



HAL
open science

A Metamodel for Enterprise Architecture

Peter Bernus, Ovidiu Noran

► **To cite this version:**

Peter Bernus, Ovidiu Noran. A Metamodel for Enterprise Architecture. IFIP TC 5 International Conference on Enterprise Architecture, Integration and Interoperability (EAI2N) / Held as Part of World Computer Congress (WCC), Sep 2010, Brisbane, Australia. pp.56-65, 10.1007/978-3-642-15509-3_6 . hal-01054819

HAL Id: hal-01054819

<https://inria.hal.science/hal-01054819>

Submitted on 8 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A Metamodel for Enterprise Architecture

Peter Bernus and Ovidiu Noran

Griffith University, Nathan (Brisbane)
Queensland 4111, Australia
P.Bernus@griffith.edu.au, O.Noran@griffith.edu.au

Abstract. The paper presents an evolved metamodel that has its origins in GERAM 1.6.3 but takes into consideration the needs of evolution of EA standards, including ISO 15704:2000 and ISO 42010:2000 – both currently under review.

Keywords: Enterprise Architecture Framework, metamodeling, GERAM

1 Introduction

This paper aims to present an evolved metamodel of GERAM. The original GERAM document that formed the basis of ISO 15704:2000 had an informally expressed metamodel (IFIP-IFAC, 1999; Fig 1) but recent trends in standards development require that the metamodel be expressed in a formal way. Such formalisation, as can be expected, brings out previously unclarified details, including the details of the difference between life cycle phases and life history stages, milestones etc. Also a more formal definition of modelling frameworks is given. In addition, the attempt to harmonise with ISO 42010:2000 brings a new insight into the relationship between enterprise models and stakeholders who are users of these models. As a side-effect of this clarified terminology, the paper also presents a new understanding of EA frameworks that are based on the life cycles of enterprise entities and the Zachman framework.

2 History

Real-world enterprises are inherently complex systems. To tackle this complexity a variety of proposals were developed in the 1980s and 1990s, and these proposals fell into two categories: (a) proposals that created generally applicable ‘blueprints’ (later to be called reference models, partial models, or ‘architectures of type 1’) so that the activities involved in the creation (or the change) of the enterprise could refer to such a common model (or set of models); (b) proposals which claimed that to be able to organise the creation, and later the change, of enterprises one needs to understand the life cycle of the enterprise and of its parts. These latter were the proposed ‘architectures of type 2’, or more intuitively ‘life cycle architectures’ (cf IFIP-IFAC

Task Force, 1999). This second type of architecture was at the time called an 'Enterprise Reference Architecture'. Several proposals emerged in those two decades – e.g. PERA (Williams 1994), CIMOSA (CIMOSA Association 1996), ARIS (Scheer 1999), GRAI-GIM (Doumeingts, 1987), and the IFIP-IFAC Task Force, based on a thorough review of these as well as their proposed generalisation (Bernus and Nemes, 1994) developed GERAM (IFIP-IFAC Task Force, 1999) which then became the basis of ISO15704:2000 "Industrial automation systems – Requirements for enterprise-reference architectures and methodologies". While the name suggests that this standard is about 'industrial automation', this standard is in fact applicable to any man-made system (enterprise, product, project, etc). Curiously, independently from this, John Zachman developed his Information Systems Framework (Zachman 1987), which later was realised to be applicable to any enterprise (product, project, etc) as well.

While neither GERAM (and its inceptors) nor the Zachman Framework were originally called an 'Enterprise Architecture Framework (AF)', this is the current term used to describe the metamodel that defines the terminology of EA. Note that as (Noran, 2003) shows, the Zachman Framework is not exactly what we would call a 'life cycle architecture', there is a clear connection between a life cycle architecture and the Zachman Framework, which makes either of these qualify as Architecture Frameworks. The 'technical trick' of ISO15704:2000 is that it is actually Framework agnostic: while it incorporates GERAM as an appendix, the normative part of the standard only lists the *requirements* that any architecture framework should satisfy. This is very important because it would be unreasonable to expect that investments into adopting a framework will be abandoned by organisations just because a new standard appeared. However, at the same time, the developers of frameworks can use the standard to evolve their own frameworks and therefore the standard has a harmonising effect on the language / terminology of enterprise architecture, as well as can be used to make EA frameworks more complete than they would be without such definition.

Later developments in the EA domain saw the inception of C4ISR – now DoDAF (DoD Architecture Framework Working Group, 2004), and TOGAF (The Open Group, 2006) to name only a few of the popular ones. Mappings of most on GERAM / ISO 15704:2000 are available (although perhaps not popularised enough) and it is clear that neither of these are complete yet in terms of satisfying *all* ISO 15704:2000 requirements, however, there does not seem to exist an obstacle to their future evolution.

Architecture Frameworks have been used in many industries, including the domains of industrial automation / manufacturing / production management, business information systems (of various kinds), telecommunications and defence.

Part of the Enterprise Architecture practice is 'enterprise engineering' and the practice of 'enterprise modelling' (or just modelling) and complete AFs describe the scope of modelling (which later can be summarised as a Modelling Framework that is part of the AF). Note that (quite independently from the above), IEEE developed a standard in 2000 (IEEE 1471, 2000) that documents some important requirements that models describing the architecture of a 'software intensive system' should satisfy, and a recent development in ISO is the adoption of this latter IEEE standard as ISO42010 (currently [2010] under review). It should not be surprising to the reader that this

standard is also applicable to a much wider domain than originally thought (i.e., not only software systems).

In the past there have been various attempts to map the existing AFs and their associated artefacts against one another (e.g. (Williams, Zoetekouw et al. 1996)); such attempts have highlighted the difficulties encountered in the mapping process such as meaning, gaps, overlapping, etc.

Subsequently, efforts were made to map the AFs against a neutral reference, able to accommodate all possible types of artefacts contained in the mapped architectures. This reference has been constructed by essentially combining all the features of the main existing architectures, filtering out coverage overlaps and adding missing (and thought to be necessary) aspects. Typically, some areas are well covered and understood in all frameworks, while others are not, for reasons relating to framework history, purpose, intended audiences, underlying framework ontologies, etc. For example, function and information are fairly well understood in all frameworks, while human and decisional aspects are not.

The result of the mappings was a matrix-like structure of requirements (Bernus and Nemes, 1994 and 1996), which has further improved understanding of the frameworks and their problems; however, the result was rather complex and difficult to follow.

A more space-efficient and user-friendly, three-dimensional structure was then proposed in order to improve and simplify the previous flat and tabular MF. Subsequently, the new MF was supplemented with specific concepts such as entity type, recursion, life history, etc. to create a true AF.

The AF was obtained by putting together the generalised MF and all essential generic concepts of enterprise engineering, such as enterprise models, modelling languages, generic enterprise modelling concepts, partial models, etc. The current outcome of these efforts is the generalised reference AF described in Annex A of ISO15704:2000 and 2005, as an example of a framework compliant with that standard.

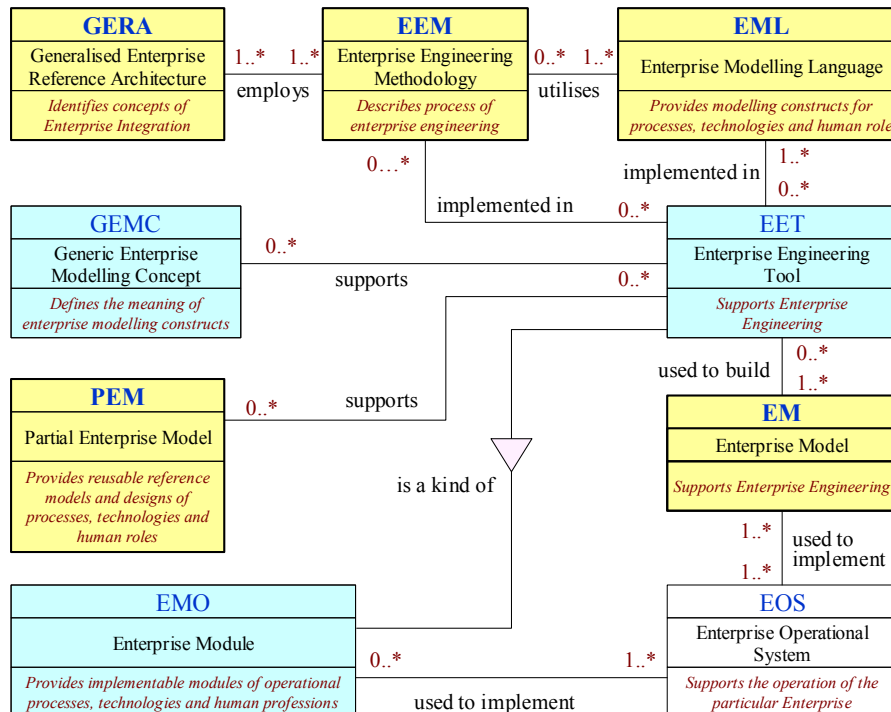


Fig. 1. A possible metamodel of GERAM (Noran, 2003) based on (GERAM 1.6.3)

Among others, GERAM has been used in practice to guide Enterprise Architecture (EA) projects (Bernus, Noran et al. 2002; Noran 2004; Mo 2007; Noran 2007), harmonize international standards, assess other enterprise AFs (Noran 2003; Noran 2005; Saha 2007) and to build structured repositories of AF elements for a project management decision support system (Noran, 2009). For a complete description of GERAM see ISO15704:2005 (ISO/IEC 2005).

3 The Need for Change

Enterprises are highly complex and dynamic entities. The continuous maturing and evolution of the EA domain reflects this; thus, existing AFs are adapted and enriched and new AFs are being created to reflect the new business environment's challenges. Zachman, TOGAF, etc. are just a few examples.

In addition to mandatory reviews, international standards in this domain go through similar change processes. For example, ISO42010 (ISO/IEC 2007) evolved from IEEE1417 in order to set updated guidelines for architecture descriptions of software-intensive systems.

The AF domain understandably displays signs of competition between the various AFs. This unfortunately translates in the difficulty for the typical user to achieve a

clear understanding of the main purpose and domain covered by each AF, and problems in employing a combination of AFs (or parts thereof) for specific projects. In the past GERAM has been used to try to classify some of the above-mentioned AFs and thus facilitate their use for specific tasks. However, for this endeavour to succeed there is a need for GERAM and ISO15704 themselves to be updated to keep up with the changes in the AF domain.

In the standards area, harmonisation rather than competition is (and has been) the main issue. Ongoing efforts attempt to reconcile and eliminate gaps and overlaps between various related standards (such as e.g. ISO15288 – systems life cycle processes and ISO12207 – software life cycle processes) using different terminology and levels of abstraction due to historic and other reasons. As part of the standardisation effort and a possible tool for reconciliation, ISO15704 and its Annexes (eg GERAM) must also go through a process of harmonisation with other relevant standards – a prominent example being ISO42010 (ISO/IEC 2007).

This paper attempts to describe a possible way forward in the evolution of GERAM and ISO15704 by proposing an enhanced and more formal description of the artefacts involved and their relationships, including some extension to show the relationship to ISO 42010.

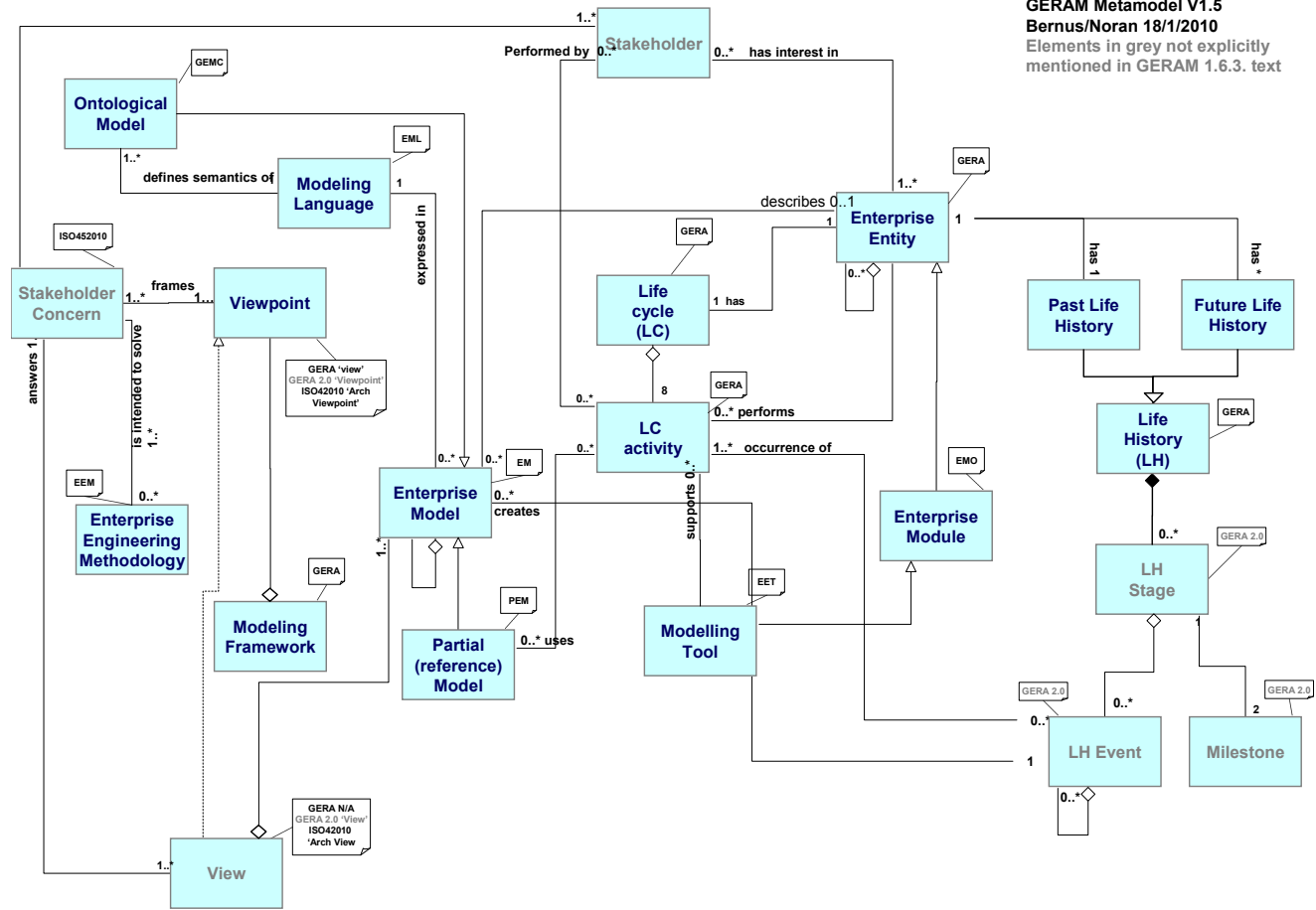
4 The Proposed Metamodel

It has been considered that due to the amount and significance of changes involved, the next version of GERAM should be designated as ‘GERAM 2.0’. The proposed changes encompass:

- a clarification of several ambiguous terms;
- more formal representation of the relations between fundamental components of GERAM;
- the introduction of several concepts and terminological equivalences in order to harmonise GERAM and ISO15704 with efforts in other areas, notably ISO42010.

Fig. 2 attempts to present the above changes in a metamodel created using a UML (OMG, 2005) class diagram. The figure presents in effect a combined AS-IS / TO-BE (present / future) state of affairs. Therefore, the notes attached to elements are crucial to the understanding of the metamodel as they show whether an element exists in the current version of GERAM 1.6.3 and whether that element is to be added / removed / kept in the next version (GERAM 2.0). The use of UML and specification of all multiplicities assists the formal representation of the relation between artefacts.

Fig. 2. Proposed metamodel v1.5



GERAM Metamodel V1.5
 Bernus/Noran 18/1/2010
 Elements in grey not explicitly
 mentioned in GERAM 1.6.3. text

4.1 A More Formal Representation of the Metamodel

4.2 New and changed Artefacts

The new metamodel proposed to introduce several new artefacts and changes the designation and/or *meaning* for others in order to achieve better stakeholder understanding and terminological harmony with other relevant standards. For example the term view in GERAM 1.6.3 was equivalent to the term ‘viewpoint’ in ISO42010. Therefore it is proposed to be changed to viewpoint while view as a term will remain with the meaning of architectural view corresponding the to terminology of ISO 42010.

The concept of life history has been present in the version 1.6.3 of GERAM. However, it now has been considered beneficial to explicitly specify that while the past life history is unique (cannot be changed), the future life history is a matter of choice between several scenarios (and of course, a proportion of chance).

ISO 15704 and GERAM has used the term life cycle phase to denote life cycle activities. The term life history *stage* is used to denote a time interval on the life history of an entity (or the time interval within a sequence of events in the life history) as the case may be. Thus the occasional misuse of the term ‘phase’ in the temporal context is eliminated.

4.3 Clarification of Relations between Artefacts

The new metamodel allows the enrichment of relationship representation by using aggregation, interface and specialisation / abstraction. Thus, it is now possible for example to represent the fact that a life history event, an enterprise model or an enterprise entity can be decomposed. It is also possible to represent the fact that a collection of viewpoints can be contained in an MF or that the life cycle of an enterprise is in fact a collection of life cycle activities. Similarly, the model also communicates that a set of enterprise models can form a view that answers a stakeholder concern.

The specialisation of reference (partial) models and ontological models in enterprise models can now also be represented. Likewise, it can now be shown that modelling tools are a special type of enterprise module (trusted off the shelf executable components) and that an enterprise module is a type of enterprise entity (consistent with the ISO 15288 component concept) .

Given the concept of Stakeholders and their relationship to models it is now clear that in the Zachman Framework the rows represent views of enterprise models rather than enterprise models themselves. This latter is a possible cause for controversy, but careful consideration shows that the problem is avoidable. Namely, Zachman insists that the cells in the Zachman Framework should contain ‘elementary’ models (models that are not a combination or the consequence of other models). On the other hand, stakeholders (according to ISO 42010 and ISO 15704) wish to see extracts of enterprise models which extracts together form a view through which the stakeholder’s

concerns are satisfied. However, one needs to consider that stakeholders both create ('author') and peruse ('read') parts of such models, thus Zachman's insistence is on stakeholders *authoring* elementary models, not on exclusively reading ones.

4.3 A Classification of Model types

Fig. 3 proposes one possible classification of models using the interpretation of the viewpoint concept by type (nature of the model) and by scope (i.e. the limit of the modelling). This is only one of the many possible taxonomies. Note, as mentioned above, that the proposed metamodel calls it a viewpoint what used to be a view in GERAM 1.6.3. The taxonomy proposed is but one possible way of classifying model types (e.g. a decisional model type is defined here as having both organisational [who is the decider] and functional aspects [what decision function is performed], this is expressed using the UML 'interfac' notation. However, model type classification may have alternatives, and is not a crucial element of the proposed metamodel.

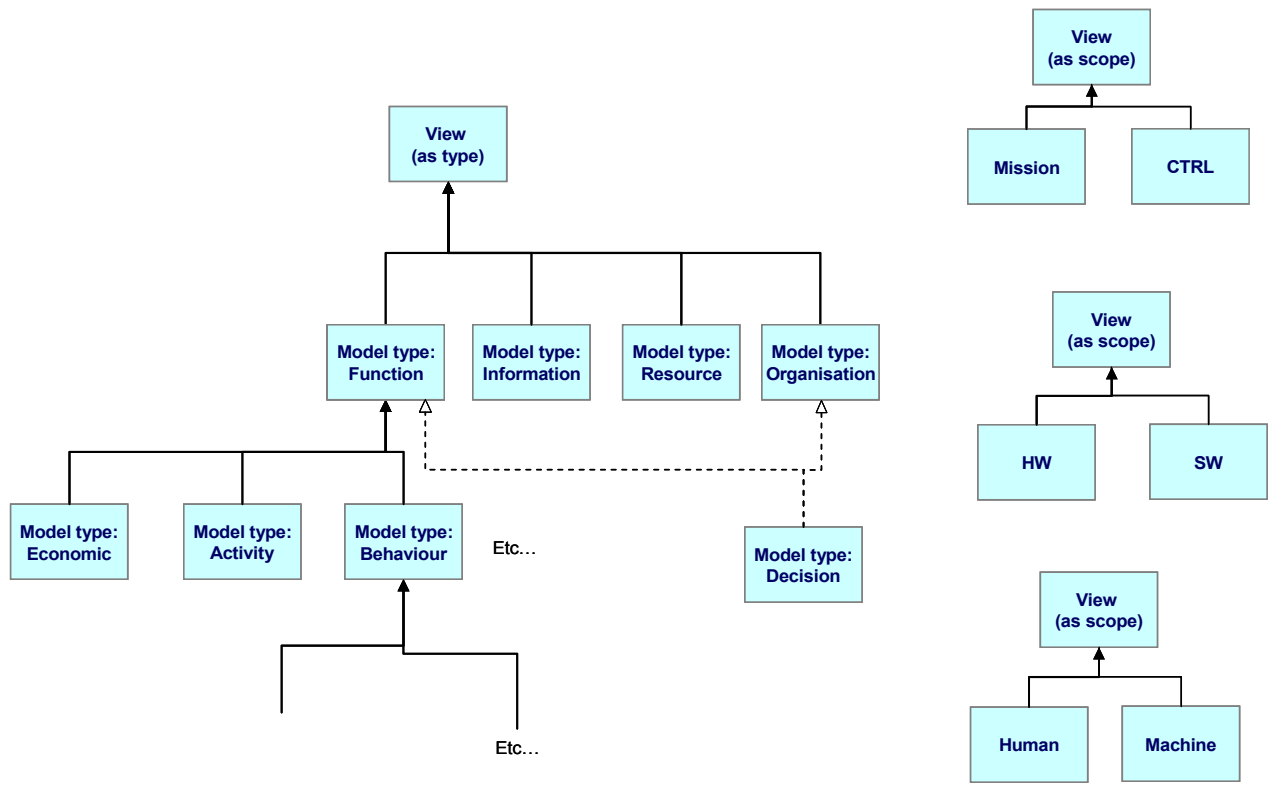
A model that belongs to a model type is claimed to be able to answer certain questions about the entity it describes. The ontological theory behind the kind of question that can be answered is defined (in ontology design) using a 'set of competency questions' (Gruninger and Fox, 1995, 1998). Based on this set, the ontological theory defines the semantics of the language in which the model is expressed

A stakeholder concern can be answered by asking a set of competency questions. This set can be further subdivided into subsets (according to rules provided by viewpoints) in order to construct the models necessary to answer the concern in question.

5 Conclusions and Future Work

This paper has proposed an updated and enhanced metamodel for GERAM. Future work will further develop and refine the metamodel according to feedback and testing in case studies. An important consideration of this work is its usability to harmonise several standards' terminologies, including ISO 15288 (Systems Life cycle processes), ISO/IEC 42010 (Architecture Descriptions of Software Intensive Systems), ISO 12207 (Software Life Cycle Processes) as well as advancing the terminology of Enterprise Architecture as a discipline.

Fig. 3. Possible model typology



References

- Bernus, P., Nemes, L. (1996a) A Framework to Define a Generic Enterprise Reference Architecture and Methodology. *Computer Integrated Manufacturing Sys* 9(3). pp 179-191
- Bernus, P., Nemes, L. (1994) A Framework to Define a Generic Enterprise Reference Architecture and Methodology In Proc. ICARV'96 4th Int. Conf. on Control, Automation, Robotics and Vision Vol 3/3. (4-6 December 1996). Singapore : Nanyang Technological University. pp88-92
- Bernus, P., Nemes, L. and Morris, B. (1996b). The Meaning of an Enterprise Model. *Modelling and Methodologies for Enterprise Integration*. P. Bernus and L. Nemes (Eds) London : Chapman & Hall: 183-200.
- Bernus, P., Noran, O., *et al.* (2002). Using the Globemen Reference Model for Virtual Enterprise Design in After Sales Service. *Global Engineering and Manufacturing in Enterprise Networks (Globemen)*. VTT Symposium 224. I. Karvoinen, R. van den Berg, P. Bernus et al. Helsinki / Finland: 71-90.
- CIMOSA Association (1996). "CIMOSA - Open System Architecture for CIM,. Technical Baseline, ver 3.2." CIMOSA Association.
- DoD Architecture Framework Working Group (2004). DoD Architecture Framework Ver 1.0..
- Doumeings, G., Vallespir, B., Darracar, D., and Roboam, M., (1987) "Design Methodology for Advanced Manufacturing Systems" *Computers in Industry*. 9(4) pp.271-296.
- Fox, M.S., Gruninger, M., (1998), "Enterprise Modelling", *AI Magazine*, AAAI Press, Fall 1998, pp. 109-121.
- Gruninger, M. and M. Fox (1995). The logic of Enterprise Modelling. *Proceedings of the Industrial Engineering Research Conference*. Atlanta, CA.
- IEEE (2000) IEEE Std 1471-2000 IEEE Recommended Practice for Architectural Description of Software-Intensive Systems
- IFIP/IFAC (1999) GERAM (version 1.6.3): Generalised Enterprise Reference Architecture and Methodology. Available at <http://www.cit.griffith.edu.au/~bernus/taskforce/geram/versions> also published as Chapter 2, *Handbook on Enterprise Architecture* (2003) Bernus,P., Nemes,L. and Smidth, G (Eds). Berlin Springer. pp22-64.
- ISO/IEC (2005). Annex C: GERAM. ISO/IS 15704:2000/Amd1:2005: Industrial automation systems - Requirements for enterprise-reference architectures and methodologies.
- ISO/IEC (2007). ISO/IEC 42010:2007: Recommended Practice for Architecture Description of Software-Intensive Systems.
- ISO/IEC 12207 (1995, 2008) Systems and software engineering – Software life cycle processes
- Mo, J. (2007). The use of GERAM for Design of a Virtual Enterprise for a Ship Maintenance Consortium. *Handbook of Enterprise Systems Architecture in Practice*. P. Saha. Hershey, USA, IDEA Group: 351-366.
- Noran, O. (2003). A Mapping of Individual Architecture Frameworks (GRAI, PERA, C4ISR, CIMOSA, Zachman, ARIS) onto GERAM. *Handbook of Enterprise Architecture*. P. Bernus, L. Nemes and G. Schmidt. Heidelberg, Springer Verlag: 65-210.
- Noran, O. (2004). A Meta-methodology for Collaborative Networked Organisations: A Case Study and Reflections. *Knowledge Sharing in the Integrated Enterprise: Interoperability Strategies for the Enterprise Architect*. P. Bernus, M. Fox and J. B. M. Goossenaerts. Toronto / Canada, Kluwer Academic Publishers: 117-130.
- Noran, O. (2005). "An Analytical Mapping of the C4ISR Architecture Framework onto ISO15704 Annex A (GERAM)." *Computers in Industry* 56(5): 407-427.
- Noran, O. (2007). Discovering and modelling Enterprise Engineering Project Processes. *Enterprise Systems Architecture in Practice*. P. Saha. Hershey, USA, IDEA Group: 39-61.
- Noran, O. (2009). "A Decision Support Framework for Collaborative Networks." *International Journal of Production Research* 47(17): 4813-4832.
- OMG (2005). Unified Modelling Language version 2.0 specification, OMG. 2010.

- Saha, P. (2007). A Synergistic Assessment of the Federal Enterprise Architecture Framework against GERAM (ISO15704:2000 Annex A). *Enterprise Systems Architecture in Practice*. P. Saha. Hershey, USA, IDEA Group: 1-17.
- Scheer, A.-W. (1999). *ARIS-Business Process Frameworks*. Berlin, Springer-Verlag.
- The Open Group (2006). *The Open Group Architecture Framework, (TOGAF 8.1.1 'The Book') v8.1.1*.
- Williams, T. J. (1994). "The Purdue Enterprise Reference Architecture." *Computers in Industry* 24(2-3): 141-158.
- Williams, T. J., D. Zoetekouw, et al. (1996). Techniques to map the architectures directly against one another. *Architectures for Enterprise Integration*. P. Bernus, L. Nemes and T. J. Williams. London, Chapman & Hall.
- Zachman, J. A. (1987). "A Framework for Information Systems Architecture." *IBM Systems Journal* 26(3): 276-292.