Architecting the Enterprise to Leverage a Confluence of Emerging Technologies

Eric Yu and Alexei Lapouchnian
University of Toronto
Toronto, Ontario, Canada M5S 3G6
{eric, alexei}@cs.toronto.edu

ABSTRACT

The simultaneous widespread and rapid adoption of a number of emerging technologies including low-cost sensing, mobile devices, social and cloud computing, and especially big data analytics is leading up to a perfect storm which will transform our conception of society and organizations. When acting in isolation, each of these technologies is already radically changing our lives and reshaping many industries. When deployed together as elements of a coherent strategy, the impact could be even more far-reaching. Yet, there is no conceptual framework for developing such strategies. In this paper, we explore some avenues for formulating such a framework.

Keywords

Adaptive enterprise, enterprise architecture, business intelligence, analytics, emerging technologies

1. INTRODUCTION

A number of emerging technologies including low-cost sensing, wireless and mobile devices, social and cloud computing platforms, big data analytics and others have seen widespread adoption in recent years and are triggering fundamental transformations in the information technology (IT) industry. While each of these diverse technologies offers unique functionalities and strengths when used in isolation, the potential for a coherent, synergistic deployment of these technologies will have even greater impact and significance. Yet there is no conceptual framework for developing such coherent strategies.

In information and systems sciences, it is well recognized that appropriate abstractions are crucial – not only for formulating and creating innovative solutions, but more importantly for conceptualizing and characterizing the problem domain.

The emergence of a whole new crop of diverse but potentially synergistic technologies offers an opportune moment to seek an appropriate level of abstraction to understand the implications (technological, social, etc.) of their introduction into systems and to shape their further development. As these technologies move beyond the low-hanging fruits of early-stage adoption and "niche" applications, we risk ending up with a patchwork of fragmented solutions. For example, there may be many sensor networks and social media channels feeding into business intelligence (BI) solutions in different pockets of an organization, each with its own objectives, sensing and monitoring their separate silos of activities. While niche solutions can be effective short-term and work well for early adopters, they tend to focus on local optima and fail to recognize enterprise-wide priorities and concerns. What conceptual abstractions are suitable for incorporating and exploiting the complementary capabilities of this diverse set of technologies? As with other areas in computing and software, we need to step back from the concrete and tangible to arrive at abstractions. From the point of view of a user organization, we want a framework that can help formulate a coherent enterprise-level vision of how an assortment of technologies can complement each other to achieve some overall objectives. From the viewpoint of technology development, we want a framework to guide the advancement of individual technologies to serve as building blocks for enterprise strategies.

The purpose of this position paper is to motivate the need for and propose the features of a technology-independent goal-driven conceptual framework that enables modeling the effects of implementation of new technologies on operational, tactical and strategic objectives of the enterprise as well as their influence on the organizational/social environment of the company. To this end, we envision a framework that supports explicit representation of objectives and the systematic exploration of options for achieving them, allows for making trade-offs, doing qualitative analysis and simulation, capturing design objectives such as responsiveness, agility, adaptiveness, compliance, etc., and specifying performance targets, measuring them and subsequently evolving/adapting system/organizational behaviour.

The paper is structured as follows. In Section 2, we focus on some observations on recent IT advances. Section 3 outlines the envisioned conceptual framework. Section 4 illustrates the potential utility of the framework on an example, while Section 5 provides discussion and conclusions.

2. MOMENTOUS SHIFTS

We believe the confluence of the current slate of emerging technologies will lead to dramatic shifts in the role of data and software in organizations.

1) Organizations will take advantage of these technologies to dramatically expand their capabilities for sensing and interpreting their environments.
2) The sensing and interpretation capabilities will eventually be coupled with already well-developed execution technologies to achieve responsiveness and timely adaptation.
3) As sensing and interpretation become highly automated and coupled with execution systems, more and more of an organization's sense-and-respond loops will shift from human scale to machine scale – in time, volume, and scope. In software systems, current divisions between design-time vs. run-time activities will be rethought.
4) Software (systems, architecture, run-time behaviour, and so forth) can no longer be conceived separately from software processes (typically labour-intensive knowledge work) since software artifacts are continuously being transformed by software processes (design, development, maintenance, and evolution) as elements within some ongoing sense-and-respond loops.

3. FRAMEWORK OUTLINE

Current frameworks for conceptualizing software and information systems are ill-equipped to handle such emerging systems. Current IT system design methods primarily aim at producing run-time execution behaviour that is pre-determined at design time.

Systems that play a sense-and-interpret role, such as BI, analytics, and decision support systems, are not recognized as such and therefore not conceptually or materially connected to the processes that their resulting insights inform and eventually direct. System architecture and behaviour are conceived, modeled, and analyzed completely separately from the human-enacted "software processes" (including design and requirements decision making) that produce them. Even as the confluence of emerging technologies enables and suggests moving some sense-interpret-decide-act loop from design-time to run-time, and from human scale to machine scale, there is no good framework to help conceptualize and analyze such reconfigurations of information artifacts (including software and data) together with a remix of human and automated processes.

Such modeling and analysis would belong most naturally at the level of "enterprise architecture", yet even the latest modeling languages such as Archimate 2.0 [1] have no provision for these considerations. To support the conceptual modeling and analysis of an enterprise architecture that takes advantage of the combined powers of emerging technologies, we propose that a modeling framework should include the following concepts and capabilities:

- An enterprise consists of a multitude of dynamic, adaptive processes loosely coupled with each other (some of them controlling or designing others), operating on many different time scales and scopes, and with different rates of change.
- Many of these processes are software-enabled and will increasingly move towards machine-scale, near-real-time adaptiveness where applicable sensory data and analytics are available.
- The framework allows for modeling adaptive change including the sensing, interpreting, decision making, and action elements of adaptive loops.
- Design-time and run-time activities (including decision making) are represented uniformly in the same model, so that the choice between performing an activity at run-time versus at design-time can be represented and reasoned about.
- The output of some process can be a design of another process, allowing some process to specify and create other processes.
- Software system rigidities will continue to be a barrier to smooth adaptiveness, as significant modifications to software systems will require development effort and capital investment. The framework will support modeling and analysis of such rigidities. Dynamic configurability and flexible provisioning (such as provided by cloud and service-oriented architectures) will therefore be available as design parameters in architecting the adaptive enterprise.
- The framework has the ability to represent the capabilities of emerging technologies, to match them with business needs, and to evaluate the impact of those technologies on the organization along the dimensions of cost, performance, flexibility, agility, maintainability, security, etc.

4. ILLUSTRATIVE EXAMPLE

The domain of supply chain management (SCM) is undergoing large-scale transformations, which can serve to illustrate the technological shifts outlined in Section 2 and the conceptual framework proposed in Section 3. Continuing trends in globalization, outsourcing, and trans-national strategic alliances are underscoring the growing importance of SCM, while intensified competition pressures enterprises into achieving supply chain reliability, responsiveness, and flexibility at a reasonable cost.

A modern supply chain includes many players such as raw material suppliers, component suppliers, and logistics, transportation, and distribution firms. It is a self-organizing network of companies that cooperate and collaborate to deliver products and services – a form of extended enterprise. Such a distributed network allows companies to focus on their core competencies, while assembling best-in-class partners to create high-performance value chains. Supported by the availability of pervasive Internet connectivity and affordable sensors and mobile communication devices, these supply networks can be created on demand and customized for specific products, regions, etc. SCM is getting more complex as companies stretch their supply chains and expand their geography, often resulting in reduced control. Price fluctuations, shorter product lifecycles, offshoring, nearshoring, and so on add to the complexity.

Sensing and interpretation technologies are increasingly being incorporated into SCM. Data analytics are used to monitor performance and decision making within particular components of the supply network as well as the overall supply chain in the extended enterprise. On the demand side, analyses of past sales and of estimated advertising campaign effects (including those in social networks) allow companies to project demand and thus to reduce or eliminate lead time, e.g., on customer orders. Real-time shipment monitoring using network-connected sensors not only offers location data, but also can measure temperature, humidity, light exposure and other parameters. The sensor networks provide a clear view into the supply chain and allow data to be automatically collected, analyzed and correlated. As sensor costs drop, shipments are being monitored at ever finer granularities – from containers, to truckloads, to pallets, and to boxes. Data volume multiplies rapidly as a result. This progression in the precision of monitoring exemplifies one of the shifts referred to in Section 2. The availability of data at each of the supply chain's components at many levels of granularity allows closed-loop sensing and acting at any of these levels and scopes. At each such level and scope, performance can be evaluated and anomalies can be detected and handled by sending alerts, finding alternative suppliers, shipping or other appropriate companies. This is becoming easier to do with the help of on-demand and anything-as-a-service offerings available in SCM. Moreover, services like FedEx's SenseAware offer transparency to customers by permitting them to track shipment progress nearly continuously [2]. With the data collected over a longer timeframe and over many process instances, the effectiveness of the current supply chain configuration can be determined and a redesign may be recommended. The above discussion demonstrates that both sensing and interpreting capabilities within the domain have dramatically increased with the help of a number of emerging technologies, illustrating several of the previously described technological shifts.

In realistic SCM scenarios, we see a complex structure of interrelated processes that differ in scope (e.g., warehouse management, physical distribution, the whole supply chain) and cycle time – e.g., the delivery of particular shipments of produce (such as peaches or tomatoes) by a mid-sized distributor (let us call this process *A*), the process of monitoring, analyzing, and improving the effectiveness/efficiency of the current process for sourcing and delivering produce (process *B*), and the process aimed at improving the supply chain across different products handled by that distributor (process *C*). Among these processes, there can be a number of different relationships. For instance, a higher-level process can control a lower-level one by adjusting that process' parameters. On the other hand, there can be a design-execute relation between processes, where a higher-level process designs or redesigns a specification for the lower-level one. We give examples of these relationships later in our discussion. The envisioned conceptual framework will be able to represent the scope, timeframe, and rate of change for each process and the above-described relationships among them.

Each process in the supply chain is operating within its own context and boundary. This context is set up within other, higher-level processes. For example, *A* – the process of supplying produce – operates in the context that was setup for it by process *B*, which controls *A* and is responsible for ensuring that *A* provides the best produce delivery method. Process *A* may support some variability in the way goods are sourced or delivered – e.g., to account for equipment breakdowns and weather emergencies or to be customized for particular products and/or customers – thus permitting some adaptation. The produce delivery process *A*, including the variability and flexibility that can be accommodated within it, and therefore its capacity to adapt, are designed by (i.e., are the output of) process *B*. Process *B* analyzes thousands of instances of *A* and, from time to time, redesigns process *A* to further improve the produce supply methods, taking advantage of supply chain advances. These redesigns themselves are done within the boundaries and constraints set up by process *C*, which is aimed at improving the overall supply chain for the company. Process *C* monitors not only *B*, but also similar processes for supplying other types of goods and controls them based on the their performance, market conditions, amount of resources available, existing priorities within the supply chain, etc. For example, suppose a new way of supplying produce that offers significant benefits becomes available. The new way requires advanced analytical capabilities to predict demand and to correlate it with crop forecasts for different produce growing regions as well as the ability to track shipments in real time. Depending on the current performance of the produce supply process (i.e., the aggregate metrics measured by process *B*), the available resources within the company, and the priority given to this area of the supply chain, process *C* may enable *B* to design a new version of *A* that utilizes the new produce supply method. As described in Section 3, the proposed framework should be able to capture the design relationship between these processes.

Why do we need to go several levels up to process *C* to make changes to process *A*? Is it not possible to effect this change entirely within process *A*? Why do we not simply let the manager responsible for a particular supply of peaches adopt the new scheme that promises better results for the company and its clients? The reason resides in the existence of barriers to change that can be resource or skills-based, organizational, legal, or, quite frequently, due to rigidities in the deployed software systems. The new supply method requires the distributor to obtain network-connected sensors and expensive advanced analytical capabilities. This incurs time and monetary investments and learning efforts, which are not feasible at the instance level of process *A* – i.e., at runtime. Moreover, due to organizational inflexibilities, a process redesign requires an approval that is likely unavailable at runtime.

Recognizing the problem of barriers to change, one can deliberately build flexibility into an organization and its IT systems to reduce its effects. Many new, emerging technologies and business model innovations aim to provide this flexibility. In SCM, mobile Internet, networks of affordable sensors with GPS, temperature, and other capabilities ensure continuous visibility into the movements of goods; cloud-based analytics drastically reduce the cost of doing data analysis (and thus performance monitoring and forecasting); and web-/cloud-based collaboration platforms allow to dynamically recruit supply chain partners to reconfigure the network on demand in order to improve it or to recover from failures. In the context of our produce supply example, emerging technologies will greatly reduce the barriers to change. For instance, the ability to do analytics in the cloud and to easily mobilize partners when necessary enables the distributor to pay for resources/capabilities as needed instead of acquiring or developing them outright. With the added flexibility enabled by these technologies, many changes to the supply chain processes can now be implemented without an approval from the high-level process *C* or can even be enacted entirely at runtime, within the instances of process *A*, at machine speeds. Moreover, the utilization of the existing company resources can be improved through making excess capacity available as a service to others.

The software system(s) supporting process *A* will be constantly changing in sync with it, continuously evolving to accommodate new services, etc. Feedback and adaptation occurs at all levels. No process is rigid – each one is monitored, analyzed, and adapted when needed. Enabled by the underlying flexible and adaptive infrastructure and/or other software systems and in turn enabling the agility in the business processes/systems that rely on it, software as an artifact now needs to be considered as part of a larger evolving/adaptive system. Compared to the current conception of software as an end-product, the proposed framework views software as more fluid and more data-driven and continuously being (re)configured and (re)designed by software processes that are themselves part of some feedback loops.

5. DISCUSSION AND RELATED WORK

Our approach builds upon a goal-oriented approach to requirements engineering (GORE) and social modeling [19]. Recent work has extended GORE to the enterprise level and incorporated strategic business reasoning for business intelligence [7]. The present work addresses complex dynamics involving multiple levels of system design and change [20] and extends earlier discussions [21][22].

Self-adaptive software systems is the area that has seen a great deal of recent work on conceptualizing adaptation using feedback loops and cycles [3][11]. Some approaches for adaptive systems design are requirements-driven [13][14]. That research discusses adaptation within software systems. Our focus in this paper is wider – it is on the adaptation and evolution that result from and are enabled by the introduction of emergent technologies into enterprises, thus affecting socio-technical aspects of adaptive systems.

With pressures growing on enterprises to adapt to rapidly changing environments, there is growing interest in adaptive enterprises and Enterprise Architectures (EAs). Adaptive enterprises are capable of gaining insights into their environment as well as internal operations, quickly reacting to external or internal changes, addressing threats and exploiting opportunities. Handling change is a major concern of EA that has been advancing both as a field of study and in practice. The benefits of enterprise modeling are becoming more widely recognized, as evidenced by the progress in standards (e.g., [1]). However, current enterprise models focus mainly on expressing a single architectural state of the enterprise (e.g., the as-is or to-be states). Current methods provide guidance on large-scale architectural transformations (e.g., the ADM in TOGAF [17]), from an as-is to a to-be architecture state, but do not deal with the full range of enterprise dynamics, such as instance-level adaptations occurring frequently and on much shorter timeframe. Runtime dynamics is usually modeled as business processes, with limited attention to variability and diversity. Current models are therefore too static and restrictive for architecting today's enterprises that must function in a fast-moving and highly uncertain world. Still, proposals exist for adaptive enterprise architectures, including one from Wilkinson [18], who proposes a particular method for designing an adaptive EA by utilizing the adaptability of a service-oriented computing infrastructure

There is extensive work on agility and adaptiveness in enterprises. This includes proposals and studies on strategies and tactics for adaptive enterprises. For instance, Haeckel [7] advocates for a sense-and-respond approach to adaptive enterprise. Lee [12] argues that supply chains should strive to be agile, adaptive, and aligned, and proposes methods to achieve each characteristic in the supply chain domain – e.g., deferring decisions to take advantage of latest market data, as practiced by apparel companies such as H&M and Zara.

Our vision of software as an artifact being continuously redesigned by the accompanying software design processes is supported by Garud et al. [6]. As argued by them, in traditional design settings, there is a stable boundary between the entity being designed and the context for which it is being designed. In this context, it is possible to fix the design purpose based on a stable set of requirements and user preferences. However, in the environments characterized by continuous change, "problems are ill-defined, preferences are fluid and solutions emerge in action" and design goals remain a constantly moving target. Here, a pragmatic approach would embrace design as both the process and the outcome, where each outcome is the beginning of a new design process.

The current confluence of emerging technologies invites a rethinking of the role of data, software, and software processes in organizations. A number of systems theory approaches appear to be relevant. These include control systems theory [8], system dynamics [5][15], complex adaptive systems [4], dynamic capabilities [16], and socio-ecology [9]. We are exploring this literature to enrich a conceptual framework for adaptive enterprise architecture.

6. REFERENCES

[1] ArchiMate 2.0 Speficication. 2012. http://pubs.opengroup.org/architecture/archimate2-doc/

[2] Bughin, J., Chui, M., and Manyika, J. 2013. Ten IT-enabled business trends for the decade ahead. *McKinsey Quarterly*, May 2013.

[3] Cheng, B., de Lemos, R., Giese, H., Inverardi, P., Magee, J. et al. 2009. Software engineering for self-adaptive systems: a research roadmap. In *Software Engineering for Self-Adaptive Systems*, LNCS Vol. 5525, pp. 1–26, Springer-Verlag, Berlin Heidelberg.

[4] Dooley, K. 1997. A complex adaptive systems model of organizational change. *Nonlinear Dynamics, Psychology, and Life Sciences*, 1(1), pp. 69–97.

[5] Forrester, J. System Dynamics – The Next Fifty Years. *System Dynamics Review*, 23(2/3), Summer/Fall 2007, Wiley.

[6] Garud, R., Jain, S., and Tuertscher, P. Incomplete by Design and Designing for Incompleteness. 2009. In *Design Requirements Engineering: A Ten-Year Perspective*, Lyytinen, K., Loucopoulos, P., Mylopoulos, J., and Robinson, B. Eds. Springer Berlin Heidelberg, pp. 137–156.

[7] Haeckel, S.H. 1999. Adaptive Enterprise: Creating and Leading Sense-And-Respond Organizations. Harvard Business Press.

[8] Hellerstein, J., Diao, Y., Parekh, S. and Tilbury, D. 2004. Feedback Control of Computing Systems. John Wiley & Sons.

[9] Holling, C. S. 2001. Understanding the Complexity of Economic, Ecological, and Social Systems. *Ecosystems*, 4, pp. 390–405. Springer.

[10] Horkoff, J., Barone, D., Jiang, L., Yu, E., Amyot, D., Borgida, A., Mylopoulos, J. 2012. Strategic Business Modeling: Representation and Reasoning. *Software and Systems Modeling (SoSym)*, October 2012. Springer.

[11] Kephart, J. and Chess, D. 2003. The vision of autonomic computing. *IEEE Computer*, 36(1), pp. 41–50.

[12] Lee, H.L. 2004. The Triple-A Supply Chain. *Harvard Business Review*, pp. 102–112, October 2004.

[13] Sawyer, P., Bencomo, N., Whittle, J., Letier, E. and Finkelstein, A. 2010. Requirements-aware systems: a research agenda for requirements engineering for self-adaptive systems. *IEEE Int. Conf. Requirements Eng.*, pp. 95–103.

[14] Souza, V., Lapouchnian, A. and Mylopoulos, J. 2012. Evolution requirements for adaptive systems. *International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2012)*, Zurich, Switzerland.

[15] Sterman, J.D. 2000. Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw-Hill.

[16] Teece, D., Pisano, G. and Shuen, A. 1997. Dynamic capabilities and strategic management. *Strategic Management Journal*, 18(7), pp. 509–533.

[17] TOGAF Version 9.1. 2011. http://pubs.opengroup.org/architecture/togaf9-doc/arch/

[18] Wilkinson, M. 2006. Designing an 'Adaptive' Enterprise Architecture. *BT Technology Journal*, 24(4), pp. 81–92.

[19] Yu, E. 2009. Social Modeling and i*. In Borgida et al, Eds. Conceptual Modeling – Foundations and Applications, Springer.

[20] Yu, E., Lapouchnian, A., Deng, S. 2013. Adapting to Uncertain and Evolving Enterprise Requirements: The case of business-driven business intelligence. In Proc. 7th IEEE International Conference on Research Challenges in Information Science (RCIS 2013), Paris, France, May 29-31, 2013.

[21] Yu, E., Deng, S, Sasmal, D. 2012. Enterprise architecture for the adaptive enterprise - a vision paper. Workshop on Trends in Enterprise Architecture Research, Open Group Conference, Barcelona, Spain, October 2012.

[22] Yu, E. 2013. Seeking conceptual foundations for rethinking IS curriculum to incorporate Big Data and Analytics. Big Data and Analytics Educational Conference (BDA EdCon), Las Vegas, NV, Nov 2-3, 2013.