. In Enterprise Integration Modeling (C. Petrie, Ed.). The MIT Press, Cambridge, MA. 1992. pp. 189-204.

[8] Vallespir,B., Chen,D., Zanettin.M., Doumeingts,G., "Definition of a CIM Architecture within the ESPRIT Project 'IMPACS'," in *Computer Applications in Production Engineering: Integration Aspects*, G.Doumeingts, J.Browne, M.Tomljanovich (Eds), Elsevier, Amsterdam (1991)pp731-738.

[9] IDEF0,1,2,3,4,5

[10] Industry-University Consortium, Purdue Laboratory for Applied Industrial Control, "An Implementation Procedures Manual for Developing Master Plans for Computer Integrated Manufacturing," Report Number 155, Purdue Lab. for Applied Industrial Control, Purdue University, West Lafayette, IN (June 1992).

[11] Seitz,B., Kirley,J., "STEP Framework – Concepts and Principles," ISO/TC 184/SC4/WGS/Pl document N6, (July 1991)