HAL open science

IFIP WG5.12 Architectures for Enterprise Integration: Twenty-Five Years of the GERAM Framework

Peter Bernus, Richard Martin, Ovidiu Noran, Arturo Molina

To cite this version:

Peter Bernus, Richard Martin, Ovidiu Noran, Arturo Molina. IFIP WG5.12 Architectures for Enterprise Integration: Twenty-Five Years of the GERAM Framework. Advancing Research in Information and Communication Technology, AICT-600, pp.245-268, 2021, 10.1007/978-3-030-81701-5_10 . hal-03325989

HAL Id: hal-03325989

https://inria.hal.science/hal-03325989v1

Submitted on 25 Aug 2021

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

IFIP WG5.12 Architectures for Enterprise Integration: Twenty-five Years of the GERAM Framework

Peter Bernus*, Richard Martin**, Ovidiu Noran*, Arturo Molina***
* Griffith University, Brisbane, AU
[P.Bernus, O.Noran]@griffith.edu.au
** Tinwisle Corp., Bloomington, IN, USA richardm@tinwisle.com
*** Tecnológico de Monterrey, México, D.F. armolina@tec.mx

Abstract. Abstract: Apart from the 25-year anniversary in 2019 of the first publication of the GERAM Enterprise Architecture Framework, the timeliness of this paper lies in the new interest in the use of systems theory in Enterprise Architecture (EA), and consequently, 'light-weight' architecture frameworks (AFs). Thus, this paper is about the use of systems thinking and systems theory in EA and about how it is possible to reconcile and understand, based on a single overarching framework, the interplay of two major enterprise change endeavours: enterprise engineering (i.e. deliberate change) and evolutionary, organic change. The paper also demonstrates how such change processes can be illustrated by employing systems thinking to construct dynamic business models; the evolution of these concepts is exemplified using past applications in networked enterprise building, and more recent proposals in environmental-, disaster- and healthcare management. Finally, the paper attempts to plot the way GERAM, as a framework to think about the creation and evolution of complex socio-technical systems of systems, will continue to contribute to society in the context of future challenges and emerging opportunities. †

Keywords: Enterprise Integration, Enterprise Architecture, Enterprise Modelling, Reference Architecture, Systems Engineering, Complex Systems.

1 Introduction: The History of GERAM

1.1 The Origins

The late 1970s and early 1980s saw a great effort in the manufacturing industry to design and build highly automated, computer-controlled systems of unprecedented complexity. The scope spanned from numerically-controlled machine tools and computer-aided design of parts and process planning, to complete factory automation featuring workshop- and factory level control, scheduling and planning (including

† This paper is an extended version of [1] and [2] to reflect the evolution of the GERAM framework between 2015 and 2019.

production planning and material requirements planning). As a result, the computer-integrated manufacturing (CIM, or 'enterprise integration' (EI)) movement has steadily evolved, increasingly demanding methods and tools to integrate the information and material flow throughout the enterprise.

In 1990, on recommendation by Jim Nevins (a former architect of the Apollo Guidance Computer, and Deputy Director of MIT Charles Stark Draper Lab), the International Federation for Information Processing (IFIP) and the International Federation for Automation and Control (IFAC) established a joint Task Force (TF) with the mandate to review existing architecting approaches to EI and to make recommendations to the industrial and research community. At that time, the IFIP constituent of the Task Force was operating under IFIP TC5 (Technical Committee on Computer Applications in Technology), and the IFAC constituent under TC-MI (Technical Coordinating Committee for Manufacturing and Instrumentation).

The TF included representatives from the industrial and research communities, with most participants having a mixture of research, engineering, industrial management, and consultancy backgrounds. The TF was first chaired by Ted Williams (Director, Purdue Laboratory for Applied Industrial Control) from 1992 to 1996 and later by Peter Bernus (Griffith University) from 1996 until 2002, until the TF completed its mandate. After 2002, the TF's legacy was inherited by IFIP WG5.12 ('Architectures for Enterprise Integration') and IFAC TC5.3 ('Enterprise Integration and Networking'). The TF reviewed several approaches developed for designing CIM systems and classified them into two categories as outlined in Section 1.2 [4][5]

1.2 From Type I to Type II architectures and their generalisation

The *first approach* was based on generic models, or designs, (called 'architectures') that could subsequently be implemented as information systems (IS) products (or product families) incorporating most or all information-processing tasks in the enterprise. These models typically covered the management and control system of the manufacturing enterprise, although the information flow implemented by the management and control system is necessarily a mirror of the material flow in the manufacturing enterprise. At the time, these models were called by the TF 'Reference Architectures of type I'. Such architectures appealed to vendors because the models produced allowed the definition of a stable set of software products.

The resulting implementations were called Enterprise Resource Planning (ERP) systems. In addition, a number of CIM system-specific Reference Models were developed, attempting to systematise the CIM systems' 'functional building blocks' (such as planning, scheduling and control system modules), thus giving rise to the set of products called 'Manufacturing Execution Systems' (MES). While on the higher levels of management & control there is a high level of uniformity among various enterprises and industries, the lower levels, close to the manufacturing process show great variety, dictated by the manufacturing technology and equipment at hand.

Unfortunately, even on the higher levels of control, the number of competing models ('Reference Architectures of type I') reviewed by the TF was in the order of several dozens (in the middle of 1990s), and neither of these achieved an industry-wide acceptance, or standard status. It is only recently that international standards were developed for the interface between ERP and Manufacturing Execution Systems

(MES), such as the IEC 62264 set of standards for enterprise-control system integration [6].

The second approach developed was based on the recognition that similarly to many engineering disciplines (chemical, manufacturing, software, civil, and systems-), enterprise engineering should also be based on a so-called 'life-cycle' approach. Accordingly, to design an integrated enterprise, the enterprise creation activities (and associated methodologies) should extend over the entire life of the enterprise – i.e., from its inception to its decommissioning. Several such architectures were developed by groups, some with manufacturing and control systems- and some with information systems (IS) background.

Therefore, the TF recommendation was that methodologies for designing and creating CIM systems should be based on architectures with a life cycle approach. These architectures were called by the TF 'Reference Architectures of type II'.

In particular, the TF identified three such architectures: CIMOSA (CIM Open Systems Architecture) [7] GRAI-GIM (Graphs with Results and Activities Interrelated) [8] and PERA (Purdue Enterprise Reference Architecture) [9], and recognised that these three had mutually complementary and useful characteristics. After deliberations, it was proposed that instead of trying to select and recommend a 'single best' reference architecture, the TF would serve the EI community better if it generalised its findings and codified these as requirements that any reference architecture should satisfy. Thus, authors of various proposed approaches could use these requirements constructively to evolve their own reference architectures.

Based on this idea, Bernus and Nemes in 1994 [10][11] developed the first such generalisation, which was adopted by the Task Force and called the 'Generalised Enterprise Reference Architecture and Methodology' (GERAM). The first comprehensive report was published in several articles and a monograph [4][5]. The TF decided to base its further work on this proposal and bring the specification of GERAM to its completion.

1.3 From GERAM to ISO15704 – 'Requirements for enterprise-referencing architectures and methodologies'

The TF realised that it must form links with appropriate standardisation bodies, and thus became a category 'A' liaison to ISO TC184/SC5/WG1 (called 'Industrial Automation Systems & Integration / Architecture, Communications, and Integration Frameworks / Modelling and Architecture'). Workgroup (WG)1 was keen to develop standards in the area and endeavoured to formulate the GERAM requirements as an ISO standard.

Between 1995 and 1999 the TF held the majority of its meetings jointly with ISO TC184/ SC5/WG1, chaired by Jim Nell of the US National Institute of Standards and Technology (NIST). The task standing before the TF was twofold: (1) complete the definition of GERAM, and (2) develop an ISO standard specifying the requirements that an (any) Enterprise Reference Architecture must satisfy. The result was GERAM 1.6.3, released in 1999 and ISO 15704:2000 [14] called "Requirements for Generalised Enterprise Reference Architectures and Methodologies" developed on this basis (three

years later GERAM 1.6.3 was later also published as a book chapter [15], thereby remaining in the public domain. The mandatory section of ISO 15704 lists requirements that any enterprise reference architecture must satisfy, while the GERAM document is a public appendix to the standard, exemplifying how to fulfil them.

In subsequent years, the same ISO committee extended ISO 15704 with additional views (decisional and economic), and also released ISO 19439:2006 [16], an ISO standard version of the GERA modelling framework (MF) that is part of GERAM.

Between 2015 and 2019, the work to update GERAM arrived at the current version 'GERAM-ISO' (as part of ISO 15704:2019 [17]), which incorporates time-tested provisions of GERAM into the normative requirements of ISO 15704, and this was carried out by ISO TC184/SC5/WG1 lead by Richard Martin, and contributions from IFIP WG5.12 members.

1.4 Other Relevant Developments in the field of EA

During the same period, the information systems (IS), software engineering, and systems engineering (SysEng) communities saw a number of similar efforts. Thus, for example, the IS community widely publicised the Zachman Framework [18] that initially targeted the enterprise IS, but was subsequently extended in an attempt to model the entire enterprise's architecture.

The evolved Zachman framework had a similar aim to the enterprise reference architectures of type II – although the stakeholder-role related 'rows' in the framework only corresponded to life cycle activities because life cycle abstraction levels were typically of concern to different stakeholders. Later there were several adaptations of the Zachman Framework, including the Federal Enterprise Architecture Framework [19].

The Defence community saw the development of the C4ISR, which became the basis of the DoDAF Architecture Framework, with originally (DoDAF1.5) three simple life cycle abstractions (operational, systems/services, technical), and a complex modelling framework (MF), subsequently replaced by eight viewpoints (that are nevertheless aligned with the original three). DoDAF, of which the latest release is v2.02 in 2010 [20], with several offshoots in the defence community (MODAF, DONDAF, NATO AF, etc).

More recently, the efforts of the Object Management Group (OMG) resulted in the so-called 'Unified Architecture Framework' (UAF)' [21], with the aim of including enterprise architecture in its scope (as opposed to being limited to 'systems architecture'). The claim is that UAF will eventually replace the above defence-related AF variations.

The IT industry also developed its own framework called The Open Group Architecture Framework (TOGAF) [22], which gained popularity with IT consultancies due to its detailed Architecture Development Method (ADM).

Consulting companies, individual researchers, and some vendors developed their own Architecture Framework (AF) versions (not listed here). The variety of proposals appears to owe to AFs being a conceptual structure deemed suitable for organising content for a domain, given typical stakeholders and their concerns, perspectives and viewpoints that determine how these stakeholders prefer to view and organise the answers to their concerns.

The American Heritage Dictionary of the English Language [23] defines a 'framework' as "*a set of assumptions, concepts, values, and practices that constitutes a way of viewing reality*." An EA framework is therefore a means of sense-making in the complex world of change, in the domain of EA. Similarly, as individual learning is a fundamentally constructivist process, organisations must construct their shared meanings of change-related concepts by building the new concepts of EA into the fabric of their own pre-existing, path-dependent concepts of organisational change.

The original terminology used by the manufacturing community called such constructs 'reference architectures'. However, the internationally accepted terminology has evolved; thus, in line with the above discussion, what ISO 15704 / GERAM and its predecessors called a 'Reference Architecture type II', is today called an Enterprise Architecture Framework (AF) (see for example ISO 42010:2011[‡] [24]). In today's terminology, GERAM is an AF (thus should perhaps more suitably called Generalised Enterprise Architecture Framework, 'GEAF') – where the origin of the acronym remains only of historical interest.

Notably, recent efforts in the Systems and Software engineering community developed ISO 42020 [25] for 'Architecture Processes' and ISO 42030 [26] for 'Architecture Evaluation', which are now complementary to ISO 15704:2019 and ISO 42010, defining concepts for the processes to develop and to evaluate the Architecture exhibited of an entity of interest. Note the terminological difference: instead of saying 'system of interest' these standards talk about an entity of interest, i) this aligns with ISO 15704's concept of Enterprise Entity, as well as ii) the declared extension of scope, where the entity of interest may not be a 'system', nevertheless has an architecture – such as for example as a product line.

Between 2015 and the end of 2019 the concerted effort and collaboration between the ISO TC184/WG1 and ISO/IEC JTC1/SC7/WG42 communities managed to harmonise their respective standards. The ticket to success was the fact that the underlying philosophy behind these communities is similar (with relatively small terminological differences), and it was acknowledged that the user community could benefit from such harmonisation. IFIP WG5.12 members have been participating in both of these standards bodies.

2. A Summary of the Contributions of GERAM to Theory and Practice

2.1 GERAM's contribution to the scientific community

The central problem of EA is how to harmonise and synthesise all the knowledge necessary for understanding, sense-making, leading, managing and executing change in enterprises as complex socio-technical systems of systems. In other words, the role of EA is to create coherency across management levels and activity domains in the

[‡] Note that ISO 42010:2011 is being reviewed (as of December 2020).

enterprise [27]. Therefore, for an AF to qualify as an enterprise AF, its scope must encompass any type of enterprise or constituent entity.

From a scientific point of view, GERAM's important contribution is that it defines a concise set of concepts that can be combined and used to evaluate AFs in terms of completeness of scope and complexity.

GERAM uses the concept of enterprise entity, defined to be any system of interest with socio-technical relevance (this may include systems of interest in the enterprise entity's environment); thus, even the product of the enterprise is seen as an enterprise entity. An enterprise entity may live for a longer period of time, or have limited life span, being dynamically created like a project or a programme; it can also be a single organisation (e.g. a company) or a virtual organisation (virtual enterprise). Certainly, a company is an enterprise entity, but so is a network of companies, a consortium, a government department, even an entire industry: essentially, any undertaking with a shared goal to produce some services, or products, can be considered an enterprise entity.

The constituency of an enterprise entity is usually changing in time: at any one point in time there may be entities joining or leaving, being created, transformed, or decommissioned; in other words an enterprise entity is often a dynamic organisation. For an AF to qualify as an EA framework, the scope of the framework must encompass all enterprise entity types.

The power of GERAM's life cycle generalisation is that the choice between what is the current entity of interest is orthogonal to all other choices. E.g., we can talk about the life cycle of an enterprise or the life cycle of a product in *exactly* the same way. This fundamentally reduces the complexity of architecture-related considerations, because i) there is no need to separately define product life cycle processes, project management processes, production processes and management processes; these are all life cycle processes of the respective entities (product, project, factory, company, etc. as illustrated in Fig. 1.), and ii) areas that previously eluded architectural treatment (and needed separate attention) become central elements of enterprise architecture (EA)).

This is illustrated in Fig. 2 by demonstrating that the combination of scope definitions of hardware/software and human/machine implemented functions and processes naturally leads to the inclusion into the architecture's scope of human skill levels, human knowledge, and role-specific communication requirements. As opposed to this, in frameworks lacking the orthogonality property, such considerations need separate metamodel level definitions, leading to excessive complexity (and possible incompleteness).

GERAM as an AF allows a number of elementary differentiations to be made, which can be combined into more powerful concepts as needed. In the GERA MF (which is one of the parts of GERAM), the underlying elementary concepts are derived from pairwise differentiations, mandating the ability to describe:

- human & automated constituents,
- service delivery & management constituents, and
- software & hardware constituents

of any enterprise entity, and the ability to do this over the entire life of the entity.

Furthermore, any combination of these differentiations can be described from the point of view of the *function* (and associated objects processed, such as *information* and *material*), the *resource* performing the function, and the mapping between function and

resource (called *organisation*). These differentiations define the scope of *potential* enterprise models, aiming for the scope to be as complete as possible and the framework to remain holistic – i.e., considering enterprise entities in their totality.

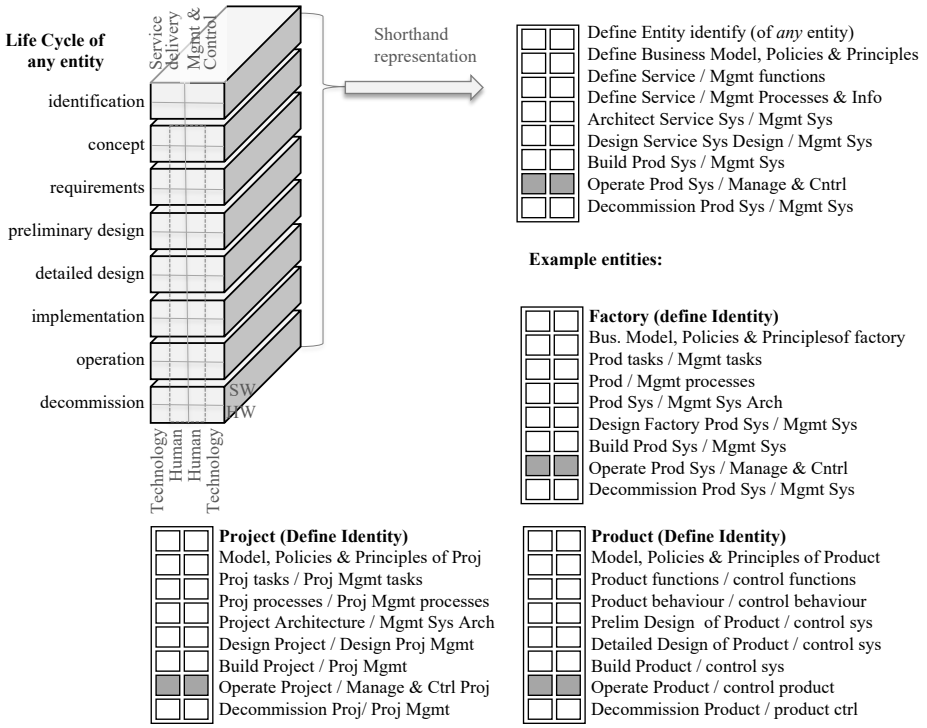


Fig. 1 Life cycle activity types and example of the scope (coverage) of their outcomes for various enterprise entities

According to Ted Williams, when the Purdue Consortium set out to write a methodology for CIM systems development it transpired that separately designing the automated parts of 'material and information processing systems' was too complex a task, and an approach was needed that simultaneously considered the human and automated elements of the enterprise.

If a framework, due to its limited scope, insists on subdividing a complex system-of-systems along a border where there are too many interconnections (such as the human-machine interface), then the designer will have limited ability to reduce apparent complexity through decomposition and modelling of interactions among the system's constituents. Thus, the advice is: '*don't cut it where it is the thickest*' – this was the first impetus behind the development of the Purdue Enterprise Reference Architecture (PERA) [9], leading to the incorporation into the scope of the AF both the human and the automated parts of the enterprise.

Finally, GERA states that descriptions (or *enterprise models*) may be created for an individual entity (called particular *models*), for a type of entity (called '*reference*

models' or '*partial models'*), as well as the *models of the languages* used for expressing these models. GERAM calls these latter 'Generic Enterprise Modelling Concepts', as they are essentially *ontologies*, which (in increasing extent of formality) may take the form of terminology definitions in natural language, metamodels, or formal ontological theories.

Note that the term 'partial model' is preferable to the term 'reference model', because a reference model is often associated with the situation where a particular system of interest is a parametric specialisation (instantiation) of the reference. Partial models on the other hand can be model fragments, building blocks, or patterns, allowing modular construction of new types of models.

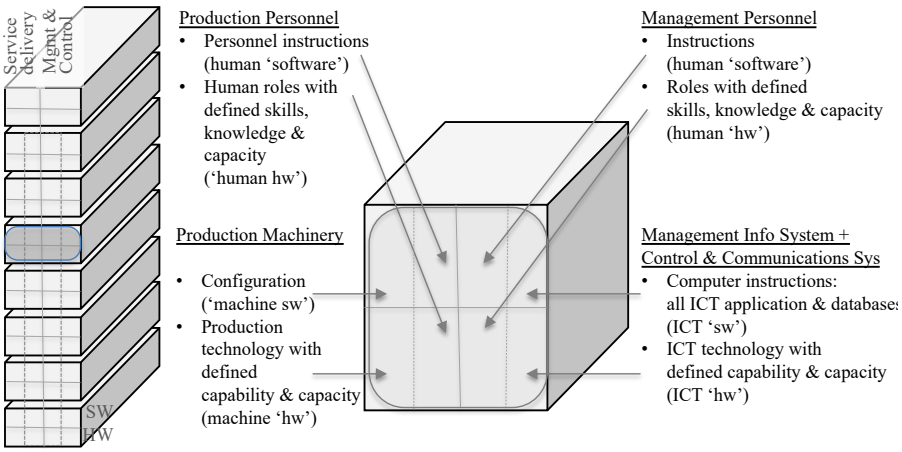


Fig. 2 Orthogonal treatment of 'human or machine' and 'software or hardware' implementation choice preserves scope completeness and leads to complexity reduction: each combination is valid

The above pairwise differentiations in modelling scope can be combined as needed due to their orthogonally to form a large number of model types that *may* be of interest to answer relevant stakeholder concerns about a given enterprise entity.

Thus, instead of creating and prescribing a very long list of model types for every usual type of enterprise entity and typical problem area, and codifying them in the AF, GERAM and its modelling framework GERA allows its user to independently contemplate, and if necessary specify, the problem at hand (as a set of stakeholder concerns), and decide what model types are needed for expressing a solution (that can be represented in various views for the stakeholders – corresponding to their viewpoints from which they consider the entity of interest).

GERA is like a 'bookshelf', with labels similar to the 'Dewey system' used in libraries: 'like' models go on the same shelf. However, even if we pinpoint a given model-type, many kinds of models can still be produced. For example, given a factory workshop as our target enterprise entity (or 'system of interest'), there exist many options to describe the functional requirements of the manufacturing equipment (machine tool hardware). The model may take the simple form of a list of functions, or a set of static process models, or a set of dynamic process models, stochastic or

deterministic models, either ignoring or representing cost, etc. The nature of the necessary model is determined by the task at hand, who needs to peruse the model and for what purpose; models are created in response to concerns of stakeholders (as per ISO 42010 [24]).

GERAM's orthogonality of modelling scope definitions may be applied in the development of architecture metamodels, to keep them as simple as possible, or possibly discover hitherto hidden aspects of modelling. For example this was relevant in the efforts [28][29] to introduce human view elements into the NATO AF (currently version 4).

Many AFs come with an 'associated metamodel', which essentially defines the kinds of facts and relationships that the AF is proposing to cover. However, these 'associated metamodels' are actually generalizations and not a correct view of the roles of metamodels. Given an enterprise entity (a sociotechnical system of systems), the same system may undergo various kinds of change, which in turn determines the stakeholders of that change, and in turn this can determine what is a suitable methodology to address the change. It is then the methodology that dictates (or advises) the kinds of models that may be necessary to build, and this determines the necessary metamodel for the modelling scope at hand. During the life of an enterprise entity multiple methodologies may be used, supported by various metamodels and modelling tools that help the user to create, represent, analyse and share these models.

The upshot of the above consideration is that there is no single enterprise architecting methodology and also there is no single 'optimal', or 'complete' metamodel for enterprise architects. GERAM as an AF defines the fact that in any particular enterprise change case there is an enterprise architecting methodology, which can define the models necessary and codify these in terms of a metamodel, or in an optimal case an ontological theory.

The relevance of this aspect for the EA community is that one needs to develop various methodologies, each for typical types of enterprise change. At the same time enterprise modelling tool vendors need to (in the long run) allow their tools to be extended for various modelling purposes. The enterprise modelling tool landscape is thus predetermined to move into the platform-as-a-product space, where the tool allows the addition, adaptation, extension to the platform (signs of this happening are already present in the current tool vendor arena, but this article is not meant to make comparisons between tools, therefore specific references are omitted).

In addition to the modelling scope, note that an enterprise model's pragmatic information content (i.e., its interpretation by the stakeholders) is also dependent on the model's status [30] in the context of use. This is because one important role of enterprise models is that they are used for communication and agreement among stakeholders (this even includes self-reflection exercised by a single stakeholder). This context of use is best described by the life history of the entity. For example, a model may be interpreted as representing the entity 'as perceived by the analyst', or as the 'agreed view of the AS-IS situation by all stakeholders', or 'as proposed to be', 'as agreed to be', and so on.

The concept of life cycle is fundamental to EA and systems engineering; however, the concept is not well-defined in the majority of EA and systems engineering literature. The following story illustrates this problem.

At a meeting of two Globeman 21 consortium participants at Sanctuary Cove, in 1997, the Japanese and Australian teams were discussing the life cycle of chemical plants, and the life cycle of the project that designs and builds these plants. Each team had its own life cycle representation. The Australian team used the GERAM life cycle model, while the Japanese team used a representation familiar to the project managers in their engineering firm. It soon became apparent that the two teams did not agree. The advocates of the GERAM representation argued that life cycle does not (and should not) represent time, while the Japanese team's model was clearly temporal. During this crucial meeting it became clear that using the same name for such dissimilar concepts is untenable, and the term 'life history' was proposed to refer to how the life of the respective entities unfolds in time.

Shortly after the above events, the IFIP IFAC TF met, and a participant proposed at the end of the meeting that the GERAM life cycle was to be extended with a 're-engineering phase'. However, when at the next TF meeting the life history concept was introduced, it became evident that 're-engineering' is a mere repetition of some already defined life cycle activities: the proposal was dropped, and the life history concept was adopted to be incorporated in the specification of GERAM.

The ISO 15704 / GERAM concepts of life cycle and life history deserve discussion, because no other EA framework has made a differentiation between these concepts, with resulting confusion by architecture practitioners. Even ISO 15288:2015 ('Systems and software engineering – System life cycle processes') makes no such differentiation [12].

The GERAM life cycle consists of life cycle *activity types* (or 'phases'), each of which consider the enterprise entity on a different extent of abstraction. These 'phases' are not temporal in nature – if they were, then the term 'cycle' would not be appropriate.

There is a two-way information flow among such life cycle activities: the less the extent of abstraction of the phase, the more concrete detail of the entity is known to the corresponding activity. For example, considering the entity at the 'detailed design' extent of abstraction needs more information than considering the same entity at the 'requirements' extent of abstraction.

The architecture of enterprise entities is usually actively changing: there are elements of the architecture that remain constant over a longer period of time, but many parts are constructed on the fly by configuring short term structures (committees, task forces, projects, etc), and this configuration activity, at least conceptually speaking, also consists of life cycle activities.

Such self configuration is not necessary for enterprises that have a very stable set of functions to perform, whereupon the structure of resources as built can directly perform the production functions, with perhaps parametric changes.

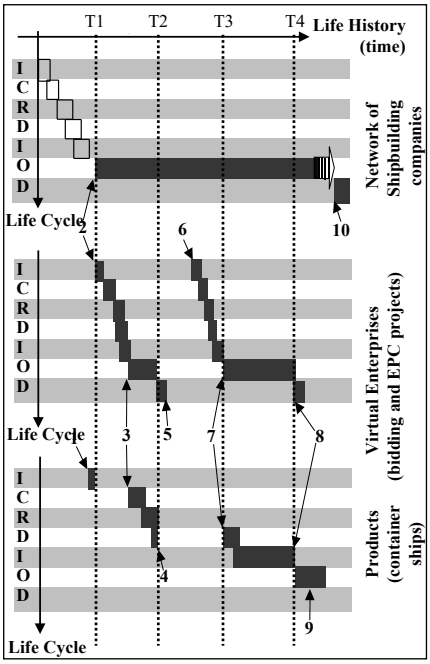
However, an enterprise may have a less pre-determined structure, and part of the task of operational management is to adaptively configure the dynamic structure (by allocation of resources and establishment of links) necessary to perform the production function.

As opposed to the life cycle, the *life history* of an entity describes how changes to the entity happen in time. The life history of an entity consists of life cycle activity instances, which in turn may form sequences of events (with some parallelism). This timeline of the evolution of an entity may be subdivided into life history stages, and milestones separating such stages can also be defined.

Finally, an important consequence of the concept of life cycle being atemporal is that the GERAM life-cycle concept can be used in two ways: (1) create models to use them in deliberate enterprise engineering activities (where models on various life cycle phases serve for expressing designs and to direct change), and (2) create models on any appropriate life cycle phase, to assess the current situation, or to make sense of *emergent* change, i.e. use models *a posteriori*, to investigate, analyse, understand change that already happened, and use this for decision support.

Even in deliberate, directed change, as-is models are usually derived by first creating detailed design phase models, then creating abstractions of these models at the preliminary design phase, or further, at the requirements phase, or even further, at the concept phase, to elucidate the principles and the explicit business model behind the concept of the current state of the enterprise.

This method of analysis can uncover a model of the current enterprise that represents it on the 'right extent of abstraction' suitable for a strategic change to be contemplated. There is a temptation to simply state that the right extent of abstraction is the extent of abstraction on which stakeholder concerns are expressed. However, this begs the question: are stakeholders expressing their concerns at the right extent of abstraction?



(1) A customer identifies a need for a product. **(2)** At time T1 the customer contacts the *network broker* and initial negotiations are made. The *broker* informs the *competence manager* about the need. The *competence manager* sets up a **configuration virtual enterprise (VE)** (for bidding purposes). This takes place in the operational phase of the network. An *information and communications manager* is responsible for the set up of the communication system. *Temporary contractors* might be included. The *contract manager* becomes leader of this VE. **(3)** In the operational phase of the **configuration-VE** the *contract manager* contacts the customer and the *VE-partners* transform the need into a product concept, requirements and a conceptual design. **(4)** This forms the basis for a quotation at time T2. **(5)** The **configuration-VE** has completed its tasks and is decommissioned. **(6)** The customer accepts the quotation, and the *competence manager* sets up a **construction-VE** (for engineering procurement and construction (EPC)). The *contract manager* leads the **construction-VE**. The *inf. and comm. manager* is responsible for the set up of the communication system. *Temporary contractors* might be included. **(7)** In the operational phase of the **construction-VE**, the product design is finalised and the product is produced, i.e. implemented and deployed. **(8)** The product is finished at time T4 and delivered to the *customer*, and the **construction-VE** is closed down (decommissioned). **(9)** After delivery, the customer uses the product, which corresponds to the operational phase of the product life cycle. (A subsequent VE might be responsible for maintenance and service, not shown). (10) For the sake of completeness, the decommissioning of the network is also shown.

Fig. 3 Parallel life histories of systems in a system of systems: each enterprise entity has its own life cycle and life history. (Example of shipbuilding network in the Baltics developed in the frameworks of the Globeman 21 Consortium [31].)

The resolution to this dilemma is that in this sense-making story, the *analysis outcomes* and *stakeholder understanding* of what are the real concerns co-evolve. It is the

interaction between the analysis process and the sense-making process through which stakeholders iterate and refine their understanding of the real problems, and that simultaneously arrives at the right extent of abstraction and stakeholder concern formulation.

The difference between these two kinds of change processes (deliberate and emerging) can be visualised as different patterns of life cycle activity instances in the life history of a system. Thus, the life cycle / life history dichotomy allows both researchers and practitioners to discuss deliberate and emergent / organic change within the same framework.

The life cycle of an enterprise entity in GERAM is recursive: when an entity operates, as part of this operation it may perform life cycle activities of other entities (or of itself). An example of this is illustrated in Fig. 3 through a real life example as applied by the Globeman 21 consortium for the creation and operation of a network of shipbuilding companies.

The system of systems concept familiar from systems engineering is naturally represented in GERAM: the system as a whole has its own life history, while constituent systems have their own life history. At any one stage of life of a system of systems, constituent systems appear, evolve, change, and may disappear. Typical such limited-time in existence entities are change programmes and change projects created by them, and other supporting systems may also be ephemeral.

From the methodological perspective, this is an important way of seeing change. By representing the parallel life histories of constituent systems in a particular change of a particular system of systems, one can identify all life cycle activities of all changing constituent systems, resulting in a guide to create a complete list of tasks necessary to orchestrate the change. This is the basis of being able to create a concrete, case-specific methodology and transformation plan [32]. Of course, for efficiency reasons there must exist a library of methodology components (techniques, metamodels, etc) for each typical life cycle activity type identified.

2.2 The contribution of GERAM to the EA community

Given that GERAM is a sense-making instrument rather than a prescriptive framework, it can be used by those who work on the development of their own respective architecture frameworks. ISO 15704 was developed to list requirements that AFs need to satisfy and not be prescriptive about how to achieve this. The logic behind this approach was that the role of the standard was to facilitate development, rather than being an attempt to override the work done by significant groups in the EA community. Saha [33](p16) writes: "*GERAM/ ISO 15704 is an excellent baseline to map and assess candidate architecture frameworks...*" and refers to GERAM as a meta-framework (a language to talk about frameworks), or 'framework of frameworks'.

A number of EA frameworks were analysed and mapped against the ISO 15704 requirements [34][35]. The conclusion is that, what is often published under the name 'Architecture Framework', is in fact a collection of artefacts that includes an 'AF proper' (as defined in Section 1.4) plus a set of artefacts described by that AF. For example, an AF proper may define the concept 'EA methodology', but any such methodology is not part of the AF proper. For a defined type of stakeholder community and typical application, AFs may come bundled with a collection of artefacts, such as

architecture description languages (modelling languages), an architecting methodology, a set of reference models, or even a suggested modelling tool. A taxonomy could be built for of any of these; for example, a factory design methodology may have specialised versions for typical factory types.

Such mappings of AFs against ISO 15704/GERAM requirements have had influence, for example on the release of new versions of DoDAF [36](p3) and [37]) aligning DoDAF terminology to the ISO 15704 and ISO 19439 standards. We believe that similar new applications for AF development could for example contribute to the popular TOGAF framework's newer versions as well, allowing change programmes, and projects to become first class citizens (enterprise entities) in that framework, similar to the way illustrated in Fig. 1.

The content of these artefacts (methodologies, principles, reference models, etc.) is normally not developed by the EA discipline. Thus, a methodology to develop the business model of an enterprise is likely to come from management science, reference models for projects and programmes from the project management discipline, process reference models for IT management from the IS community, and so on.

Prescribing a methodology even for a typical EA problem can quickly become complicated, as the author must prepare advice for all types of situations, contexts, and user skill levels. An alternative is to prescribe a meta-methodology, which is a more economical approach (see Section 3.1).

In summary, the EA discipline is (or should be) an interdisciplinary synthesis of the contributions of underlying disciplines. The EA terminology and a theory of EA must be able to describe the interactions among theories, methods, models, and tools developed by these disciplines and how new insights and practices can emerge through such interaction. Thus, EA can be seen as the systems science of change.

For EA researchers, GERAM can also be used as a research framework, with the view of circumscribing the scope of research and formalising the research problem at hand – in other words, to translate the initial informal research question into a formal one [37][38][39][40].

2.3 Manufacturing and Engineering Applications

The first industrial application of GERAM was in the Globeman 21 Consortium of approximately 40 industry partners that demonstrated the use of EA principles in a number of applications in the manufacturing industry. The final report of the consortium [41] outlined the most important outcomes as follows:

- The Virtual Enterprise Workbench for World-wide Integration & Development (VIEWBID), providing a company with an integrated set of tools and methods for co-ordinating its global business process for bidding / tender preparation;
- Virtual and Real Information Technologies driven Global Engineering/ Enterprise (VRIDGE) - a virtual enterprise (VE) that carries out the design, procurement, construction, and manufacturing of a chemical plant. This VE facilitated the understanding of global business processes and the investigation of requirements for global product information access and control, as well as of IT infrastructure for VEs.

- GlobOS, that demonstrated a concept of networking among shipyards where pre-qualified partners in the network deliver different generic steel structure assemblies for the final ship. GlobOS illustrates how facilities, methods and assemblies can be modelled and shared within a network.

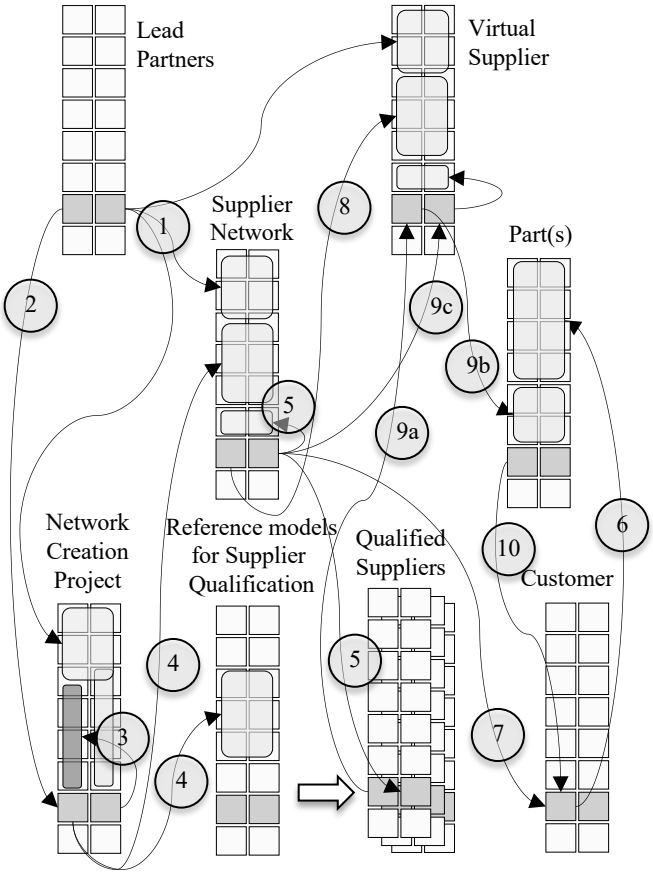


Fig. 4 A Dynamic Business Model of the creation of an Enterprise Network and its Virtual Enterprises (VEs)

The subsequent Globemen consortium of 16 industry and 3 academic partners focused on inter-enterprise collaboration based using global information exchange and control [13][42]. Globemen demonstrated the applicability of the GERAM framework (specialised by the consortium under the name VERAM) to the creation of global enterprise networks that are capable, on demand, of creating *service* virtual enterprises [31][38] that provide after-sales service (troubleshooting, turnaround, factory optimisation, etc.) in a transparent way to chemical factories, although each service typically requires the competencies of multiple globally distributed companies.
GERAM has also been applied to support the development of Small and Medium Enterprises (IMMPAC) [43], to design and create the concept of Virtual Industry Cluster [44], and technological platforms such as PyME CREATIVA [45][46][47], and

to build an EI engineering reference framework and toolbox to improve the design and operations of manufacturing companies [48].

GERAM was the methodological basis for the development of a reference model for Integrated Product and Process development (IPPMD) and which was subsequently applied in multiple industry cases [49]. Three case studies have been published; the first is related to a product transfer for an aeronautical firm to a Mexican manufacturing company. The second case study was implemented in a manufacturing firm to automate a planning process for design and manufacture of an automotive part.

Another case (illustrated in Fig. 4) describes how a Small or Medium Enterprise (SME) created its development process for a new product. Thus, IECOS is a Virtual Supplier integrating qualified suppliers into a network, that according to their website is "an engineering and manufacturing firm that focuses on developing high added value solutions for […] clients and allies through three different business units: IECOS Supply-, Engineering- and Technology Services". 'IECOS Supply' integrates the capabilities of more than 20 small and medium-sized metal-mechanical and plastics manufacturing companies [50].

Finally GERAM has been used as the theoretical basis to define a reference model to create a sensing, smart and sustainable manufacturing enterprise [51][52].

2.4 Applications to the Standards Community, Environmental and Emergency Management, and Collaborative Healthcare

The wide applicability of GERAM as an overarching, non-prescriptive ('light-weight', see Section 3) AF has allowed its involvement in conjunction with systems thinking and system-of-systems paradigms to address several major challenges faced by society today.

For example, in the standards community, currently there is limited, mainly ad-hoc collaboration between custodian work groups within each technical and scientific committee, causing low awareness of other groups' work and resulting in gaps and overlaps in standards' scope and inconsistencies in standards' glossaries. This makes standards difficult to use together – which is typically expected for any sizeable project. GERAM has been proposed as an essential tool in discovering gaps and overlaps in the areas of relevance and building an intelligent repository underlying an expert system that allows achieving an improved and sustainable interoperability of standards [53].

Environmental management (EM), an increasingly essential aspect of any enterprise or network, is often not properly integrated in the business, resulting in sub-optimal environmental performance and merely minimal compliance with standards. GERAM was used to support the discovery of the areas and extent of EM integration and to ensure that environmental information of adequate quality and detail is promptly delivered to the management to properly support the long and short-term decisional processes [54].

Emergency management is another important field of application in the context of climate change and increasing rate of natural and man-made disasters. Here, GERAM has been used to provide a checklist of participants and aspects required in modelling the present (AS-IS) and desired future (TO-BE) states of emergency management services [55]. This was done with a view to improve the effectiveness and resilience of

the command and control and response teams but also to achieve suitable preparedness of the disaster response and management effort participants.

In addition, Noran and Panetto [56] proposed the use of GERAM and systems thinking principles in the lifecycle-centric analysis and design of improvements to healthcare interoperability and collaboration, to meet long term challenges (aging population, patient safety, complexity of services) and improve response to short term, acute challenges (large scale medical emergencies such as pandemics).

Further applications of GERAM continue to emerge, such as enterprise sustainability [57][58], and in preparedness building for Mergers and Acquisitions [59].

3. 'Heavy Weight' vs. 'Light Weight' Architecture Frameworks

3.1 The complexity conundrum

Readers following the development of AFs presented above would have noticed the following contradiction:

i) AFs are intended to help simplify the complex and thus sometimes unpredictable and sensitive enterprise change processes ('change' may mean incremental or major change, or the engineering and construction of a green field installation);

ii) Change, including deliberate (engineered and managed), has multiple aspects and numerous stakeholders, with many concerns and relationships; an AF must address all of these to achieve the desired coherency of decision-making and action.

The conundrum lies in the fact that according to the above logic, the 'AF that can address everything' would likely be complex, thus creating unpredictability, whose avoidance was the primary objective in the first place [58][60]).

If the AF is too complex for the stakeholders to fully understand, then how can we use it to solve the problem, when it creates another problem of similar nature? The question is: is it possible to have a simple AF as a basis for shared understanding and the development of an approach that can be used to adequately manage the complexity of change?

The objective of leading and managing change cannot be deterministic control, because enterprises operate in an environment capable of producing chaotic, non-deterministic behaviour. An EA methodology should help management navigate this partially determined, partially controllable world, steering change into a future where some important characteristics will hold, without having to predict the exact future trajectory and exact future state. With a vision of the future, management must *channel* change in a desirable direction, and an EA methodology should help achieve this. Part of this channelling is the establishment of change governance (with principles and processes that work in parallel with the change) without *a priori* prescribed sequence of actions.

When members of the Globemen Consortium requested a step-by-step methodology for developing an enterprise network that creates 'Service Virtual Enterprises', they realised that a pragmatically useful generic methodology is too hard to produce, due to the complexity of preparing for all possible cases. They saw, however, that for a fixed set of involved entities (e.g. a leading engineering firm, some representative factory

owners, OEM vendors, and typical local service providers) whose life cycle relationships were determined (i.e., which entity should contribute to which life cycle activity of the another), a relatively simple, customised methodology could be defined [32][61].

Thus, the answer to the dilemma seems to be: rather than bundling prescribed generic methodologies with AFs, use a simple framework and a 'meta-methodology' to adaptively create up-to-date change processes, combining proven methods, techniques and tools of underlying disciplines as necessary.

Some frameworks have a built-in methodology (e.g. TOGAF ADM), but methodology should be tailored to the specific case. Thus, the ADM is a repository of techniques and methods, and a meta-methodology should be used to identify the parts needed for a particular case.

The elements of the simple framework may be described using a (meta) meta-model [62], but care must be taken not to confuse (or mix) with meta-models expressing the semantics of enterprise modelling languages.

According to the same argument we must avoid creating a 'heavy weight' AF by prescribing model deliverables. However, for typical problems one can define a methodology, with associated models and languages.

3.2 Systems science and systems thinking in EA

It is worth illustrating how systems thinking can be applied to locate the need for change and establish the structure of a model that is underlying a meta-methodology.

A typical case is continuous improvement, whereupon someone in the operations notices a problem (on the shop floor or in management) and is searching for a solution by analysing the observed problem and its causes. One can use a systems thinking diagram, or one of the many types of root cause analysis tools, for this to identify the entities involved, those affected, and the location of needed change.

For example, Fig. 5 shows a systems thinking diagram that a lead partner may have used to represent the problem of decreasing orders. The systems thinking diagram identifies the root cause: the SME's limited capability set to fulfil complex orders.

The solution is to create a virtual entity that looks like a medium sized supplier, able to perform complex operations. One way to achieve this is to create a network, in which processes, information flow, planning and scheduling, performance and resource management are coordinated. This needs tactical and possibly strategic information sharing beyond operational interactions, which is only viable if partners are pre-qualified and trusted and they subscribe to the network's operating policies and principles (finances, contracts, technological compatibility, quality and process).

This calls for a tightly regulated network, which in turn can create the desired virtual manufacturing company that looks like a single, well-managed medium sized supplier. This thinking is useful to understand the need for transformation (change, or innovation); subsequently, a *dynamic* business model can be used to represent transformation from the present (AS-IS) to the chosen future (TO-BE).

The change process can be modelled as a 'dynamic business model', a special type of systems thinking diagram of action and outcome used to consider what *may* have to be performed as part of a change. In this model, each entity is represented by its life cycle,

and each arrow connects the entity that performs some change activity to the life cycle activity(ies) of the entity being changed. Such diagrams have been used in practice to define complex engineering and change processes; it has been found that despite a seemingly complicated structure, the models created have become a collaborative thinking tool towards defining the scope and location of change.

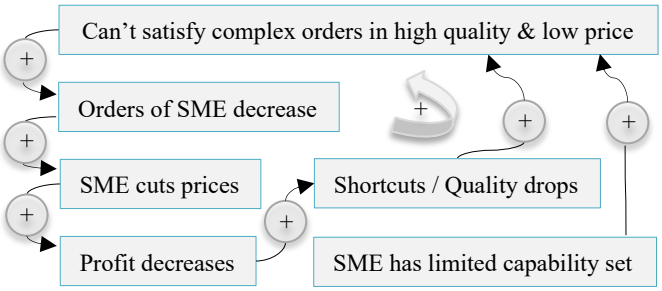


Fig. 5 Systems thinking diagram illustrating why is it hard for SMEs to compete

Fig. 4 previously shown is a model of action and outcome leading to the creation of a Virtual Supplier by a network of qualified suppliers. Arrows may stand for generative relationships or operational interactions. A 'generative' arrow represents a relationship expressing that as an entity operates, it contributes to the life cycle activity of another entity. E.g., in Fig. 4, arrow ① between Lead Partners and Network Creation Project represents the fact that Lead Partners' action is to identify the need for, develop the concept of, and define the mandate (which is part of the requirements) of a network creation project. Arrow ② represents an operational interaction, expressing the fact that Lead Partners contribute to the operation of the Network Creation Project.

The Systems Story that accompanies Fig. 4 unfolds like this: the lead partner(s) identify(ies) the entities to be part of the business model, and the concept and mandate of each, and sets up a Network Creation Project (1). Lead partner(s) contribute to this project (2). Project management designs the details of the project and oversees project execution (3). The tasks are to design the supplier network, and a set of reference models (functions, processes, standards, policies and principles etc.) that a qualified member of the network will have to follow (4). Network management builds the network from qualified partners (5).

When a potential customer designs a part (6), leaving some or all detailed design decisions to the supplier, it contacts the network for a quote (7). If successful, the Network Office creates a 'Virtual Supplier' (8), assembled from qualified suppliers compliant with the reference models (4). The selected suppliers contribute to the Virtual Supplier's processes (9a) to perform the detailed design, production and delivery, so as the parts eventually reach the Customer (10). Note that the VE does have its own management function, but its operations are also overseen by the Network Office (9c).

4. Conclusion

4.1 Why is it necessary to work on the future of EA?

Today's enterprises struggle with the number of fragmented models, tools and methods proposed to them by multiple disciplines, and the resulting adoption is less than coherent [27]. We believe that the future of EA is in its ability to develop an interdisciplinary language and theory enabling a concerted and synergistic application of contributions of underlying disciplines (results from management science, systems, industrial, manufacturing and software engineering, IS, Artificial Intelligence, and so on).

GERAM is a valuable *baseline meta-framework* - to discuss the above, to create new theories, schools of thoughts, integration of engineering practices and tools, explanations of how technologies can be part of EA, methodologies for EA implementation, and Partial and Particular models for re-use.

4.2 The evolution of the Science of EA

EA is an evolving discipline, including and evolving Theory, Framework, Standards, Methodologies, Tools and Models. First and foremost the EA community needs to *continuously improve and enrich the theory of EA* for it to remain relevant and useful. As expressed in Section 2.2, *EA should be the systems science of change*. As a consequence, the EA community should:

- Develop a theory and harmonise the contributions of the above disciplines, so that all stakeholders (not only architects) of enterprise change and evolution can use EA as a common language [60].
- Develop recommendations of who and how should *learn this harmonised language*, contribute to *harmonise EA-related international and industry standards*, and continue interpretation and *mapping* of particular frameworks as they develop, in keeping with the common language,
- Extend the theory to cover the *interplay* of *evolution* and *enterprise engineering*, where complexity management plays a crucial role, and extend the scope to include social and ecological systems.

4.3 Emerging Technologies and EA

- Keep demonstrating the usefulness of the EA language and theory by developing commonly understandable *interpretations of existing and new technologies*, as a tool for industry analysts to moderate the 'hype cycle' of new technologies. E.g., one could start with explaining the role in EA in 'big data analytics', 'cloud computing', 'sensing enterprise', 'machine learning', etc.
- *Re-interpret* notable and important *architectural styles*, *schools of thought*, and *reference models* using EA's common language, to make these accessible, comparable and synergistic. This is a very large agenda, which needs the help of

underlying disciplines. However, the outcome can be a 'LEGO' of components (models, tools, languages, methodologies) enabling innovation.

- Notable reference models / Reference Architectures exist for service delivery, management & control, or both. The following is an example list of well known ones (with bibliographic references omitted): Service orientation on the IT level and on the Business level, Fractal organisations, Model Driven Architectures to instantiate process oriented organizations, governance models (of IT and of EA), process improvement (Kaizen, CMMI, SPICE, Six Sigma), Reference Architectures for Programme and Project management, etc.

4.4 Applications, Enterprise Integration Engineering and EA

- Continue research on Architecture principles, and conduct and publish architecture evaluations (*cf* ISO 42030), and comparative studies to demonstrate value.
- Publish case studies and solutions to contribute to knowledge of the EA practice, e.g. challenges to introduce EA practice (soft factors, human interactions, change management, interoperability, etc.)
- Enrich the relevance of the discipline by publishing case studies of EA in government defence, petroleum, cement, retail, global companies, , health care, etc.
- Develop problem-specific methodologies, arising from *scale* (micro-enterprises, SMEs, global companies) and *type of change* (for implementing new business models, work practices, automation, M&As, new management style, reorganisation, productivity improvement), so as to reflect a relevant sample of typical stakeholder concerns and disseminate solutions to address them.
- There has been decades of debate about 'integration' *vs* 'interoperability' in the domain of Enterprise Architecture. It appears that the debate has never been resolved [3][63][64] because the interoperability community never seems to have acknowledged that the goal is still *integration*, not interoperability per se: the latter is only a tool to create, maintain – and if needed re-create in a dynamic fashion – an integrated material and information flow in the enterprise.

4.5 EA Community Building

The community is currently fragmented by *industry* (IT/Systems Engineering, Manufacturing, Public Sector, Defence, Service industries), *nature of enquiry* (scientific / applied), and by *schools of thought*. Communities include academia (IFAC, IFIP, IEEE), Research Groups, Industry Associations & standards bodies (Open Group, Object Management Group, INCOSE, ISO, NIST, DIN), consortia, consulting firms, CIO councils, etc.

These communities have their own conferences or other fora, but there is no forum to bring all of these together. We believe that a revival of the ICEIMT-style conference and workshop series could play a role in creating coherence among the existing 'EA silos'; perhaps EA practice could be used to provide an answer.

Acknowledgements

The authors acknowledge the contributions of the members of IFIP Working Group 5.12 to the development of ISO 15704/GERAM, and especially the tireless efforts of Richard Martin to update GERAM (now 'GERAM-ISO') as part of ISO 15704:2019.

Dedication

This article is dedicated to the memory of Jim Nevins and Ted J. Williams.

References

1. Bernus P, Noran O, Molina A.: Twenty Years of the GERAM Framework. In Boje, E. Xia, X. (Eds). Proc 19th IFAC World Congress. Cape Town. *IFAC-Papers Online*. pp3300-3308. (2014).
2. Bernus P, Noran O, Molina A.: Enterprise architecture: Twenty years of the GERAM framework. *Annual Reviews in Control*. (39):83-93. (2015).
3. Chen, D.: Framework for Enterprise Interoperability. In Bernard Archimède and Bruno Vallespir (Eds.) Enterprise Interoperability. Wiley. pp1-18. (2017).
4. Williams, T.J., Bernus, P., Brosvic, J., Chen, D., Doumeingts, G., Nemes, L., Nevins, J.L., Vallespir, B., Vlietstra, J., Zoetekouw, D.: Architectures for Integrating Manufacturing Activities and Enterprises. *Computers in Industry. Special Issue on CIM*. 24(2-3):111-140. (1994).
5. Bernus, P., Nemes, L., Williams, T.J. (Eds): Architectures for Enterprise Integration. London : Chapman & Hall, (1996).
6. IEC 62264: Enterprise-control system integration. Geneva : ISO. (2013).
7. Kosanke, K., Vernadat, F., Zelm, M. CIMOSA: Enterprise engineering and integration. *Computers in Industry*, 40(2-3):83-97. (1999).
8. Doumeingts, G., Vallespir, B., Chen, D.: GRAI Grid Decisional Modelling. In P. Bernus, K. Mertins and G. Schmidt (Eds.) Handbook on Architectures of Information Systems. Berlin : Springer. pp313-339. (1998).
9. Williams, T.J.: The Purdue enterprise reference architecture. *Computers in Industry*. 24 (2):141-158. (1994).
10. Bernus, P., Nemes, L.: A Framework to Define a Generic Enterprise Reference Architecture and Methodology. In Proc. ICARV'96 Vol3/3. Nanyang Technical University. pp88-92, (1994).
11. Bernus, P., Nemes, L.: in *Computer Integrated Manufacturing Systems*. 9(3):179-191. (Reprint of Bernus & Nemes (1994)), (1996).
12. ISO 15288: Systems and software engineering – System life cycle processes. Geneva : ISO, (2015).
13. Bernus, P., Baltrusch, R., Tølle, M., Vesterager, J.: Better Models for Agile Virtual Enterprises – the Enterprise and its Constituents as Hybrid Agents . In I.Karvoinen et al (Eds). Global Engineering and Manufacturing in Enterprise

Networks (Globemen), VTT Symposium Series 224, (9-10 OCT 2002). Helsinki : VTT. pp 91-103. (2002).
14. ISO 15704: Industrial automation systems – Requirements for enterprise reference architectures and methodologies. Geneva : ISO, (2000).
15. IFIP/IFAC Taskforce: The Generalised Enterprise Reference Architecture and Methodology. in Handbook on Enterprise Architecture. Berlin : Springer. pp22-64, (2003).
16. ISO 19439: Enterprise integration – Framework for enterprise modelling. Geneva : ISO, (2006).
17. ISO 15704: Industrial automation systems – Requirements for enterprise-referencing architectures and methodologies. Geneva : ISO. (2019).
18. Zachman, J.: A Framework for Information Systems Architecture. *IBM Systems Journal*, 26(3):276-292. (1987).
19. FEAF (Federal Chief Information Officers Council): Federal Enterprise Architecture Framework. Version 1.1. (1999).
20. DoD: DoD Architecture Framework Version 2.02. US Department of Defense. (2010).
21. OMG: Unified Architecture Framework 1.1. Object Management Group. (2020).
22. The Open Group: The Open Group Architecture Framework. v8.1.1; v9.1. The Open Group. (2006, 2011, 2018).
23. The American Heritage Dictionary of the English Language (4th Edition): Boston : Houghton Mifflin. (2000, Updated 2009).
24. ISO/IEC/IEEE 42010: Systems and software engineering – Architecture description. Geneva : ISO. (2011).
25. ISO/IEC/IEEE 42020: Systems and software engineering – Architecture processes. Geneva : ISO. (2019).
26. ISO/IEC/IEEE 42030: Systems and software engineering – Architecture evaluation. Geneva : ISO. (2019).
27. Doucet, G., Gøtze, J., Saha, P., Bernard, S: Coherency Management: Using Enterprise Architecture for Alignment, Agility, and Assurance. *Journal of EA* :1-12. (2008).
28. Handley, H.A.H., Smillie, R.J.: Architecture framework human view: the NATO Approach. *Systems Engineering*. 11(2):156-164. (2008).
29. Handley, H.A.H., Smillie, R.J.: Human view dynamics – the NATO approach. *Systems Engineering*. 13(1):72–79. (2010).
30. Hysom, R.: Enterprise Modelling – The Readiness of the Organisation. In Bernus, P., Nemes, L., Schmidt, G. (Eds). Handbook on Enterprise Architecture. Berlin : Springer. pp373-416. (2003).
31. Vesterager, J., Bernus, P., Pedersen, J. D., Tølle, M.: The what and why of a Virtual Enterprise Reference Architecture. in E-work and E-commerce. Novel solutions and practices for a global networked economy. Amsterdam : IOS Press. pp846–852. (2001).
32. Noran, O.: Towards a Meta-methodology for Collaborative Networked Organisations. In L. Camarinha-Matos (Ed.) *IFIP AICT*, Vol.149:71-78. (2004).
33. Saha, P. (Ed): A synergistic assessment of the Federal Enterprise Architecture Framework against GERAM (ISO15704:2000). In Handbook of Enterprise Systems Architecture in Practice. IGI Global. (2006).

34. Noran, O.: Mapping of Individual Architecture Frameworks. In Bernus, P., Nemes, L., Schmidt, G. (Eds). Handbook on Enterprise Architecture. Berlin : Springer. pp65-210. (2003).
35. Saha, P.: Analyzing TOGAF from the GERAM Perspective. http://www.opengroup.org/architecture/wp/saha/TOGAF_GERAM_Mapping.htm, (2004).
36. DoD: DoD Architecture Framework Version 2. US Dept. of Defense. (2009).
37. Chaharsooghi, K., Achachlouei, M.A.: Developing life-cycle phases for the DoDAF using ISO 15704 Annex A (GERAM). *Computers in Industry*. 62(3):253–259. (2011).
38. Vesterager, J., Bernus, P., Larsen, L.B., Pedersen, J.D., Tølle, M.: Use of GERAM as Basis for a Virtual Enterprise Framework Model. In Mo, J., Nemes, L. (Eds) Global Engineering, Manufacturing and Enterprise Networks (DIISM2000) Dordrecht : Kluwer. pp75-82. (2000).
39. Noran, O.: Using Reference Models in Enterprise Architecture. in Fettke, P. and Loos, P. (Eds). Reference Modelling for Business Systems Analysis. Hershey : Idea Group. pp141-165. (2007).
40. Magoulas, Th., Hadzic, A., Saarikko, T., Pessi, K.: Alignment in Enterprise Architecture: investigating the Aspects of Alignment in Architecture Approaches. In Proc 5th European Conf. on Information Management and Eval. Reading : Acad. Publ. Ltd. pp321-331. (2010).
41. Brown, R.H., Syntera, H. (Eds): Globeman 21 (Global Mfg. in the 21st Century) Final Report. http://www.ims.org/wp-content/uploads/2011/11/2.4.10 .2-Final-Report-GLOBEMAN-21.pdf, (1999).
42. Karvonen, I. *et al*. (Eds): Global Engineering and Manufacturing in Enterprise Networks GLOBEMEN. VTT Symposium Vol.224. Helsinki : VTT Technical Research Centre of Finland. http://www.vtt.fi/inf/pdf/ symposiums/2003/S224.pdf (2003).
43. Molina, A., Carrasco, R.: The use of GERAM to support SMEs development in Mexico, in Handbook on Enterprise Architecture. Berlin : Springer. pp757-778. (2003).
44. Molina, A., Flores, M.: A Virtual Enterprise in Mexico: From Concepts to Practice. *Journal of Intelligent and Robotic Systems*. Vol. 26:289-302. (1999).
45. Molina, A., Mejía, R., Galeano, N., Najera, T. & Velandia, M.: The HUB as an enabling IT strategy to achieve Smart Organizations. In Mezgar, I. (Ed.) Integration of ICT in Smart Organizations. Idea Group. pp64-95. (2006).
46. Giraldo, J., Galeano, N., Molina, A.: Virtual Organization Breeding Environment: Experiences from its Implementation in Mexico, in Proc. 8th IFAC Symposium on Cost Oriented Automation Affordable Automation Systems, La Habana, Cuba (CD Memories). (2007).
47. Nogueira, J.M., Romero, D., Espadas, J. &Molina, A.: Leveraging the Zachman Framework Implementation Using Action-Research Methodology – A Case Study: Aligning the Enterprise Architecture and the Business Goals. *Enterprise Information Systems*. 7(1):100-132. (2013).
48. Vallejo C., Romero D., Molina A.: Enterprise Integration Engineering Reference Framework & Toolbox. *Int J. of Production Research*. 50(5):1489-1551. (2011).

49. Pereda, F.J., Molina, A.: Model driven architecture for engineering design and manufacturing. *IFAC-Papers Online*, 6(1):400-407. (2013).
50. Molina, A., Velandia, M., Galeano, N. Virtual Enterprise Brokerage: A Structure Driven Strategy to Achieve Build to Order Supply Chains. *International Journal of Production Research*. 45(7): 3853–3880. (2007).
51. Chavarría-Barrientos, D., Batres, R. Wright, P.K., Molina, A.: A methodology to create a sensing, smart and sustainable manufacturing enterprise. *Int. Journal of Production Research,* 56(1-2): 584-603. (2018).
52. Molina, A., Ponce, P., Miranda, J., Cortés-Serrano, D.: Enabling Systems for Intelligent Manufacturing in Industry 4.0. To be published in March 7, 2021, ISBN 978-3-030-65546-4, Springer. (2021).
53. Noran, O.: Achieving a Sustainable Interoperability of Standards. *Annual Reviews in Control*. 36(2). (2012).
54. Noran, O.: Engineering the Sustainable Business: An Enterprise Architecture Approach. In Doucet, G., Gotze, J., Saha, P. (Eds.), Coherency Management: Architecting the Enterprise for Alignment, Agility, and Assurance. Int. Enterprise Architecture Institute. pp179-210. (2009).
55. Noran, O., Bernus, P.: Effective Disaster Management: An interoperability Perspective. *LNCS* 7046:112-121. (2011).
56. Noran, O., Panetto, H.: Modelling a Sustainable Cooperative Healthcare: An Interoperability-driven Approach *LNCS* 8186:238-249. (2013).
57. Franco dos Reis Alves, D., Campos, R., Bernardi de Souza, F.: GERAM: Building Sustainable Enterprises. 6th IFAC Conference on Management and Control of Production and Logistics (MCPL13), Fortaleza, Brazil. (2013).
58. Kandjani, H., Bernus, P. and Nielsen, S.: Enterprise Architecture Cybernetics and the Edge of Chaos: Sustaining Enterprises as Complex Systems in Complex Business Environments. In Ralph H and Sprague Jr. (Eds). Proc HICSS2013. *IEEE XPlore*. pp3858-3867. (2013).
59. Vaniya, N., Bernus, P., Noran, O.: Examining Potentials of Building M&A Preparedness. in Proc. 15th Int. Conf. on Enterprise Information Systems. Vol 3. Setubal : SCITEPRESS. pp201-212. (2013).
60. Kandjani, H., Bernus, P.: The Enterprise Architecture Body of Knowledge as an Evolving Discipline In Maciaszek, L.A., Cuzzocrea, A. and Cordeiro, J. (Eds.) LNBIP 141, Berlin : Springer. pp452-471. (2013).
61. Bernus, P., Noran, O., Riedlinger, J.: Using the Globemen Reference Model for Virtual Enterprise Design in After Sales Service. In Karvoinen, I. *et al*. (Eds). Global Engng and Mfg in Enterprise Networks (Globemen), *Symposium Series 224*. Helsinki : VTT. :71-90. (2002).
62. Bernus, P., Noran, O.: A Metamodel for Enterprise Architecture. in *IFIP AICT, 326*. Berlin : Springer :56-65. (2010).
63. Panetto, H., Whitman, L.: Knowledge Engineering for Enterprise Integration, Interoperability and Networking: Theory and Applications . *Data & Knowledge Engineering*. (105):1-4. (2016).
64. Weichhart, G., Stary, C., Vernadat, F.: Enterprise modelling for interoperable and knowledge-based enterprises. International Journal of Production Research 56 (8), 2818-2840. (2018).