Architectural Pattern for Inter-Organizational Middleware Systems

Radhouane B.N. Jrad[1(✉)] and David Sundaram[2]

[1] OJI Fibre Solutions, Auckland, New Zealand
Rad.Jrad@OJIFS.co.nz
[2] University of Auckland, Auckland, New Zealand
D.Sundaram@Auckland.ac.nz

Abstract. Effective Business-to-Business (B2B) relationships typically rely on seamless integration of partner's processes. Inter-Organizational Information Systems (IOIS) have largely been endorsed as B2B enablers. They are defined as automated Information Systems crossing organizational frontiers and aiming to synergize partners' efforts in increasing competitiveness and cost management (Eom 2005). The components in IOIS responsible for the actual bridging between partners' heterogonous systems are referred to as Inter-Organizational Middleware Systems (IOMS). While IOMS critically hold business information, they lack both research and standardization. Instead, chaotic and costly efforts to architect and manage IOMS have dominated the market. As a remedy, we propose an IOMS-specific architectural pattern that could be used to develop its architecture(s). First, the notions of frameworks and architectures are presented. Then, approaches to IOIS architecture and process management are discussed, before IOMS's need for its specific architectures is presented. The MAPIS architectural design is then proposed and its merits and limitations are discussed.

Keywords:: MAPIS · IOMS · IOIS · IOS · Framework · EA · SOA · EDA · Adaptive · Middleware · Architecture · Pattern · IADR · Action Research · Design Science

1 Introduction

Humans have been architecting their assets since the beginnings (Jarzombek 2013). Classically, architecture is regarded as the art and the practice of carefully designing and constructing buildings in a manner that typically reflects the style of a specific period, place, and/or culture (Oxford 2014). Over the last decades, however, the scale of the definition has been widened to embrace other human artifacts including Information Systems (IS). The aim of this paper is to look at architectures in IS, and to put forward arguments about the need for a specific architecture to Inter-Organizational middleware Systems (IOMS), and then to actually propose and validate an architectural pattern towards answering such a need.

The ever-increasing complexity of IT created a need for order and structure in a similar way to the construction field. IEEE presents architecture in IT as "fundamental concepts or properties of a system in its environment embodied in its elements,

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2016
P.C. Vinh et al. (Eds.): ICTCC 2016, LNICST 168, pp. 384–390, 2016.
DOI: 10.1007/978-3-319-46909-6_35

relationships, and in the principles of its design and evolution" (Drews and Schirmer 2014). Since Zachman identified the need for architecture in IS (Zachman 1987), multitudes of subgenres emerged to specifically focus on various aspects of the enterprise and its IT assets. Enterprise Architecture (EA) could be defined as a precise and unambiguous future-oriented practice for conducting fundamental analysis, design, planning, implementation, and governance of an enterprise's present and target IT landscapes; and optimizing and integrating its processes, while specifically accounting for fragmented and legacy processes (Federation of Enterprise Architecture Professional Organizations 2013; Kotusev and Storey 2015; Niemann 2006; Winter and Fischer 2006). However the development of EAs in a multi-organizational environment or for the inter-organizational use remains scarcely researched (Drews and Schirmer 2014). In today's turbulent business conjuncture where alliances are true needs in the business spectrum, the lack of integrative architectures at various levels represents a business risk (He et al. 2015). It has therefore become important that organizations promote the integration of B2B partners into their EA analysis and planning (Drews and Schirmer 2014). In that line, a clear and adaptive architecture for processes that spread beyond the organization's frontier has become a must.

2 IOIS Architecture and Processes

B2B can be achieved by different means. One particular concept that gained an explosive interest from the business world over the last decades is the Inter-Organizational Information System (IOIS). IOISs are shared automated ISs spanning over multiple organizations, with every partner managing their part of the IOIS within their own organization's structural, strategic, technological and commercial context (Jrad and Sundaram 2015a). Despite the IOIS as a phenomena has links with virtually every major area of IS research it still fails to attract enough research interest (Haki and Forte 2010; Jrad 2014). In particular, handling IOIS diffusion over multiple organizations, locations, countries, legal systems, cultures, and time zones remains feebly investigated (Jrad and Sundaram 2015b). It is therefore only fair to say that there has been no real or strong effort standardizing the approach to architect IOIS.

From an architectural perspective, Event-Driven Architecture (EDA) has been present in IOISs because it is business-logic friendly. In effect, EDA allows the development of processes that react to event-driven change of statuses (Maréchaux 2006; McGovern et al. 2006), e.g. when stock status changes to "critical". However, the arrival of Service-Oriented Architecture (SOA) has allowed IOIS to become better and more agile at integrating processes because of SOA's emphasis on loosely coupling as well as reusing systems, components, and processes through common standards and protocols (Haki and Forte 2010). Through SOA, IOIS permits otherwise independent organizations, to share common functionalities instead of each cumbersomely developing their own version of the same functionality or to develop too-complex mechanisms for sharing information (He et al. 2015; Maréchaux 2006). While the reliance of SOA on web services as front-ends permits IOIS to be efficient in including and excluding partners, the architecture allows for a lower level of trust amongst partners in the same IOIS. Indeed, since web services are the first point of contact for partners,

disabling or enabling their access usually equates to adding or deleting an organization from the IOIS process.

The key feature associated with B2B and IOIS is "Integration". In order to enable business adaptivity, it is undeniably important that the integration aspect of inter-organizational business processes is designed, architected, governed, and managed on its own specifities and merits, independently from other IOIS layers. From industrial view, the authors have noted that a lack of explicit isolation of the integration layer has been an important element in the stories of failed IOIS and ERP implementations.

3 MAPIS: An IOMS Architectural Pattern

The integrator component in IOIS allowing the actual bridging between partnering organizations is referred to as the Inter-Organizational Middleware System (IOMS) (Fig. 1). IOMS is defined as an "inter-organizational collection of Enterprise Application Integration (EAI) and Enterprise Messaging Systems (EMS) policies, procedures, methods, and services that work together to allow heterogeneous applications from different organizations to communicate, exchange information and validate each other's input and output" (Jrad 2014). If it is SOA-based, IOMS can be regarded as an advanced inter-organizational Enterprise Service Bus (ESB).

IOMS goes beyond merely establishing technological channels between otherwise architecturally heterogeneous computer systems from different partners. It in fact is a true part of the business process spectrum and as such holds business logic. However, because of its invisibility to most stakeholders, and because it is not a business generating instrument, it is common to ignore IOMS requirements in terms of lifecycle management and governance or to include them as part of IOIS or ERP processes (Jrad et al. 2013; Jrad 2014). IOMSs have in reality their own unique characteristics and particular risks and impact associated with their governance and projects. Accordingly, they need to be managed in a specific and significantly different way from other ISs. IOMS has resolutely become an expert domain requiring a strong combination of general IT knowledge with specific domain knowledge as well as business knowledge (Jrad and Sundaram 2015b). Subsequently, the importance of engaging the organization's IOMS teams in projects has increased in importance. As opposed to other ISs, IOMS projects are indeed better managed and run by the organization's employees while being supported by external resources, not the other way around

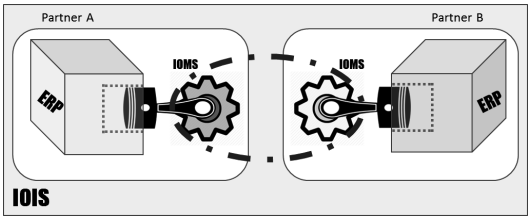

Fig. 1. Placement of IOMS inside IOIS

(Jrad and Sundaram 2015b). As such, researching IOMS would be most suitable using methodologies that allow the researcher to be knowledgeable of the subject and involved in the project, even part of the organization itself.

Architecting IOMS cannot be fully achieved using existing IOIS and ERP upgrade methodologies (Jrad 2014). Oddly, not only there is a lack of research into the subject, but market observation has highlighted a clear lack of managerial acknowledgement of the need to architect IOMS. Instead, organizations tend to resort to ad-hoc, "quick-and-dirty" solutions to fulfill integration requirements. The culmination of reliance on quick fixes is often a serious problem of legacy processes that has developed even though IOMS as a concept is relatively new (Jrad and Sundaram 2015c). One of the reasons for the absence of standardized IOMS architectures is the shortage of IOMS frameworks. Applying partly or completely irrelevant frameworks has been identified as a common cause for failed IOMS upgrades (Jrad and Sundaram 2015c). In this work, we are proposing a high-level IOMS-specific architectural pattern that could be applied to create actual adaptive architectures for IOMS. We label this architectural pattern as the Middleware Architecture Pattern for Inter-organizational Systems (MAPIS) (Fig. 2). MAPIS was constructed based on the Framework for Upgrading IOMS (FUI) which is an IOMS-specific framework (Jrad and Sundaram 2015c). MAPIS accounts for both SOA and EDA designs. While SOA lacks reactivity to events, EDA suffers from processes dependency. The ED-SOA combination (Levina and Stantchev 2009) allows for event and service based processes to coexist while ensuring services decoupling.

Looking at Fig. 2, MAPIS divides transactions handling into 2 tiers: At first Front-End processes receive details and content of transactions and transform them into canonical/standardized formats. Then, Back-End processes perform the required tasks associated with these transactions. The separation of the processes into 2 tiers ensures that regardless of what is being received all data of same nature is processed in a similar (standard) way. For instance, while various partners would send invoices in different formats, these invoices are converted into the organization's standard invoice format before being processed. When a new partner joins the IOIS, the effort is put into transforming their invoices into the standard format without affecting other partners'

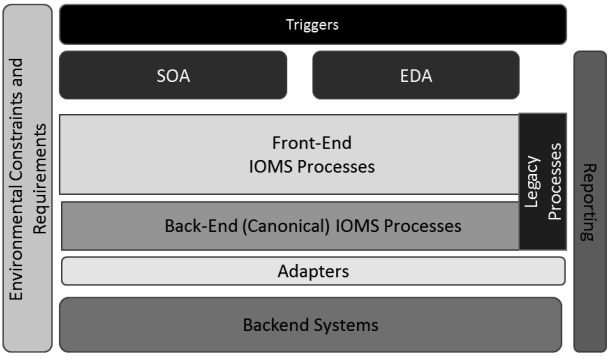

Fig. 2. Middleware Architecture Pattern for Inter-organizational Systems (MAPIS)

invoices. In the eventuality that the canonical format requires modification, the change would be in one place for all partners. Such maneuver enables the flexibility required in managing partnerships. Importantly however, MAPIS acknowledges that not all processes can be easily subjected to the 2-tier process. Legacy processes can be a challenge to reengineer or even impossible to modify. These types of legacy processes are referred to as unupgradable legacy processes (Jrad et al. 2013).

The triggers in the MAPIS refer to internal and external start points for the process (e.g. user passing an order), while the environmental constrains and requirements represent legal, project, security and other environmental aspects. The MAPIS architectural pattern has been constructed based on a combined 20 years of the authors' experience in IOIS and IOMS projects. As such, it has gone through multitude of iterations from concept to refinement, to application and validation. To test its implementability, an IOMS project in an international company with advanced e-business capabilities and cross-continental IOIS and IOMS systems was used. Guided by the concepts and principles of Insider Action Design Research (IADR), we embarked on a project to upgrade the IOMS infrastructure to the latest available technology (if/where possible). IADR methodology uses the researcher's interaction with the observed phenomenon and the feedback loops to create and implement a system as an outcome of the research (Jrad et al. 2014; Jrad and Sundaram 2015c). IADR consists of Design Science iterations aiming at building and refining the solution, backed by iterations of Action Research cycles to implement and analyze the said solution (Jrad et al. 2014). After agreeing on the FUI framework as a basis for developing the architecture, and leaning on IADR, Design Science cycles were used to design and collect feedback about the architecture, while the actual implementations were performed using Action Research rounds. Every loop returned feedback that was used to enhance subsequent cycles. Multiple iterations were executed until the final and agreed architecture was reached.

4 Discussion and Conclusion

In this paper we proposed a tailored architectural pattern for IOMS that we labeled Middleware Architecture Pattern for Inter-organizational Systems (MAPIS). We argued that IOMS cannot be addressed as part of the IOIS or ERP systems, but instead as an independent concept. MAPIS architectural pattern was based on the FUI framework supported by the authors' extensive experience in IOIS and IOMS. To ensure adaptivity as well as efficiency in reacting to changes in the business network, the architecture allows for event as well as service driven approaches, while remaining in compliance with the SOA principles. As a means of evaluation, the architecture was put to implementation in the context of a multinational organization taking part of a complex IOIS. Driving the implementation based on the concept of Insider Action Design Research (IADR), multiple iterations were conducted to validate both the design and implementation of the solution, with continuous feedback. The merits of MAPIS, however, should not hide its limitations. First, further implementations are required to validate the findings, and as such, another implementation is planned in a second multinational organization. Second, MAPIS does not account for simpler

business scenarios. Indeed, it might be regarded as too complex for basic business contexts. MAPIS is therefore restricted to the context of mid to large organizations subscribing to complex IOISs. Further research is critically needed to offer different options when it comes to IOMS architecture.

References

Drews, P., Schirmer, I.: From enterprise architecture to business ecosystem architecture. In: 2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations, pp. 13–22. IEEE Comput. Soc. (2014)

Eom, S.B.: Inter-Organizational Information Systems in the Internet Age. IGI Global, Missouri (2005). Eom, S.B. (ed.)

Federation of Enterprise Architecture Professional Organizations: A common perspective on enterprise architecture. Archit. Gov. Mag. **9**, 1–12 (2013)

Haki, M.K., Forte, M.W.: Inter-organizational information system architecture: a service-oriented approach. In: Camarinha-Matos, L.M., Boucher, X., Afsarmanesh, H. (eds.) PRO-VE 2010. IFIP AICT, vol. 336, pp. 642–652. Springer, Heidelberg (2010)

He, J.J., Jrad, R.B.N., Sundaram, D.: Adaptive business network systems. In: Proceedings of the Sixth International Conference on Information, Intelligence, Systems and Applications (IISA 2015) (2015)

Jarzombek, M.M.: Architecture of First Societies: A Global Perspective. John Wiley & Sons Inc., New Jersey (2013)

Jrad, R.B.N.: A roadmap for upgrading unupgradable legacy processes in inter-organizational middleware systems. In: 2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS), pp. 1–6 (2014)

Jrad, R.B.N., Ahmed, M.D., Sundaram, D.: Upgrading unupgradable middleware legacy processes: misconceptions, challenges, and a roadmap. In: Proceedings of SEM 2013: 1st International Workshop in Software Evolution and Modernization, vol. 1(1), p. 8 (2013)

Jrad, R.B.N., Ahmed, M.D., Sundaram, D.: Insider action design research: a multi-methodological information systems research approach. In: Research Challenges in Information Science (RCIS), pp. 28–30. IEEE Comput. Soc., Marrakech (2014)

Jrad, R.B.N., Sundaram, D.: Challenges of inter-organizational information and middleware system projects: agility, complexity, success, and failure. In: Proceedings of the Sixth International Conference on Information, Intelligence, Systems and Applications (IISA 2015) (2015a)

Jrad, R.B.N., Sundaram, D.: Inter-organizational information and middleware system projects: success, failure, complexity, and challenges. In: Americas Conference on Information Systems (AMCIS), p. 12. Communications of the AIS, Puerto Rico (2015b)

Jrad, R.B.N., Sundaram, D.: Inter-organizational middleware systems: a framework for managing change. In: Proceedings of the Sixth International Conference on Information, Intelligence, Systems and Applications (IISA 2015) (2015c)

Kotusev, S., Storey, I.: Consolidating enterprise architecture management research. In: Bui, T.X., Sprague Jr., R.H. (eds.) 48th Hawaii International Conference on System Sciences (HICSS), pp. 4069–4078 (2015)

Levina, O., Stantchev, V.: Realizing event-driven SOA. In: Proceedings of the 2009 4th International Conference on Internet and Web Applications and Services, ICIW 2009, pp. 37–42 (2009)

Maréchaux, J.-L.: Combining Service-Oriented Architecture and Event-Driven Architecture Using an Enterprise Service Bus. IBM Developer Works, pp. 1–8, March 2006

McGovern, J., Sims, O., Jain, A., Little, M.: Enterprise Service Oriented Architectures: Concepts, Challenges, Recommendations (2006)

Niemann, K.D.: Enterprise Architecture at Work: Modelling, Communication, and Analysis. Springer, Heidelberg (2006)

Oxford. Oxford Dictionaries (2014)

Winter, R., Fischer, R.: Essential layers, artifacts, and dependencies of enterprise architecture. In: Proceedings - 2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops, EDOCW2006, pp. 1–12, May 2006

Zachman, J.: A framework for information systems architecture. IBM Syst. J. **26**(3), 276–292 (1987)