An analysis of MoReq2010 from the perspective of TOGAF

Conference Paper · October 2011

DOI: 10.1007/978-3-642-24355-4_34

3 authors, including:

Ricardo Vieira
Technical University of Lisbon
28 PUBLICATIONS **127** CITATIONS
SEE PROFILE

José Borbinha
University of Lisbon
194 PUBLICATIONS **929** CITATIONS
SEE PROFILE

An analysis of MoReq2010 from the perspective of TOGAF

Ricardo Vieira[1], Francisco Valdez[1], José Borbinha[1]

[1] INESC-ID, Information Systems Group
Rua Alves Redol, nº 9 1000-029, Lisbon
{rjcv, francisco.valdez, jlb}@ist.utl.pt

Abstract. The practice of Records Management in organizations is becoming a subject of research as organizations increasingly face the dematerialization of their processes. Standards and requirements have emerged in order to assist organizations to manage digital records with the purpose of streamlining their processes and ensuring accountability and evidence of their actions. However, there is still no guidance on how to embody these requirements into the organization's Enterprise Architecture. As a result, this paper explores and analyzes a requirements specification for records management (MoReq) from the perspective of an enterprise architecture framework (TOGAF). It intends to give the basic insights about how to incorporate such requirements into the development process of building an Enterprise Architecture by matching them with the corresponding architecture specification element or domain.

Keywords: Records Management, Enterprise Architecture, MoReq, TOGAF, Architecture Requirements Specification.

1 Introduction

Organizations are facing new emerging challenges due to the rapid change of their business and technological environment, so Enterprise Architecture has emerged as a valuable concept to help them dealing with the increasing complexity of their information systems [1] [2]. Although these governance techniques are responsible for handling the relevant information of an organization, it is also essential to properly manage the records created within business processes. As such, the principles, standards, and requirements related to Records Management must also be embodied in these architectures. Apart from constituting an essential part of any organization and being considered a good business practice, the management of records can also bring benefits in terms of transparency, compliance, legal pursuance and process efficiency.

An actual important reference in the area of Records Management is the Model Requirements for Electronic Records (MoReq)[1]. However, there is still no comprehensive knowledge about how to incorporate those requirements into an organization's Enterprise Architecture. That is the motivation of this paper, where we

[1] http://www.dlmforum.eu/

present an analysis of MoReq from the perspective of The Open Group Architecture Framework (TOGAF) [3], an intensive documented approach for developing Enterprise Architectures.

2 Records Management and MoReq2010

The ISO 15489 defines **Records Management** as "the field of management responsible for the efficient and systematic control of the creation, receipt, maintenance, use and disposition of records, including the processes for capturing and maintaining evidence of and information about business activities and transactions in the form of records" [4].

In this context, an **Electronic Records Management System** (ERMS) is "an automated system used to manage the creation, use, maintenance and disposal of electronically created records for the purposes of providing evidence of business activities" [5]. In order to complement this concept it is also essential to define a **record** as the consequence or product of an event or a business transaction, which can exist in any format.

Within this domain, MoReq stands for "Model Requirements for Electronic Records" and is a requirements specification for ERMS that has been developed by DLM Forum and strongly supported by the European Commission. Its first version was published in 2001 and revised in 2008, resulting in the untitled MoReq2. The major changes in this second version were the inclusion of an independent MoReq-based XML Schema, as also of a test framework and the addition of a "Chapter Zero" section to provide guidance on legislative and regulatory requirements [6].

Since the publication of MoReq2 the MoReq Governance Board, responsible for monitor and plan new developments for this specification, has received considerable feedback. As a consequence it was decided to undertake two fully public consultation phases as part of the development of a third version, named MoReq2010. By the time of this paper the consultation phases are already closed, and a stable MoReq2010 draft is available for public consultation, on which this paper writes about (the final version is expected for May 2011). MoReq2010 contains 475 mandatory requirements (9 more than MoReq2), which are divided in 16 sections and 6 modules.

MoReq2010 brought a big change in the structure of the requirements from the earlier versions, now grouped in modules of functionality surrounding a reduced set of core requirements. Those modules are conceived to be easily combined to specify a specific MoReq compliant ERMS according to the needs of an organization, resulting in multiple possible shapes of ERMS [7]. This new approach also allows evolutionary revision and upgrade, enabling MoReq2010 to adapt to new innovations and new practices in both the technology and in the field of records management.

3 TOGAF

The emerging requirements related to the access of information are increasingly becoming a critical part of any organization as its size and complexity growths. This

is precisely where Enterprise Architecture comes into play by providing rigorous descriptions of complex systems with diverse concerns.

In this context, **Enterprise Architecture** is defined as "a set of principles which guide design, selection, construction, implementation, deployment, support, and management of information infrastructure" [8]. More specifically, Enterprise Architecture describes the structure of an enterprise by exposing the composition of the enterprise components and their relationships with the environment. It aligns enterprise goals and drivers, business processes, organizational structures and behaviors, information, software applications, computer systems and technological infrastructure [1]. As a result, Enterprise Architecture is also a tool for identifying opportunities to improve the enterprise and the business itself by pursuing its future effectiveness and efficiency.

In order to realize these architectural descriptions, architects rely on frameworks that offer the tools, methods, techniques, descriptions, views, models and guidelines which are able to support the production of enterprise architectures.

TOGAF stands for "The Open Group Architecture Framework", and it was developed by the Architecture Forum of the Open Group[2] with the purpose of defining a framework for developing an Enterprise Architecture. It is based on an iterative process that supports four architectural domains [3]:

- **Business Architecture**: defines the business strategy, governance, organization, and key business processes.
- **Data Architecture**: describes the structure of an organization's logical and physical data assets and data management resources.
- **Application Architecture**: provides a blueprint for the individual application systems to be deployed, their interactions, and their relationships to the core business processes of the organization.
- **Technology Architecture**: describes the logical software and hardware capabilities that are required to support the deployment of business, data, and application services.

These domains are developed through a repeatable and iterative process that supports their development in terms of content, transition, and governance, according to business goals and opportunities. This method is called "Architecture Development Method" (ADM) and as represented in Fig 1 is understood as in the core of TOGAF [3].

In the preliminary phase of TOGAF, the organizational context is reviewed as well as relevant principles, standards and guidelines that will impact the architecture.

The process begins with the development of an architecture vision (phase A) in which the stakeholders are defined together with their own concerns. Also, business requirements, goals, drivers and constrains are identified and confirmed. This architecture vision guides the development of the business architecture (phase B), information systems architecture (phase C), and technology architecture (phase D).

Based on these architectures, solutions and opportunities are identified (phase E) in order to derive "Transition Architectures" that deliver continuous flow of business value in support of enterprise goals. This phase takes both a business and technical

[2] http://www.opengroup.org/

perspective to address IT governance activities. Next, a migration plan is created (phase F) to ensure that the adequate resources are allocated and that business value is delivered to each project.

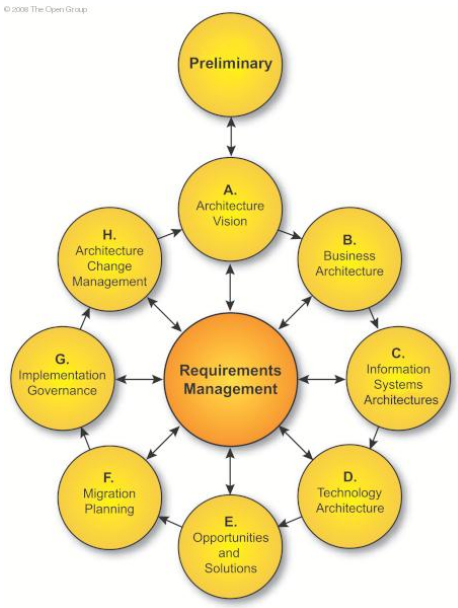

Fig 1: TOGAF – The phases of ADM **[3]**.

The architectures produced are then implemented (phase G) through an organizational-specific development process. Finally, these are subject of an architecture change management (phase H) which aims to ensure that the architecture continues to fit-for-purpose and still delivers business value.

TOGAF also provides the concept of Enterprise Continuum which explains "how generic solutions can be leveraged and specialized in order to support the requirements of an individual organization" [3]. In this sense Enterprise Continuum includes both the Architecture Continuum and the Solution Continuum. While the former specifies methods for classifying architectures assets as they evolve from generic to more specific ones, the latter describes the realization of Architecture Continuum through reusable building blocks and vice-versa.

4 Requirements on TOGAF

Requirements management is the central process of the TOGAF ADM, as shown in Fig 1, and is crucial to the entire framework since it clarifies some of the uncertainty and the need of managing changes during EA processes. The objective of this phase is to "identify requirements for the enterprise, store them, and feed them in and out of the relevant ADM phases, which dispose of, address, and prioritize requirements" [3]. It's also important to state that the requirements management process is designed to

be a dynamic process that reacts to, and encourages, changes by adding, removing and updating requirements continuously.

TOGAF recognizes that the world of requirements engineering is very rich so they don't mandate any process or tool to perform the requirements management process. Rather, they prefer to describe what outputs a process like this should have and recommend some resources for the process.

One of those resources is the Business Scenario – a technique to discover and document business requirements for the architecture and the implied technical requirements. A Business Scenario "represents a significant business need or problem, and enables vendors to understand the value to the customer organization of a developed solution" [3]. The depth of the description given in the technique depends on the scenario being described but typically it should have references to the business process, or the application that can be supported by the architecture, the technology environment, the actors of the scenario and the expected results from the execution of the EA process.

With a good Business Scenario it is possible to derive a set of requirements that describe the whole problem description, or to clearly understand the value of the problem and the relevance of potential solutions. It's also essential to understand that the creation of a scenario is a task not only for the architect but also for all the stakeholders involved in the business problem being described [3]. That involvement is crucial to ensure that the problem, goals and requirements are properly captured.

Another recommended source suggested by TOGAF is the Volere[3], a trademark brand owned by Atlantic Systems Guild that offers a collection of requirements resources which include courses, templates, books, processes, etc. One of the most known resources is the Volere Requirements Specification Template [9], a useful template to use as a foundation for requirements specification. It provides a structured template that describes the most appropriate requirements types that should be used in the business, scientific and software systems domains providing a checklist, structure, and traceability paths for the requirements. Another important resource provided by Volere is the Requirements Tools section[4] that offers a large, and increasing, number of Requirements Management tools.

Therefore, according to TOGAF, after finishing the Requirements Management process, one should produce two major outputs: the Requirements Impact Assessment and the Architecture Requirements Specification. The first one is used to assess how the architecture requirements imply the current state of the architecture by identifying the changes that should be made and their implications. It is typically identified over requirements by exploring the stakeholders, the importance or given priority, the phases that need to be reviewed, new requirements priority according to new requirement(s), recommendations on how to manage the impact of the modifications, and a repository reference number.

On another hand, the Architecture Requirements Specification "provides a set of quantitative statements that outline what an implementation project must do in order to comply with the architecture" [3]. It addresses success measures; architecture

[3] http://www.volere.co.uk
[4] http://www.volere.co.uk/tools.htm

requirements; service contracts; implementation guidelines, specifications and standards; interoperability requirements; constrains and assumptions.

Table 1: Definition of the sections that an architecture requirements specification must contain.

Section	Definition
Success Measures	Lists "indicators or factors that can be tracked, usually on an ongoing basis, to determine success or alignment with objectives or goals" [3].
Architecture Requirements	Section divided in three architecture requirements types: Business, Information System and Technology. Definitions below.
Business Architecture Requirement	Requirement related to all the aspects of the business processes that can be identifiable as a segment of an organization's overall mission [10] [11].
Information System Architecture Requirement	Requirement that reflect a property that is essential for an IT system to perform its function.
Technologic Architecture Requirement	Requirement with concerns about the technologic components available from the market or configured within the organization into technology platforms.
Architecture Service Contracts	Describes the joint agreements between development partners and sponsors on the deliverables, quality and fitness-for-purpose of an architecture. The contracts can be about business or application services [3].
Implementation guidelines, specifications and standards	Describes the guidelines, specifications and standards recommended or mandated for achieving success.
Interoperability Requirements	Lists the requirements that define or relate to the ability to share information and services.
Constrains	Requirements that place a condition on the achievement of a goal describing a restriction or a limitation [12].
Assumptions	Requirements where we assume something to be true in order to achieve it.

5 MoReq2010 as an Architecture Requirements Specification

MoReq2010 is a requirements specification for ERMS resulting from a requirements management process that gathers, analyzes, and categorizes requirements for that kind of systems. But is this specification broad enough to be used as an Architecture Requirements Specification? Does the specification address all of the sections defined in TOGAF? To answer these questions one should define what an Architecture Requirements Specification must be according to TOGAF. This is given by Table 1,

from where we can compare MoReq2010 specification with the elements listed in Table 2 and match the corresponding requirements.

Table 2: Number of Architecture Requirements present in the non-functional requirements chapters of MoReq2010.

MoReq2010 Non- Functional Requirements Chapter		Business Requirements	Information System Requirements	Technologic Requirements
1.16	**Business Continuity**	1	9	5
1.17	**Performance and Scalability**	1	15	2
1.18	**Operation Compliance**	0	15	0
101.4	**Graphical User Interfaces**	0	12	0
102.4	**Application Programming Interfaces**	2	0	5
202.4	**Monolingual Thesauri**	0	1	0
301.4	**Managed Components**	0	2	4
302.4	**Unmanaged Components**	5	3	0
	Total	**9**	**55**	**16**

Table 3 Possible Architecture Service Contracts found on MoReq2010.

Architecture Service Contract	Stakeholders Involved	MoReq2010 Requirements	Goal of the contract
Business	ERMS Developer	R1.5.11 N1.6.2-9 N1.18.8-15 N102.4.4-5	- Schedule periodic updates and backups of the ERMS - Define system maintenance and support- Define costs - Define Documentation to be provided - Schedule API updates
Application	Authentication Service Developer	R1.7.1 N1.18.2	- Provide secure access to the ERMS user
Application	Directory Service Developer	R1.7.4 N1.18.3	- Provide user information to the ERMS

Analyzing MoReq2010 we couldn't find any requirement that can be understood as a possible ***success measure***. However, like his ancestor MoReq2, the specification is going to have a framework test that intends to provide a way of determining the alignment of a system with MoReq2010, so in that sense achieving the same goal as the success measures.

Since MoReq2010 is a requirements specification for an information system (an ERMS) it was not strange to realize that all the functional requirements described fall in the category of Information System ***Architecture Requirements***. However it also was possible to identify some business and technologic requirements in the non-functional requirements of MoReq2010, as the Table 2 shows.

In what concerns ***Architecture Service Contracts*** there isn't any section in MoReq2010 that list them but it's possible to find references to three possible contracts, described in Table 3, on the requirements of the specification.

Regarding guidelines, specifications and standards, MoReq2010 states that:

- ISO 15489 and the emerging ISO 30300 are solid internationally recognized resources for organizations to refer to when planning their wider business objectives around MoReq2010 compliant records systems.
- The ERMS must generate universally unique identifiers (UUID) and recommends using the approaches listed in RFC4122 for the process.
- The ERMS must use valid language identifiers compliant with RFC5646 and the IANA Language Subtag Registry.
- The ERMS must be able to capture, store and export Unicode text and must use it for all textual metadata elements.
- The ERMS must keep timestamps that are compatible with W3C XML dateTimeStamp formats.
- The ERMS must support the export of entities, their metadata and their components using the W3C EXI (Efficient XML Interchange) standard.
- When the GUI of the ERMS is hosted within a web-browser interface the ERMS must meet the W3C Web Content Accessibility Guidelines (WCAG) 2.0 and the W3C Accessible Rich Internet Applications (WAI-ARIA) 1.0 to at least conformance level A.

Interoperability in MoReq2010 is restricted to the XML Schema that underpins the specification. Using that schema, the ERMS is compliant with MoReq2010 which enables the transfer of records between systems without losing contextual data and providing the ability to maintain the record's lifecycle after transfer. Moreover, in order to achieve the goal of interoperability, MoReq2010 is much more prescriptive than its predecessors in the sense that it can provide a way of interpreting the metadata of another system.

Interoperability is also another subject that DLM Forum wants to continually address in future versions of the specification so that real-time collaboration between records and archival systems, live transfers, and accessibility can also be possible. Concluding on this, MoReq2010 can be seen as a starting point for that goal.

When looking for ***assumptions and constrains*** we realize that almost all MoReq2010 requirements are constrains in the sense that they are so specific that avoid other interpretations or implementations. This complies with the DLM-Forum objective of making the specification as precise as possible to improve the

interoperability of MoReq2010 systems. In what concerns the assumptions, the requirements that are related to the architecture application service contracts previously described are the only ones that can be seen as assumptions since they depend on the supposition that the authentication and directory services work correctly.

6 Conclusions and Future Work

By the results observed in the previous section we can conclude that MoReq2010 is not complete enough to be used as an Architecture Requirement Specification in the scope of an Enterprise Architecture framework as TOGAF. Mostly it lacks on success measures and on insufficient number of Business and Technologic Architecture Requirements. However the requirements specification address almost all the concerns that an Architecture Requirements Specification must have, so it should be considered at least as a good starting point for further research in this area.

Another conclusion that we can extract from our analysis is that are concerns that maybe MoReq2010 fails to properly address. We could agree that some of them may be out of the scope of the specification, but other relevant requirements may be really pointed as missing. For example the lack of technologic architecture requirements can be seen as a valid approach since it brings less constrains to the information system architecture, meaning that the ERMS doesn't depend on specific technology to be implemented. But that's only one viewpoint since others may say that some functions of the system fully depend on technologic requirements. In fact this can be an issue to explore in upcoming work.

In the same topic, we pretend in the future to use the same methodology to analyze and adapt other records management resources besides MoReq, so that in the end we can achieve a complete records management profile that fits the TOGAF framework. We also expect that this work will be relevant for the future development of MoReq.

7 Acknowledgments

This work was supported by FCT (INESC-ID multiannual funding) through the national PIDDAC Program funds and by the European Commission through the SHAMAN project (contract FP7-ICT-216736).

8 References

1. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication, and Analysis. Springer, Berlin/Heidelberg (2005)
2. Borbinha, J.: It Is the Time for the Digital Library to Meet the Enterprise Architecture. In : ICADL'07, pp.176-185 (2007)

3. The Open Group: TOGAF Version 9: The Open Group Architecture Framework. Van Haren Publishing, Zaltbommel, Netherlands (2009)
4. International Organization for Standardization: ISO 15489: Information and documentation - Records management - Part 1: General. ISO Std. (2001)
5. National Archives of Australia: Digital Recordkeeping: Guidelines for Creating, Managing and Preserving Digital Records. Exposure draft ISBN 1 920807 08 X, Commonwealth of Australia (May 2004)
6. DLM-Forum: MoReq2010 - a new world standard for compliance & the best practice information management for all sectors. (2010)
7. Fresko, M.: MoReq2010 - on the track or off the rails? (2010)
8. Spewak, S.: Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology. John Wiley & Sons, New York (1995)
9. Robertson, J., Robertson, S.: Volere Requiremens Specification Template. (2006)
10. R., K., M., P., M., R.: A framework for integrating business processes and business requirements. In : EDOC 2004, Monterey, USA (2004)
11. Lavoie, B., Henry, G., Dempsey, L.: A service framework for libraries. (2006)
12. Antón, A.: Goal-based Requirements Analysis., Colorado (1996)