A Survey on Determining Factors for Modeling Reference Architectures

Camelia Maga and Nasser Jazdi

Institute of Industrial Automation and Software Engineering (IAS), Universität Stuttgart,
Pfaffenwaldring 47, 70569 Stuttgart, Germany
{Camelia.Maga, Nasser.Jazdi}@ias.uni-stuttgart.de

Abstract. Motivated by increasing demands for efficiency, reuse approaches have been increasingly gaining significance during the last years. Systems engineering shifted from single systems to product lines, in which industrial automation systems share the same reference architecture. Different types of reference architectures are modeled with help of domain specific modeling languages (DSML). This paper discusses determining factors for modeling high-quality reference architectures in industrial automation. The identified determining factors are interesting for the following reasons: first, they are relevant for developers of reference architectures, since they aim to provide optimal reference architectures. Second, they are interesting for the design of new DSML, since these have to define appropriate concepts to support creation of highly qualitative reference architectures. Third, they are significant for tool developers, since they have to implement the required concepts.

Keywords: reference architecture, industrial automation, reusability, DSML

1 Introduction

An industrial automation system encompasses a technical system with the contained technical process, the computer and communication system, and the process operators involved. Thereby a technical system can be a technical product or a technical plant, in which a technical process takes place [1]. A coffee machine is a typical example of a technical product, because it is a mass product with a high degree of automation, having few sensors and actuators. In contrast, a technical plant is a unique system with a large number of sensors and actuators and a medium to high degree of automation. A typical example for a technical plant is an oil refinery.

For both technical plants and technical products, engineering is rarely performed from scratch. In most cases, there is a legacy principle within the domain, intended to systemize and increase the reuse degree of industrial automation systems. In consequence, many industrial automation systems within a domain share the same reference architecture. A reference architecture is defined in [2] as an abstract organization of primary elements within the domain. Reference architectures contain the concepts and rules that define the structure and the relationships among the parts

of a system, capturing the high-level design of the entire set of industrial automation systems within the domain [2], [3].

Since future industrial automation systems within a domain are based on one reference architecture [4], [5], it is crucial to ensure that the reference architecture fulfils a set of quality requirements. In case that the reference architecture has an inferior quality, engineered industrial automation systems based on it will be of low value. The reference architecture decides on the success or failure of a multitude of industrial automation systems and has implicitly strong repercussions on the entire product line. However, it is still unclear, which determining factors have to be regarded in a DSML, in order to facilitate the creation of good reference architectures. The necessity to identify these factors motivated the current investigation.

This paper is structured as follows. Section 2 gives an overview of the investigation method applied. Section 3 offers a detailed presentation of determining factors identified during the investigation. Section 4 contains the results of applying the determining factors on two different reference architectures. Section 5 concludes with a summary and provides an outlook for future work.

2 Investigation Method

The investigation was carried out for general industrial automation systems, covering both technical products and technical plants. Different approaches describing reference architectures in the domains “Elevators” (both technical product and technical plant properties), “Automotive” (technical product properties), “Avionics” (both technical product and technical plant properties), “Computer Games” (technical product properties) and “Chemical Plants” (technical plant properties) have been identified [2], [6], [7], [8], [9], [10], [11]. The domains were intentionally chosen as differently as possible, in order to investigate diverse reference architectures. Although industrial automation systems in considered domains present major differences concerning their characteristics and constraints, the identified factors are applicable for all reference architectures. During the investigation, the template shown in table 1 was used:

Table 1. Template for criteria investigation

Criterion	Purpose	Parameter Value
Which characteristics are evaluated?	How does this look in an optimal reference architecture?	How to measure the concerned reference architecture?
e.g. reuse degree	e.g. high	e.g. number of projects in which the reference architecture has been successfully used

The first column contains criteria for the evaluation of reference architectures together with explanations, why this criterion is relevant. The second column presents the ideal occurrence of the criterion, representing its influence upon the reference

architecture in the best case. The last column contains the parameter values (metrics) that permit to measure a certain criterion.

During the investigation, it has been observed that some parameters are competing. A reference architecture containing a large number of black-boxes could present deficits in variability, in case that variability was conceived mainly with module internal adjustments. Another example of competing parameters considers the support of legacy models versus the utilization of industry-supported standards. The introduction of open standards in a reference architecture might lead to incompatibilities with elder industrial automation systems, which were designed to have proprietary interfaces. Because of competing parameters, the determining factors presented in the following sections are interdependent.

After analyzing different domains, we differentiated between five types of reference architectures:

- **Functional based architecture**. This type of reference architecture considers the purpose of an industrial automation system and its decomposition into functions (e.g. data base access, coding & encoding, passenger-identification) [12]. Distribution on physical components of the industrial automation system is neglected.
- **Physical based architecture**. This type of reference architecture considers the physical modules from which the industrial automations system is built of (e.g. cabin, well, drive, buffer). This type of reference architecture can be gained from the combination of a logical and a location based reference architecture, as shown in [8].
- **Location based architecture.** This type of reference architecture considers the topographic arrangement of the modules of an industrial automation system (e.g. location of the single machines in the layout of a chemical plant, location of a controller in the electrical control cabinet, location of a component on a circuit board). The notations for this structuring type are documented in [12].
- **Logical based architecture.** This type of reference architecture considers the modules, which the industrial automation system is composed of (e.g. elevator emergency system[1], infotainment system[2], cooling system, and transport system). This type of structuring reference architectures is named "product based" according to [12], because modules in a logical based architecture contain more than physical components and can be separately sold as products [12].
- **Issues based architecture**. This type of reference architecture considers a combination of functional, physical and logical modules, which are grouped around issues (e.g. rotor-bow protection, cockpit signals, possible engine starting modes) [9]. These issues were considered relevant for further analyses [9], so that the reference architecture is constructed around them.

[1] e.g.: it contains emergency power supply, emergency lighting well, emergency lighting cabin, emergency brakes, overspeed governor, buffers, emergency phone

[2] e.g. from the automotive domain: iDrive (BMW), MMI (Audi), COMAND (Mercedes-Benz) or Porsche Communication Management (Porsche)

3 Determining Factors for Modeling Reference Architectures

This section contains the results of the survey. First, the filled-out templates were analyzed and compared with related literature. Second, a grouping of similar issues was accomplished. The single determining factors are presented in the following sections.

3.1 Utility

Reference architectures must support the underlying realizations, in order to be used as a base for an entire set of industrial automation systems. The best reference architecture would be useless if not deployable for engineering concrete, customized industrial automation systems in real projects. Optimal reference architectures should show an increased utility, as shown in [13]. The criterion implies the following parameters:

- **Functionality.** Reference architectures should ensure the functionality of the industrial automation system. More concretely, an industrial automation system based on a high-quality reference architecture should fulfil the functional requirements within the considered domain. This can be measured by taking into consideration the percentage of covered functions (realizable and still needed functions) by the reusable modules of the reference architecture. The reference architecture contains in best cases all functions required in a given domain.
- **Evolution capability**. Technology and the possible realizations of the reference architecture change in time. There are many emerging technologies and concepts, which are promising for the future [14], [15]. Reference architectures are conceived for a longer usage than single system architecture. Hence, they might incorporate modules that are realized in different technologies. We recommend avoiding that old reusable modules, whose functionality is already covered by new modules, are still kept in the reference architecture. This parameter is measured by considering the age of the existing modules in the reference architecture in relation to the number of deployments in concrete industrial automation systems.
- **Horizontal Integration**. This parameter covers the temporal aspect of the integration. Good reference architectures should offer support during all engineering phases of an industrial automation system (analysis, design, realization, test). In addition, it should be possible for the reference architecture to be deployed without translation losses between engineering phases. The parameter can be measured by the number of engineering phases, in which the reference architecture can be used.
- **Vertical Integration**. This parameter covers the multitude of disciplines involved in the engineering of an industrial automation system. When a reference architecture is valid only for a single discipline, it will be eventually discarded. Good reference architectures should be used by all disciplines involved, like electrical engineering, computer science, control engineering or mechanical engineering. This parameter can be measured by

the ratio between the supported disciplines and the total amount of disciplines involved in the engineering process.

- **Deployment.** Reference architectures contain modules with different levels of realization. Some of them are fully specified and ready to be reused, while others are generic and provide only the basic conditions for the deployment within a concrete project. It is easier to deploy a reference architecture containing fully-specified modules than generic modules. In case that it is difficult to derive concrete industrial automation systems from a reference architecture, the concerned reference architecture will not be used. This parameter considers the estimated/measured effort, which is necessary to derive a concrete industrial automation system from the reference architecture.

3.2 Usability

A reference architecture is intended to be deployed by numerous developers. Since developers have different knowledge backgrounds and stem from different disciplines, it is decisive for a reference architecture to be usable for all persons involved in the engineering process. A high usability of the reference architecture overcomes possible acceptance barriers [13], [16], [17] and finally encourages the reuse of the reference architecture. The evaluation of usability implies the consideration of a multitude of aspects like learnability [16], information display or user satisfaction [18], [19]. In [18] is recommended to focus on measurable aspects. These are grouped into the following parameters:

- **Understandability.** People use what they know and understand, so that only understandable reference architectures will be deployed for different industrial automation systems within a domain. Therefore, it is essential for reference architectures to be understandable. This parameter is measured by the existence of documentation for the given reference architecture, the existence and the quality of a construction plan which describes how to derive concrete industrial automation systems from the reference architecture and the effort necessary to get used to the reference architecture with limited previous knowledge [20], [21].
- **Tool support.** This parameter considers the existence of dedicated tools to create concrete industrial automation systems without translation errors and with less engineering effort. Since industrial automation systems become more and more complex [14], a similar evolution is observed for their reference architecture as well. In consequence, reference architectures will be used only if an appropriate tool support is ensured. This parameter can be measured by the existence and the quality of the provided tools.

3.3 Reuse Degree

A possibility to create different industrial automation systems with reduced engineering effort is trough the systematic reuse. Optimal reference architectures

should support and encourage reuse. Reuse is often considered a goal, containing a set of subordinated criteria [20], [21]. For the evaluation of the reuse support provided by reference architectures, the following parameters have been identified:

- **Coverage degree** between reusable modules of the reference architecture and concrete industrial automation systems. When industrial automation systems are built completely from reusable modules, then there is less engineering effort necessary for those realizations. This parameter measures the coverage of concrete industrial automation systems with modules provided by the reference architecture [22], [23]. A gain is registered when the reuse percentage is high, even if the parameter "value" is set to low.
- **Number of projects** in which the reference architecture has been successfully used. An indicator for the success of a reference architecture is the number of utilizations [22], [23]. When the reference architecture is reused in every project, then the gain is considerable. This remains even when the parameters "percentage" and "value" are set to low. Frequently used reference architectures of the past are very probably suitable for future utilizations [17]. In the best case, the reference architecture will be used in all future projects.
- **Value of reusable modules** compared to the entire value of the industrial automation system. Modules that cause the biggest engineering effort, and thus are expensive to engineer, should be preferentially conceived as reusable modules. When these valuable reusable modules are contained in the reference architecture, then the gain is ensured, even if the parameter "percentage" is set to low.
- **Granularity level** of reusable modules contained in the reference architecture. According to [24], there are three possibilities to handle granularity levels: as the percentage of a reusable model in the entire industrial automation system, as the complexity of the module's interfaces, and as the size of the operational code within a module. The granularity level assesses at which level the reference architecture can be reused [16]. At the first sight, coarse-grained modules are more appropriate for reuse, since they "cover" more of the final industrial automation system. Unfortunately, coarse-grained modules require effort for understanding, configuration and parameterization. In contrast, fine-grained modules imply more effort for searching, connecting and integrating them in a project-specific industrial automation system, since numerous interdependencies must be handled

3.4 Utilization of Industry Supported Open Standards

Utilization of industry supported open standards is presented in Harvard's Vetronics Reference Architecture [2] as crucial criterion for the success of reference architectures. It ensures the creation of flexible, modifiable industrial automation systems that are independent of third party companies. A reference architecture that encourages open standards remains flexible in the realization, while the industrial automation systems based on it are easy to maintain and modernize. The standardized interfaces between modules are clear (see also criterion "Usability") and can be tested

with reusable test cases. Furthermore, the utilisation of industry supported open standards facilitates the engineering of modules with plug-and-produce capability.

The current criterion is measured by considering the number and locations of non-standard modules/interfaces. The non-standard parts of an industrial automation system can be company-specific (by the sake of know-how protection), which can be considered as a positive aspect. A non-standard part can be also a module purchased from a third party company, fact that creates a dependency for this company. This aspect is considered here as negative. It must be regarded in the evaluation of the criterion if the non-standard interface/module has a positive or negative influence on the reference architecture.

3.5 Support of Legacy Models

Reference architectures should be compatible with the already existing industrial automation systems within the domain. The deployment of reference architectures should not break the compatibility chain established between various generations of industrial automation systems existing within a domain. Optimal reference architectures should offer, if possible, a valid structure for elder industrial automation systems. In this context, the number of existing industrial automation systems within the domain, which were excluded from further analyses, are evaluated. Many excluded industrial automation systems indicate incompatibility between existing and future industrial automation systems. It is recommended to analyze and document carefully the reasons for the exclusion of an existing industrial automation system from the further analyses.

3.6 Encapsulation Degree

Industrial automation systems with a high encapsulation degree are easy to understand, engineer and maintain. It is sufficient to have a high-level of understanding of the interconnected modules in order to get an overview of the whole industrial automation system. For engineering and maintenance purposes, encapsulation plays a significant role as well. Detailed knowledge is required only for some modules and the interfaces to neighbored modules. Encapsulation is also relevant for know-how protection when the delivered, encapsulated modules represent only a part of the final industrial automation system. A modular, encapsulated industrial automation system can be started-up more easily in case of repair or modifications, compared to an industrial automation system that is not modularly designed. Encapsulation, modularization and standardization are the basic principles for providing plug-and-produce –capable industrial automation systems [15].

Encapsulation plays a decisive role in engineering sustainable industrial automation systems. Two parameters for measuring the encapsulation degree of reference architectures are proposed:

- **Percentage/value** of modules that are realized as black-boxes. The ideal case is to provide reference architectures with encapsulated modules, which can be reused without knowledge of their internal behavior. Fully

encapsulated modules can be reused without any changes or tailoring activities. When the percentage of black-boxes within the reference architecture is high, then the engineering effort required to create new industrial automation systems by tailoring existing modules is reduced to a minimum. Since project-specific modifications within modules still have to be made, this parameter considers only the modules that can directly be reused as black-boxes. The differentiation between the other types of modules (configurable modules, respective modules with white spots) is made in the next parameter.

- **Percentage/value/effort** to tailor engineered modules with well-defined parameters. These configurable modules are adjusted to different requirements by changing the parameter values at well-defined, known positions. This parameter regards also the engineering effort that is necessary to tailor modules containing white spots.

3.7 Variability

Reference architectures must cover the entire domain, in which they are used. Industrial automation systems within the domain fulfill specific requirements, so that they are optimized for their specific goals. The necessity for specialized industrial automation systems leads to variants. When variants are built unsystematically, undesired effects like variants explosion, redundant engineering and maintenance difficulties occur. Good reference architectures should prevent these undesired effects by introducing a systematic variability concept in the domain. More concretely, fulfilling the criterion "Variability" means that the reference architecture provides hooks, so that new functionalities can be easily added or modified

For measuring whether a given reference architecture supports a systemized variability, we recommend to evaluate the number of variants of the modules/industrial automation systems that can be created from the reference architecture. This includes evaluation of variation points, which were defined for each module of the reference architecture. A small number of variation points indicate that only few different industrial automation systems can be derived from the reference architecture.

3. 8 Connectivity

A reference architecture is more than a collection of reusable modules. A reference architecture provides a framework, in which reusable modules are put into relation to each other. Since relationships between modules are decisive for engineering suitable industrial automation systems, the need to measure how far a good reference architecture supports these connections becomes evident. The criterion can be measured by analyzing the parameters coupling and cohesion [5], [25] between modules in the reference architecture.

When the reference architecture has two or more strongly connected modules, then these modules should be grouped together. Having different types of connections

means that energy, signals and materials are interchanged between modules. Numerous connections between modules have negative repercussions on understandability, maintainability, simplicity and encapsulation. These effects become positive when specialized tools to manage the different disciplines-specific connections are available.

When only few connections exist between the modules of the reference architecture probably some aspects are missing. The connectivity degree is measured by considering the number and type of connections to other modules.

It has been observed that increasing the vertical integration of reference architectures will raise the connectivity degree. In case a new discipline is considered, the necessary connections from its point of view must appear in the reference architecture, as well. This requires the existence of specialized tools to manage the multitude of different connections.

3.9 Traceability

Traceability includes (according to [26]) aspects like impact (analysis of related changes within the industrial automation system when a modification is made), derivation (analysis and storage of decisions made during the engineering of the industrial automation system) and coverage (gap analysis between requirements and the single modules of an industrial automation system). Because systematic and quantifiable methods to evaluate traceability are still missing [5], [26], we recommend to evaluate the traceability support.

High-quality reference architectures should support traceability. Decisions made during the creation of the architecture should be available for review of their comprehensibility at a later point in time. Another motivation for traceability is to understand which requirements are fulfilled, supported or incompatible with the reference architecture. The parameter “Traceability” is related to “Horizontal integration” and is evaluated by considering the capability to trace modules of the reference architecture to their correlated requirements or test cases. The optimal traceability support contains specialized tools for all three aspects mentioned above, namely impact, derivation and coverage.

3.10 Simplicity

A reference architecture must be simple, in order to be understood and reused in engineering of new industrial automation systems. Simplicity is relevant for understandability, encapsulation and reuse degree, as well. The evaluation of simplicity includes the analysis of following parameters:

- **Number of modules**. This parameter depends on the degree of automated support provided. When specialized tools for modeling, visualization and management are available, the maximal number of modules of the reference architecture can increase without negative effects.
- **Number and type of interfaces** between modules within considered reference architecture. Reference architectures for industrial automation

systems contain different types of interfaces. These are necessary to exchange materials, information or energy between the modules. When a reference architecture contains numerous interfaces of different types, then that reference architecture is complex. We recommend visualizing only one type of interface at a moment in time, in order to reduce complexity. In addition, a possibility to filter discipline-specific connections and interfaces is required.

- **Module size**. The optimal module size depends on the reference architecture type. There is no universally valid module size. Hence, we recommend evaluating this parameter through comparisons with similar modules in already existing industrial automation systems. This parameter is related to the granularity levels presented in [24].
- **Module nesting depth**. Hierarchically built reference architectures have modules with different nesting depths. With regard to understandability, encapsulation degree, hierarchy and maintenance, a high nesting depth of the single modules is recommended. This parameter enables the principle of information hiding. A small number of hierarchy levels shows that the reference architecture is not well structured. A large number of hierarchy levels could lead to information overflow, if not properly displayed (see also the criterion "Encapsulation degree").

3.11 Validation and Simulation Support

For each newly created industrial automation system, it is necessary to prove if the requirements are fulfilled and the industrial automation system will work. The validation against requirements and the simulation possibility should be therefore already supported in the reference architecture.

Since a reference architecture is valid for an entire set of different industrial automation systems, the validation can consider only requirements, which are common for all industrial automation systems within the domain. These requirements are named "domain requirements" [4]. In addition to the coverage check between reference architecture and domain requirements, the existence of a validation support ensures that possibly "dead" modules within the reference architecture, meaning modules, which are not needed any more, are identified and removed.

The same principle applies for the simulation capability. Because the reference architecture represents the basic architecture for a set of different industrial automation systems, the simulation must include the possibility of being adjusted, depending on the modules selected to fill in the reference architecture. The criterion evaluates the support by considering the existence of (eventually automated) validations and simulation possibilities (e.g. tools, automated check of coverage with the domain requirements, 3D simulations of the reference architecture, etc.).

3.12 Cost-value-ratio

A reference architecture should make the engineering of profitable industrial automation systems possible. Even when all other criteria are fulfilled, this fact does not guarantee that the industrial automation systems will bring profit. For example, it is possible to deliver high qualitative industrial automation systems, including multiple features, although these are not explicitly required. This situation named "Over Engineering" [27] is caused by vaguely formulated requirements, the wish to cover eventual future customer or market options and the existence of modules with a high number of features already included. Over engineering is a negative effect that should be avoided in highly qualitative reference architectures. A reference architecture should cover both low- and high-costs segments, in order to be viable for a given domain. In other words, based on the same reference architecture it should be possible to derive both high-cost industrial automation systems with an increased number of functions and low-cost ones, presenting a restricted set of functions.

This criterion is measured by the parameter over-engineering amount. This considers number/percentage/value of features, which are delivered although they are not required by customer or by the actual market situation. Please consider that this criterion can be verified only ex post, when some projects based on the given reference architecture have been executed.

3.12 Maintainability

Modeling reference architectures means to design an architecture with a longer lifetime than the system architecture of a single industrial automation system. The reference architecture should be valid for all industrial automation systems within the domain, be flexible and able to evolve. The maintainability should be facilitated for both the reference architecture itself and industrial automation systems based on a certain reference architecture. With regard to this duality, we identified the following parameters:

- **Tool support** to modify the reference architecture itself. The parameter considers the existence of dedicated tools to create, change, search or delete modules of the reference architecture. This parameter was described in "Usability".
- **Maintainability of industrial automation systems** based on the reference architecture. This parameter evaluates how difficult it is to maintain industrial automation systems based on the reference architecture. It includes aspects like the effort necessary for module replacement, independency of single realization technologies and independency of third-party companies.
- **Existence/detailing** degree of documentation provided for the single modules of the reference architecture. This parameter is relevant for both the maintainability of the reference architecture and of the industrial automation systems derived from it.

The determining factors presented in the previous sections shall be analyzed with regard to the domain. Domain-specific requirements change in time. Hence, the determining factors must be weighted for every domain. For example in the

“Elevators” domain, where the profit is increasingly gained with maintenance services, the criterion “maintainability” has an essential role. In consequence, it shall be appropriately weighted.

4 Results

In order to evaluate the determining factors for modelling reference architectures presented in the previous sections, we exemplarily assessed two reference architectures. The first one was an issue based reference architecture in the domain “Avionics”. The second one was a physical based reference architecture in the domain “Elevators”.

As presented in the previous sections, each criterion has one or more parameters. We used for each parameter a scale with minimum 0 and maximum 3 points. Zero indicates that the criterion is not supported. One means “low support”, two signalizes “medium support” and three indicates a “high support”. The better a reference architecture is, the more points are cumulated at the end. An example of the evaluation sheet used for the reference architecture in the domain “Elevators” is shown in the following table.

Table 2. Exemplary result for a reference architecture in the domain "Elevators"

Criterion	Parameter	Points	Sum
Utility			
	Functionality	2	
	Evolution Capability	1	
	Horizontal Integration	2	
	Vertical Integration	1	
	Deployment	1	
			7
Usability			
	Understandability	3	
	Tool Support	3	
			6
Reuse Degree			
	Percentage	1	
	Number of Projects	1	
	Value of Reusable Modules	2	
			4
Utilisation of Industry Supported Open Standards		2	
			2
Support of Legacy Models		3	
			3
Encapsulation Degree			
	Percentage / Value	3	
	Percentage / Value / Effort	3	
			6
Variability		0	
			0
Connectivity		0	
			0
Simplicity			
	Number of Modules	2	
	Number of Interfaces	0	
	Module Size	3	
	Module Nesting Depth	3	
			8
Validation and Simulation Support		1	
			1
Cost-Value-Ratio		3	
			3
Maintainability			
	Existence / Degree of Automated Supc	2	
	Maintainability of the Automation Sys	2	
	Existence / Detailing Degree	0	
	Hierarchy Levels	3	
			7
Total Points			**47**

Conclusion **47 of 81**

The results of the evaluation confirmed that modeling reference architectures requires significant improvements. Theoretically, this problem could be solved by using best-practices for modeling reference architectures. Unfortunately, there is still a lack of best-practices based on metrics and feedbacks, able go beyond personal opinion of the modeler. We propose to use the determining factors presented in this paper for defining concrete metrics for modeling high-qualitative reference architectures.

A second problem regards the DSMLs, in which these reference architectures are modeled. Many DSMLs do not support the determining factors identified in this paper. DSMLs have been created for supporting few aspects in the domain, without taking into consideration requirements stemming from the reusable character of reference architectures. Considering current DSML, we registered a rather low

support in modeling variability in reference architectures comprehensibly. Another suboptimal aspect was the validation and simulation support, which should be improved, as well. A third suboptimal aspect regards the support of discipline-specific views upon a reference architecture. For these reasons, we recommend to consider the determining factors when new DSMLs are designed. In case that DSMLs already exist in a domain, these results are useful for selecting the appropriate DSML before starting to model a reference architecture in that domain.

5 Conclusion

This paper presents a survey about determining factors for modeling highly qualitative reference architectures. The determining factors are intended to be used for assessing existing reference architectures modeled in different domains. Furthermore, they shall be considered when a new reference architecture is created. In addition, the results of the survey are relevant for deriving requirements on DSMLs and on modeling tools.

Future work focuses on the identification of concrete metrics for determining the quality of different reference architecture. Based on this, we aim to identify research issues related to the design of new DSMLs.

References

1. Göhner, P.: *Industrial Automation.* Lecture Notes, Summer Term 2009, Institute of Industrial Automation and Software Engineering, Universität Stuttgart, 2009.
2. Smith, M.: *Vetronics Reference Architecture.* Lecture at the DCS Corporation, Vetronics Department, Harvard, 2001.
3. The Consultative Committee for Space Data Systems (CCSDS): *Reference Architecture for Space Data Systems.* Draft recommended practice, 2007.
4. Pohl, K.; Böckle, G.; van der Linden, F.: *Software Product Line Engineering*. Springer Verlag Berlin Heidelberg, 2005.
5. Graaf, B.; van Dijk, H.; van Deursen, A.: *Evaluating an Embedded Software Reference Architecture – Industrial Experience Report.* Proceedings of the Ninth European Conference on Software Maintenance and Reengineering (CSMR'05), IEEE, 2005.
6. Pahl, G.; Beitz, W.: *Konstruktionslehre Grundlagen*. 7.Aufl., Springer-Verlag, 2006.
7. Schütten, U.: *Konzept eines modulbasierten Engineerings in der Anlagenautomatisierung*. Dissertation im Fachbereich Maschinenwesen an der Universität Duisburg-Essen, 2003.
8. Kustosch, M.; Frei, R.; Kind, W.; Kallenbach, R.: *Flexible Markencharakterisierung mit der Cartronic-Systemarchitektur.* Auto & Electronic, Heft 1, 2003.
9. Kelly, T.P.; Lam, W.; Whittle, B.R.; Mowles, A.; Rimmer, R.: *Diary of a domain analyst: a domain analysis case study from avionics.* Rolls-Royce University Technology Centre, University of York, 1996.
10. National Petroleum Refiners Association: *The Refinery of the Future Workshop Proceedings - February 1994*. Argonne National Laboratory, http://www.fischer-tropsch.org/DOE/_conf_proc/MISC/CONF-94013319/CONF-94013319_toc.htm; access in August 2009.

11. Flatholm, M.; Nedstam, J.: *Shorter Time-to-Market: Reference Architecture and Component Reuse in Internet Game Development.* Lund Institute of Technology, Dept. of Communication Systems, Lund University, 2000.
12. DIN EN 61346-1: *Industrial systems, installations and equipment and industrial products - Structuring principles and reference designations.* DIN Deutsches Institut für Normung e.V., Beuth Verlag, 1997.
13. Mili, H.; Mili, A.; Yacoub, S.; Addy, S.: *Reuse-Based Software Engineering: Techniques, Organization, and Controls.* Wiley, New York, NY, 2002.
14. Zentralverband Elektrotechnik und Elektronikindustrie e.V. (ZVEI): *Integrierte Technologie-Roadmap Automation 2015+.* Institut für Zukunftsstudien und Technologiebewertung (IZT), Frankfurt, 2006.
15. International Council on Systems Engineering (INCOSE): http://www.incose.org/, access in August 2009.
16. Dujmovic, S.: An *Understandable and Configurable Domain-Specific Framework for Industrial Automation Applications.* 33rd International Conference on Technology of Object-Oriented Languages and Systems (TOOLS), Mont-Saint-Michel, Frankreich, 2000.
17. Jost, P.: *Evolutionäres Domain-Engineering zur Entwicklung von Automatisierungs-systemen.* Dissertation am Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart, 2006.
18. Bias, R.; Mayhew, D.: *Cost-Justifying Usability.* Academic Press Limited, Boston, San Diego, New York, 1994.
19. Process Improvement Technical Committee (PITC): *Practical Software and Systems Measurements. Measurement for DoD Projects.* INCOSE Technical Committee, Working Group Technical Data 1.2.3., 2004.
20. Sommerville, I.: *Software Engineering.* 5th Edition, Addison-Wesley, Reading, 1995.
21. Karlsson, E.: *Software Reuse: A Holistic Approach.* Wiley, New York, 1995.
22. U. Löwen, K. Dencovski, T. Wagner*: Integration of Information and Tools: Where are the Gaps?*. 15th Daratech Plant Conference, February 23-25, Houston, Texas, http://www.daratechplant.com/, 2009.
23. VDI/VDE: *Richtlinie 3695: Engineering von Anlagen- evaluieren und optimieren.* Herausg. vom VDI/VDE. Ausg. 2009.
24. F. Griffel: *Componentware. Konzepte und Techniken eines Softwareparadigmas.* Dpunkt Verlag Heidelberg, 1998.
25. Göhner, P.: *Software Engineering II.* Lecture Notes, Sumer Term 2009, Institute od Industrial Automation and Software Engineering, Universität Stuttgart, 2009.
26. Saglietti, F.: Vorstellung des Forschungsprojektes „Musterbasierte Analyse und Korrektur von Trace-Anomalien“, http://www11.informatik.uni-erlangen.de/Forschung/Projekte/PTRAC/index.html , 2008.
27. Broy, M.; Geisberger, E.; Kazmeier, J.; Rudorfer, A.; Beetz, K: *A Requirements-Engineering-Reference Modell.* In: Informatik-Spektrum, 30 (2007) Nr. 3, S.129-130, 2007.