Architecture constraints for Interoperability and composability in a smart grid

Conference Paper · August 2010

DOI: 10.1109/PES.2010.5589366 · Source: IEEE Xplore

Andreas Tolk
MITRE

Architecture Constraints for Interoperability and Composability in a Smart Grid

Andreas Tolk

Abstract—**This paper presents the Levels of Conceptual Interoperability Model (LCIM) in the context of supporting composability, interoperability, and integratability of systems and identifies the architecture constraints for enterprise and systems architectures. While enterprise architectures set the conceptual context for the systems, the system architecture defines the implementation details. These research results allow practitioners to identify which metadata are needed and where they can be stored. Current research on composability and research shows that these metadata are necessary, but not sufficient to support interoperability and composability in a smart grid. Some of the related challenges even have been proven to be undecidable.**

Index Terms—**Cooperative Systems, Intelligent Systems, Interconnected Systems, Interoperability**

I. Introduction

THE results presented in this paper are based on the research work on system of systems engineering with focus on integrating model-based solutions into operational systems conducted at Old Dominion University, Norfolk, VA. Although the origins of this work lay in the modeling and simulation (M&S) domain, the results and recommendations are applicable to support smart grid solutions as well, as the challenges regarding the integration of existing legacy solutions that provide a certain functionality that needs to be reused in a different context are similar. The core of the results led to the development of the Levels of Conceptual Interoperability Model (LCIM), which was applied in several domains outside M&S successfully.

The use of architectures to support interoperable solutions is well known to engineers. Enterprise architectures are used to capture the enterprise-wide concepts describing who is doing what, where, and when. In addition, the enterprise architecture shall define why a concept is needed. The implementation of these conceptualizations is done by systems whose architectures map the requirements to implementable solutions (adding the how something is done). However, in practical application, there is often a gap between the conceptual ideas of the enterprise and the implementation details of the systems.

Another challenge is that many systems that should be integrated to enable a smart grid were not defined and designed from the perspective of a common enterprise-wide conceptualization. This raises the question whether legacy systems can be described to ensure their seamless integration into a common conceptualization after the fact.

If several legacy systems provide similar functionality needed for a smart grid solution, we need to be able to identify those systems, select the best system set, compose them into a system of systems, and orchestrate their execution. To allow these, architectural descriptions are needed that address the conceptual side as well as the implementation side. Such descriptions are normally referred to as metadata. In the context of this paper the question needs to be answered which metadata are necessary to support the identification, selection, composition, and orchestration of legacy solution to provide the functionality required for an enterprise operation.

To address these questions, the paper introduces some basic definitions and the LCIM in section two. In section three, the main ideas of enterprise and system architecture regarding interoperability challenges will be discussed. In section four, some tentative ideas on how to apply these results to support interoperability in a smart grid will be discussed.

This paper can only scratch the surface of most problems. The purpose of this paper is to point the reader into hopefully interesting directions and build a discussion foundation to address these challenges more effectively in the future.

II. Levels of Conceptual Interoperability Model

The IEEE definition for interoperability is *"the ability of two or more systems or components to exchange information and to use the information that has been exchanged"* [1]. There are several implications to be derived from this definition for smart grid interoperability requirements.

First, the infrastructure must allow exchanging information and transporting it from senders to receivers, in other words, network connectivity needs to be assured (the ability to exchange bits and bytes).

Second, the implementations of the participating solutions must be able to make sense of the information provided. In other words, common symbols, protocols, and implementation specific interpretations thereof are needed.

An often overlooked or underestimated third aspect is the conceptual alignment of solutions to make sure that the

This work was supported in part by the U.S. Joint Forces Command, the U.S. Army Test and Evaluation Command, and the NATO Research and Technology Organization. The application for smart grid challenges was supported by the Pacific Northwest National Laboratory.

A. Tolk is an Associate Professor for Engineering Management and Systems Engineering at Old Dominion University, 241 Kaufman Hall, Norfolk, VA 23529 USA (e-mail: atolk@odu.edu).

978-1-4244-8357-0/10/$26.00 ©2010 IEEE

implementation specific interpretations in each system are conceptually aligned within the context of the common operation. In other words, is the receiving system really using the information the way it was intended by the sending system? Page et al. [2] recommended the use of three different terms to cope with the resulting challenges: integratability, interoperability, and composability.

Integratability contends with the physical/technical realms of connections between systems, which include hardware and firmware, protocols, networks, etc. *Interoperability* contends with the software and implementation details of interoperations; this includes exchange of data elements via interfaces, the use of middleware, mapping to common information exchange models, etc. *Composability* contends with the alignment of issues on the modeling level. The underlying models are purposeful abstractions of reality used for the conceptualization being implemented by the resulting systems.

It is worth mentioning that these terms can be used in two very different contexts, namely development and reuse. While during development prescriptive norms are needed to ensure that the resulting system will be interoperable, reuse needs descriptive metrics to allow evaluating whether a legacy system can be used in a new operational context or not. Both contexts are connected when legacy systems have to be changed in order to migrate towards new solutions.

In order to address the related issues in a common framework, the LCIM was developed. The LCIM was first presented in [3], but it went through several phases of improvement utilizing the recommendations of colleagues and research partners as documented in [4]. Figure 1 shows the current version of the LCIM.

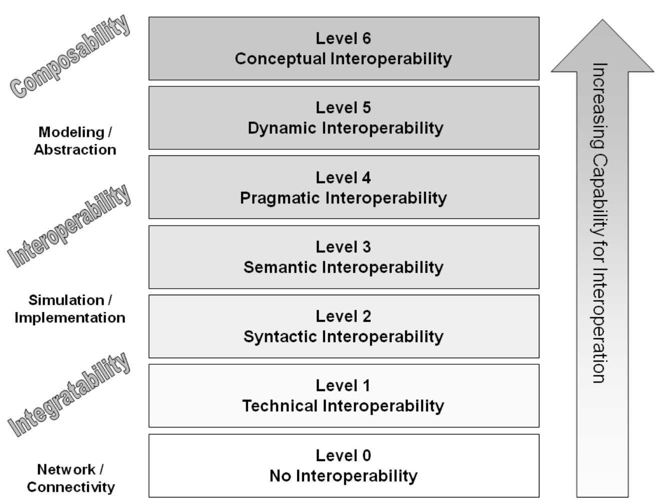

Fig. 1. The Level of Conceptual Interoperability Model (LCIM)

The LCIM exposes six levels of interoperation. The *technical* level deals with infrastructure and network challenges, enabling systems to exchange carriers of information. The *syntactic* level deals with challenges to interpret and structure the information to form symbols within protocols. The *semantic* level provides means to capture a common understanding of the information to be exchanged, often in form of a common data model or object model. The *pragmatic* level recognizes the patterns in which data are organized for the information exchange, which are in particular the inputs and outputs, often in form of messages, replication patterns, etc. The *dynamic* level adds a new quality by taking the response of the system in form of context of the exchanged information into account. The same information sent to different systems can trigger very different responses. It is also possible that the same information sent to the same system at different times can trigger different responses. Finally, assumptions, constraints, and simplifications need to be captured. This happens on the *conceptual* level. In his dissertation, King [5] showed that it is possible to have systems that are perfectly aligned on the first five levels can expose conceptual misalignments. In other words, it is essential to capture assumptions and constraints of solutions to avoid the composition of conceptually misaligned solutions.

Another dissertation research based paper evaluated the LCIM regarding its use describing a state versus prescribing or recommending an integrated solution [6] showing the potential in both categories as well as shortcomings of current standards for interoperability.

In summary, the LCIM allows a structured approach to cope with the challenges that have to be met when evaluating and managing interoperation and composability tasks in complex systems. The ideas were applied and adapted in several domains including the Gridwise Interoperability Framework discussed in [7]. Figure 2 shows the adapted view.

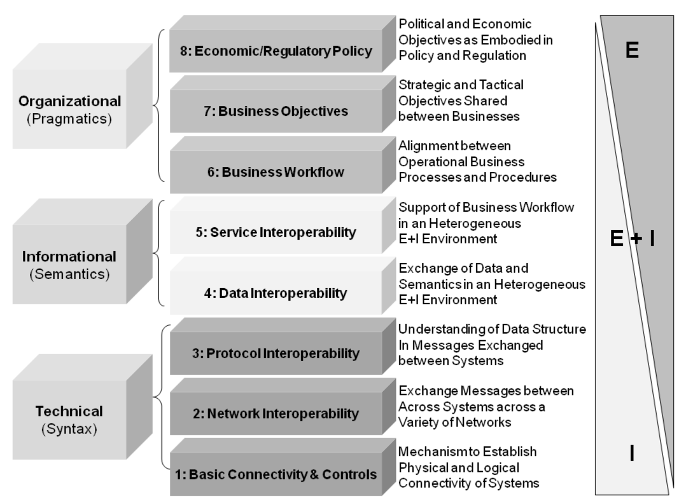

Fig. 2. Interoperability Categories for Smart Grid Solutions

It is worth mentioning that the term pragmatic in the Gridwise model is addressing the organizational use of systems, which addresses the conceptualization of solutions. The LCIM term pragmatic addresses the system specific use of semantically specified data.

Another aspect is addressed by the Energy domain (E) and Information domain (I) graphic on the left. It shows that information technology is interwoven with the organizational aspects of the energy domain on all levels. It is worth mentioning that the same is true for many other domains, like transportation (when including traffic control systems, etc.) and information, the financial system and information, etc. This points towards the necessity of cross-domain governance solutions to protect critical infrastructures, which have been defined as networks for the provision of telecommunication and information services, energy services (electrical power, natural gas, oil and heat), water supply, transportation of people and goods, banking and financial services, government services and emergency services [8]. Protection of these services needs to be supported by interoperable solutions beyond the border of the energy domain.

In the next section, we will discuss how such requirements are currently addressed by architectures.

III. Enterprise Architectures and System Architectures

When discussing architecture constraints it is beneficial to distinguish between enterprise architectures and system architectures. Again, many definitions exist for these terms. IEEE defines an architecture as *"the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution"* [9]. This definition addresses several aspects that are of interest in the context of interoperability and composability. First, it addresses how a system is organized in relation to its environment. Next, the principles governing the design of the system are addressed. The current design goes hand-in-hand with evolution principles. And finally, how the system implements the provided functionality is addressed.

Architectures are therefore needed for the conceptualization part of systems and enterprises as well as the implementation part, latest particularly for systems. The have to provide structured documentation of who is doing what, where, when, why, and how. This information must be provided in machine understandable form to allow support of identifying solutions for potential reuse, selecting the best systems in case of alternative solutions, compose the solutions into a system of systems providing the required functionality, and orchestrate the execution.

A. Enterprise Architectures and the Conceptual View

Whenever a group or collection of organizations have or want to work together to accomplish a common task or to reach a common set of goals, they can be seen as an enterprise. For smart grid solution, such an enterprise comprises government organizations under the Department of Energy, the power providers, the power distribution providers, and many more organizations down to those who collect data on the actual use of power. Each one organization provides services needed to make the smart grid work.

The Federal Enterprise Architecture (FEA) [10] is an example how conceptual constraints and principles governing the design and evolution can be established. FEA defines also several layers allowing using a business-driven approach to identify components of a component-based architecture. These components then need to be implemented as systems.

The FEA Performance Reference Model (PRM) is structured around measurement areas, categories, groupings, and indicators that identify the metrics to be applied to measure success, efficiency, etc.

The FEA Business Reference Model (BRM) provides a framework facilitating a functional view of the federal government's lines of business, the conceptual view of what services are provided.

The FEA Service Component Reference Model (SRM) provides a framework classifying Service Components according to the capabilities they provide to business functions. They become the anchor point to systems.

Finally, the FEA Technical Reference Model (TRM) provides a framework to describe how standards and technologies support the secure delivery, exchange, and construction of Service Components. This is furthermore supported by the FEA Data Reference Model (DRM), a business focused data standardization effort enabling cross-agency information exchanges.

The FEA allows the comparison between very different implementations and contributions. It also addresses how to identify evolution from as-is towards to-be architectures and uses project information to address cross-agency dependencies in developments.

Another enterprise architecture approach of interest is the Zachman Framework [11]. This framework defines a matrix in which the six questions regarding who is doing what, where, when, why, and how are addressed regarding how the describe the business, define relationships, specify components and services as concepts, identify applicable technical solutions as implementations, and select solutions.

While FEA and Zachman are both focusing on human users, the Department of Defense Architecture Framework (DoDAF) Version 2.0 [12] takes machine understandability into account as well, as all information needed to drive the models used in DoDAF to produce desired viewpoints is defined and captured. It should be mentioned that DoDAF is not a pure enterprise architecture, but the latest version support the conceptual viewpoints as well as implementation challenges.

DoDAF uses eight viewpoint categories. The All Viewpoint (AV) models provide information pertinent to the entire architectural description, such as the scope and context of the architectural description. The Capability Viewpoint (CV) captures the enterprise goals associated with the overall vision for executing a specified course of action. The Data and Information Viewpoint (DIV) captures the business information requirements and structural business process rules. The Operational Viewpoint (OV) captures the organizations, tasks, or activities performed, and information that must be exchanged between them to accomplish the desired missions. The Project Viewpoint (PV) captures how programs are grouped in organizational terms as a coherent portfolio of acquisition programs. The Services Viewpoint (SvcV) captures system, service, and interconnection functionality providing for, or supporting, operational activities. The Standards Viewpoint (StdV) is the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements. Systems Viewpoint (SV) captures the information on supporting automated systems, interconnectivity, and other systems functionality in support of operating activities (and will likely be completely replaced by SvcV).

These viewpoints allow describing the concepts, where and how they are currently provided, which projects and portfolios contribute when, and how and when they will likely evolve in machine understandable form. However, as the latest version was only very recently released, how this will work in practice remains to be evaluated. DoDAF should be aligned with the Ministry of Defence Architecture (MODAF) as well as with the the NATO Architecture Framework (NAF).

Another enterprise architecture model effort of interest is the Open Group Architecture Framework (TOGAF) [13]. TOGAF is developed and maintained by The Open Group Architecture Forum and its members. It applies the IEEE definition of an architecture and focuses on the conceptual aspects. To do this, TOGAF supports the following four interdependent architecture types.

The business architecture captures business strategy, governance, organization, and key business processes. The data architecture defines the structure of an organization's logical and physical data assets and data management resources. The application architecture is a blueprint for the individual application systems to be deployed, their interactions, and their relationships to the core business processes of the organization. Finally, the logical software and hardware capabilities that are required to support the deployment of business, data, and application services are summarized in the technology architecture. This includes IT infrastructure, middleware, networks, communications, processing, and standards.

With the exception of DoDAF, the detailed implementation of the systems that provide the functionality is not specified in these architecture frameworks. This needs to be done in system architectures.

B. System Architectures and the Implementation View

While the conceptual view allows describing the big picture the detailed "where" in form of components and "how" in form of functions need to be specified in system architectures. There are various tools and methods available, but they generally agree on the following steps [14].

System architectures are driven by requirements (that are derived from needs and capability gaps in the enterprise). Based on these requirements, a functional and a physical architecture are constructed, identifying functions and components of the system. The allocated architecture combines the two and allows validating and verifying the design and implementation.

The result is a detailed blueprint for the system, depending on the tool used potentially even executable. It specifies how the system will provide the desired functionality under the specified constraints. Even if the system is emerging, such detailed documentation still remains necessary.

In practice, there are three major challenges: (1) Too often the requirements are not derived or traceable to enterprise concepts and constraints but the product of an enterprise independent collection of stakeholder inputs. As a result, the system may fulfill the need of the group of participating stakeholders, but does not necessarily contribute to the overall desired capability portfolio of the enterprise. (2) Once the system is implemented, the system architecture becomes "shelf-ware" and is not kept up-to-date with the evolving system. (3) Environments like the smart grid are evolving constructs with many designers, implementers, regulators, and other actors whose actions are not as well aligned as desired. In addition, the interconnectivity of systems may result in secondary effects of one system in another system.

The result of these challenges is that assumptions and constraints, contributions to general capabilities and other information that is not derivable from the implementation are often lost. If the system needs to be integrated into an alternative context, integration and migration decisions have to be made on the basis of implemented designs and are not supported by conceptual background (which can lead to significant errors, as shown in [5]). Unfortunately, once lost this information cannot be reengineered.

IV. A SYSTEM ENGINEERING PROCESS FOR SMART GRID

From the discussion so far it becomes clear that metadata are needed to support the evaluation if two systems can interoperate in support of a common goal. What is not immediately obvious is that we actually need three sets of metadata: (a) the two sets of metadata describing the systems, hopefully directly derived from the systems' architectures, and (b) the metadata describing the blueprint of the desired operation. The desired operation can either be part of the enterprise architecture or needs to be design based on the operational requirements.

In their work on composability challenges, Page and Opper [15] showed that the reuse of systems in a new context, such as the reuse of existing power related systems in the smart grid context, introduces new complexity to the problem. They extend the work of Overstreet and Nance [16] who described the Condition Specification (CS) formalism that formally and implementation independent captures the idea of conceptual blueprints (condition) and implementation blueprint (specification) as introduced here as well. In his work, Overstreet demonstrates that any CS has an equivalent Turing Machine (TM) specification, which makes sense, as we are talking about information technology support. However, this insight has far reaching consequences, as it enables us to utilize the mathematical proofs for TM challenges in the context of composition of legacy solutions to support a common goal.

Examples of undecidable problems (which really means that they cannot be decided, there exist no algorithm that can help, the best we can do is to look for a good heuristic) are questions like *"Will the system terminate?", "Are two modeled actions order independent or do I have to orchestrate* *them?", "Is the specification complete?", "Is the specification minimal?",* or "*Are two specifications functionally equivalent, in other words, do the delivery the same functionality?"*

In their work, Page and Opper [15] observe that intuitively, component-oriented design offers a reduction in the complexity of system construction by enabling the designer to reuse appropriate components without having to re-invent them, but they show that this assumption is wrong when applied in the context of CS, or – as defined in smart grid solutions – conceptualization and implementation. Although determining if a collection of components satisfies a set of requirements becomes feasible under certain assumptions, we still have to solve a potentially computationally intensive problem. Selecting the right component to fit into an enterprise is a non-trivial task that cannot be generally solved or left to technology.

On the other hand side it is obvious that it is not possible to specify the electricity system as one single system. The enormity and complexity of interfaces and interdependencies would surely lead to a break down of every centrally applied approach. Instead, a self organizing system of systems approach is needed that allows to be driven by rules and laws similar to a natural ecosystem. Such rules can be captured in the enterprise architecture. Nonetheless, in order to build a foundation for such a system of systems, each contributing system itself must unambiguously describe its components and functionality in its system architecture. As it is naïve to assume that every organization within the enterprise will use the same standards to implement its systems, we need a common way to represent the conceptual fundamental aspects as metadata. This allows the mediation of solutions using these common metadata [17]. Composable tasks can now be performed by heterogeneous systems, or, in other words, a common way of describing systems allows mapping them to the appropriate places in the enterprise wide blueprint.

Therefore, without the metadata describing the various levels of interoperation – as identified earlier in this paper as the fundamental aspects of integratability, interoperability and composability –, the tasks of (1) identifying applicable systems to support a desired functionality needed in a smart grid, (2) selecting the best fits (based on metrics agreed to on the enterprise level, one might choose the fewest components, or choose systems that can be connected via standardized interfaces, or choose systems that are inexpensive, etc.), (3) composing them into a coherent solution (which may require to migrate the selected solutions into a smart grid environment), and (4) to orchestrate their execution are not feasible. We need metadata supporting good heuristics.

What metadata is needed can be guided by the LCIM. Where the metadata can be stored can be guided by the results of current enterprise and system architecture research. Unfortunately, based on the results of composability research, these are only necessary, but not sufficient conditions for success. Even worse, based on the fundamental undecidability of respective problems, we can never be sure of success. All we can do is avoid failure and provide for good heuristics.

V. Acknowledgment

The author gratefully acknowledges the contributions the members of his research team for their work on the original version of this document. In particular, Saikou Diallo, Robert King, and Charles Turnitsa shall be mentioned.

VI. References

[1] IEEE. (1990). *A Compilation of IEEE Standard Computer Glossaries.* New York: IEEE Press

[2] Page, E. H., R. Briggs, and J. A. Tufarolo (2004). Toward a Family of Maturity Models for the Simulation Interconnection Problem. In *Proceedings of the Spring Simulation Interoperability Workshop*, IEEE Computer Society, Washington, DC

[3] Tolk, A., and Muguira, J. A. (2003). The Levels of Conceptual Interoperability Model (LCIM). In *Proceedings of the Fall Simulation Interoperability Workshop*, IEEE Computer Society, Washington, DC

[4] Tolk, A. (2006). What Comes After the Semantic Web - PADS Implications for the Dynamic Web. In *Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation,* pp. 55-62, IEEE Computer Society, Washington, DC

[5] King, R. D. (2009). *On the role of assertions for conceptual modeling as enablers of composable simulation solutions*. Ph.D. thesis, Old Dominion University, Norfolk, VA

[6] Wang, W., A. Tolk, and W. Wang (2009). The Levels of Conceptual Interoperability Model – Applying Systems Engineering Principles to M&S. In *Proceedings of the Spring Simulation Multiconference*, IEEE Computer Society, Washington, DC

[7] Ambrosio, R, and S.E. Widergren. (2007) A Framework for Addressing Interoperability Issues. In *Proceedings of the 2007 IEEE PES General Meeting, Tampa*, FL

[8] Gheorghe, A. V., and M. Masera (2006) *Critical infrastructures at risk: securing the European electric power system*. Springer

[9] ANSI/IEEE Std 1471-2000

[10] http://www.whitehouse.gov/omb/e-gov/fea/ (last visited Jan 2010)

[11] http://www.zifa.com/ (last visited Jan 2010)

[12] US Department of Defense (2009) *DoD Architecture Framework Version 2.0*, Washington DC, May 28

[13] http://www.opengroup.org/togaf/ (last visited Jan 2010)

[14] Buede, D. (2009) *The Engineering Design of Systems: Models and Methods*, 2nd Edition, Wiley

[15] Page, E. H. and Opper, J. M. (1999) Observations on the complexity of composable simulation. In *Proceedings of the Winter Simulation Conference*, P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, Eds. WSC '99. ACM, New York, NY, Vol I, pp. 553-560

[16] Overstreet, C.M. and Nance, R.E. (1985) A Specification Language to Assist in Analysis of Discrete Event Simulation Models, *Communications of the ACM,* 28(2):190-201

[17] Tolk, A. (2004) Metamodels and Mappings – Ending the Interoperability War. In *Proceedings of the Fall Simulation Interoperability Workshop*, IEEE Computer Society, Orlando, FL

VII. Biography

Andreas Tolk is an Associate Professor for Engineering Management and Systems Engineering at the Frank Batten College for Engineering and Technology at Old Dominion University in Norfolk, Virginia. He is also affiliated as a Senior Research Scientist with the National Center for System of Systems Engineering (NCSOSE) and the Virginia Modeling Analysis and Simulation Center (VMASC).

He has published more than 150 book chapters, journal contributions, and conference papers on interoperability and architectures, over 30 of them were awarded as outstanding papers. His work on interoperability and composability challenges is reflected in the Gridwise Interoperability Framework.