Design Dimensions for Business Process Architecture

Alexei Lapouchnian[1], Eric Yu[1], Arnon Sturm[2]

[1]University of Toronto, Canada; [2]Ben-Gurion University of the Negev, Israel
alexei@cs.toronto.edu, eric.yu@utoronto.ca, sturm@bgu.ac.il

Abstract. Enterprises employ an array of business processes (BPs) for their operational, supporting, and managerial activities. When BPs are designed to work together to achieve organizational objectives, we refer to them and the relationships among them as the business process architecture (BPA). While substantial efforts have been devoted to designing and analyzing individual BPs, there is little focus on BPAs. As organizations are undergoing changes at many levels, at different times, and at different rates, the BP architect needs to consider how to manage the relationships among multiple BPs. We propose a modeling framework for designing BPAs with a focus on representing and analyzing architectural choices along several design dimensions aimed at achieving various design objectives, such as flexibility, cost, and efficiency.
Keywords: business process architecture, goal modeling, requirements

1 Introduction

Organizations rely on multitude of business processes (BPs) for their everyday functioning as well as for longer term viability and sustainability. These processes need to work together coherently, constituting the *process architecture* of the organization.

BP analysis and design have garnered a lot of attention, unlike the development of business process architectures (BPAs), which is about how BPs should relate to each other to better serve their organization’s objectives. Given the vast changes faced by today’s enterprises due to business/technological innovations, BPs can no longer be optimized individually. Rather, the onus should be on BPAs, which cannot remain static and need to be re-designed to account for changes inside and outside of organizations.

BPA development involves making trade-offs across processes, particularly regarding how to balance flexibility and agility with other design objectives such as costs, efficiency, and so on. Existing conceptions of BPA (e.g., [3], enterprise architecture (EA) frameworks [15], etc.) mostly see BPAs as a given, or as something to be discovered, with no effort to shape or evolve them. We, however, see the need to explore BPA alternatives and to support the selection among them, while taking complex trade-offs into consideration. An appropriate modeling notation capable of addressing these requirements is needed. This paper proposes temporal and recurrence dimensions that,

adfa, p. 1, 2011.

we believe, serve as key elements of such a framework. We chose the public transportation domain, which is rich, diverse, and easy to understand to illustrate our approach.

This paper is organized as follows. Section 2 outlines the architectural design space and Section 3 discusses two dimensions of that space. Section 4 talks about the analysis of BPAs. Then, Section 5 covers the related works and Section 6 concludes the paper.

2 Variability Space for Business Process Architectures

In a framework aimed for BPA design, we have to represent a space of BPA alternatives and thus require a modeling notation capable of this and of reasoning about the identified choices. We propose two BPA design dimensions to structure the space of architectural alternatives: (1) *temporal* is about placing a process element (PE) – an activity or a decision – earlier or later relative to other PEs within a BP; (2) *recurrence* is about placing a PE into a BP that is executed more (or less) frequently relative to other BPs. Then, changes to BPAs can be seen as movements along these dimensions.

When determining the positioning of PEs along these dimensions, we look at the trade-off between flexibility and efficiency, a major concern in BPA design. How this tension is resolved is domain- and organization-specific. For a static business domain, BPs in a BPA can be tightly coupled and globally optimized. Instead, in a dynamic domain, BPAs have to provide flexibility to support organizational agility. This flexibility requires that organizations keep their options open. This may incur costs, decrease performance, and increase complexity and hence leads to the above-mentioned trade-off. Thus, the BPA design space model needs to represent where/what options exist and provide ways to analyze the choices w.r.t. important quality objectives (NFRs).

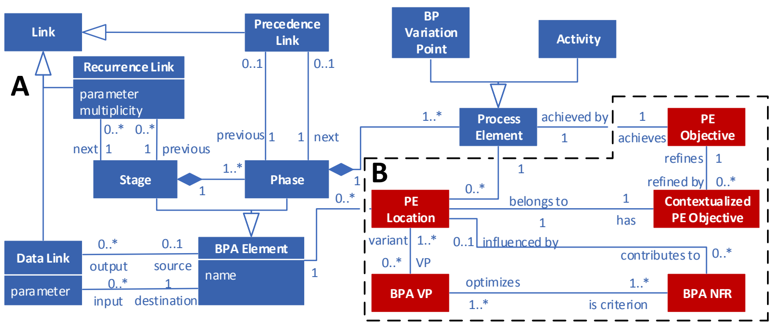

Fig. 1. The metamodel for the approach combines (A) BPA modeling and (B) goal modeling.

These two dimensions were determined based on existing studies (see Section 5), our own experience with BPAs, and the analysis of existing BPAs and their potential changes. We are not claiming that no other dimensions exist. In fact, here, we elaborate only on the core portion of our research on designing adaptive BPAs. Further design dimensions for evolving BPAs are proposed in [8,9]. In addition, ideas on utilizing emergent technologies for system flexibility with the help of BPAs are discussed in [7].

The architectural description we are targeting should outline the major BPA elements and relationships while avoiding over-specification. Fig. 1 presents a metamodel for our modeling notation where we combine BP modeling and goal modeling. In Section 3, we discuss the modeling of BPAs (the "A" part of the metamodel), while Section 4 talks about using a goal-based notation (the "B" part) for comparing them.

3 Dimensions for Designing Process Architectures

3.1 The Temporal Dimension

There may be multiple possible placements for PEs along the temporal dimension in a BP that achieve the same functional objective and comply with the existing functional dependencies, but differ in their non-functional characteristics. These choices need to be resolved by looking at how each variant affects the quality criteria that the organization is interested in. Trip payment options for passengers in Fig. 2A illustrate this: unlike the standard fare charged before a trip, a distance-based fare can be charged after one. Both obtain the payment, but differ in the charged amount, payment fairness, etc.

Definitions. Determining what is better, to charge the customer before a trip, during the trip or after travel depends on one's point of view. Going for payment fairness, we want taxi-like distance-based fares. Here, the system needs a richer context (the traveled distance) only available after travel. So, payment on exit is a better option for fairness. We identify *phases* – portions of a BP such that placing a PE anywhere within them is the same in terms of the functional and non-functional results. Moving PEs *across* phase boundaries may affect action outcomes and decision quality. Unlike most software variability approaches, we analyze in which phase (i.e., when) it is best to place PEs. Note that an output of a phase of some BP instance can only be used by the subsequent phases of the same instance. No reuse happens. Phases help reduce analysis complexity by abstracting from low-level details that do not affect BPA-level concerns.

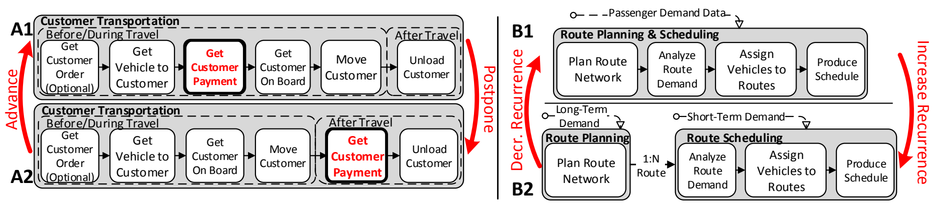

Fig. 2. A: Trip payment options – before (A1) or at (A2) the end of a trip.
B: Splitting a single stage into two (B2) or merging two stages into one (B1).

Postponement. We look at postponing or advancing PEs by placing them in later or earlier phases respectively (see Fig. 2A). *Postponement* is a well-known business strategy (e.g., [11]) that delays activities/decisions that require up-to-date information while expecting more precise data to be available at a later point in a BP. This produces better, more context-sensitive outcomes. Instead, *advancement* supports stability/uniformity and is enabled either by PEs that rely on less information and thus have more tolerance for uncertainty or by predicting the missing information. Availability of data required by the postponed PEs and data collection/analysis costs are important concerns here. E.g., for distance-based fares, the system needs to obtain the traveled distance.

3.2 The Recurrence Dimension

Unlike the previous section where PEs were executed for every process instance, the recurrence dimension is about reusing PE outcomes in multiple BP instances: how often should certain decisions or actions be (re)executed and under what conditions?

Definitions. We group PEs with the same execution frequency into process portions called *stages* that contain one or more phases (e.g., in Fig. 2A, Customer Transportation is a stage with two phases). Once a stage executes, its output is available to the subsequent stages (if any) until it is re-executed. In our models, stages are connected with control flow links labeled with "1:N" to indicate the relationship cardinality (see Fig. 2B). A *stage boundary* between two stages points to the two options for placing PEs. Moving a PE across this boundary can lead to a major change in the PE's execution frequency. Fig. 2B shows this change in both directions for Route Scheduling. A PE can also be moved across more than one stage. A stage represents a (sub-)process and while the temporal dimension was about intra-process analysis, here the focus is on inter-process relationships – relative rates of execution/change cycles among processes.

Using the dimension. We identify PEs that can be reused for multiple BP instances (i.e., that are independent of the variations in those instances), at least for a period of time, to arrive at a stage-based configuration. Such PEs form a stage with its output available to be reused by the subsequent stages, thus saving time, money and/or other resources and perhaps affecting other NFRs. E.g., buying a transit pass eliminates the need to pay for each trip separately and hence improves convenience. Another heuristic is to identify PEs with the same execution frequency (e.g., yearly product redesigns accompanied by the product manual updates) or are triggered by the same data-driven trigger (e.g., a passenger demand change forces a bus schedule revision and the subsequent publication of the new schedule). Splitting a stage into two or merging two stages into one (see Fig. 2B) is another type of change in this dimension.

Domain example. One way to perform public transit route network planning and scheduling is to combine both into a single stage (Fig. 2B1). Whenever routes need to be changed (e.g., due to major demand changes), Route Planning & Scheduling stage is triggered. The stage's data input (the message flow at the top) has all the available passenger demand data. Both route planning and scheduling are bundled together and have the same change cycle, meaning that changing schedules without a route network redesign is not possible. For a predictable constant demand this will work well. However, the rigidity will hurt the company's ability to change its schedules more often than its routes given an evolving passenger demand. As a solution, we can unbundle the two stages as shown in Fig. 2B2, creating Route Planning and Route Scheduling, which are triggered when long-term or short-term passenger demand changes, respectively. Then, the route network (the outcome of the Route Planning stage) can be reused for multiple Route Scheduling instances (again, note the "1:N" annotation) supporting frequent rescheduling on the same route and thus enabling more flexibility, but likely having higher cost, complexity, etc. Given the choices along the recurrence dimension, an organization needs to pick the one that best fits its needs and preferences, while reflecting its business domain volatility.

Fig. 3 presents a BPA fragment for a transit company showing the relationships among stages (we abstract from phases here) and implementing the more flexible options from the previously discussed examples. Data inputs for a stage come from other stages (see Required Capacity) or from the outside – e.g., through monitoring (see Social Network Data). A BPA model gives an overview of the organization's BPs and points to where reuse happens. Fig. 3 shows a *possible* BPA configuration. Others are obtained by moving PEs among stages/phases. The starting point for developing a BPA using our notation is an existing BPA in another notation or a collection of BPs. We are working on a goal-driven approach for designing the initial BPA using ideas from [5] that would refine organizational goals into objectives to be achieved by individual PEs.

The next section elaborates on modeling and analysis of BPA configuration choices.

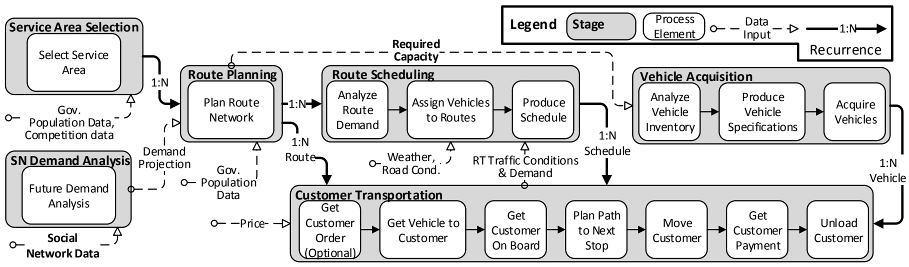

Fig. 3. A fragment of a business process architecture for a public transit company.

4 Analyzing Process Architecture Alternatives

We use a goal model (GM) variant to represent and analyze BPA choices along the two dimensions (see Fig. 1B for its metamodel). GMs are used in Requirements Engineering to capture stakeholder/system objectives. They can represent variability in achieving goals using OR decompositions (exclusive ORs in our case) and capture NFRs to analyze goal refinement choices. We adapted GMs to represent possible PE placements in BPAs and to evaluate them against the relevant NFRs. A GM (see Fig. 4) focuses on a single PE, so multiple GMs will be used to analyze various portions of a BPA.

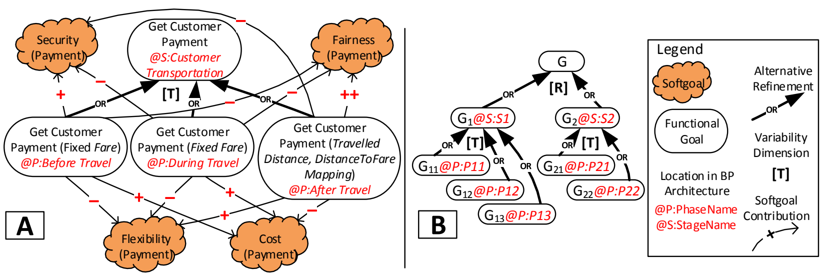

Fig. 4. A: Analyzing the temporal placement of customer payment PE.
B: A generic goal model for positioning a PE with the objective G in a BPA.

Identifying BPA choices. To show how a GM is developed and used, we look at the trip payment options discussed in Section 3.1. A GM provides the intentional (vs. the operational) view, has the goal of the PE in question as its root node (see the top Get Customer Payment node in Fig. 4A), and shows how it can be refined when placed into different BPA stages/phases. E.g., Fig. 4A shows the three phases of Customer Transportation stage where Get Customer Payment PE can be placed. The @P:PhaseName and @S:StageName annotations refer to these locations. Each goal refinement looks at the PE placement options along one dimension ([T] for temporal and [R] for recurrence, as in Fig. 4). The refinement choices for the temporal dimension are the stages where a PE can be placed (while respecting the existing functional constraints). The phases within a *previously selected* stage are the choices for the temporal dimension – e.g., in Fig. 4A, customers can pay before (@P:Before Travel), during or after travel. This leads to the common pattern of analysis where [T] follows [R] (see Fig. 4B).

Varying parameters in goal nodes capturing alternative PE placements represent the possibly different implementations of a PE in various places in a BPA. Implementations depend on the available data, etc. E.g., the traveled distance is only known after the trip. Later stages/phases usually have higher contextualization and thus more parameters.

Using NFRs. NFRs for evaluating PE placement choices must be elicited. They are modeled as softgoals and evaluated using *contribution links* (e.g., [5]). The evaluation can be qualitative, with values such as help(+)/hurt(–), make(++)/break(--) (as in Fig. 4A). A softgoal is *satisficed* if there is more positive than negative evidence for this claim. Given extra information, a more precise evaluation is possible. Softgoal prioritization can help in case of conflicting NFRs that cannot be satisficed at the same time.

Handling trade-offs. Moving PEs along the two dimensions affects many NFRs. Some NFRs are domain- and PE-independent. The effects of moving PEs on typical NFRs are shown in Table 1 (cost/flexibility also feature in Fig. 4A). Others are domain- or PE-specific – e.g., security and fairness for fare payment (Fig. 4A). The resulting trade-offs are resolved based on two things: 1) domain dynamics (what changes, how frequently, etc.); 2) the organization's ranking of the above-mentioned quality criteria.

Once the analysis of BPA alternatives for a PE is done, a place (stage/phase) in the BPA is identified for it. It can be seen as the delta between the as-is and to-be BPAs.

Dimension	PE Movement	Effect of Movement on NFRs
Temporal	Postpone	+: flexibility, context-awareness; –: cost, complexity, stability
	Advance	+: cost, complexity, stability; –: flexibility, context-awareness
Recurrence	Increase Recurrence	+: flexibility; –: cost, reuse, stability
	Decrease Recurrence	+: cost, reuse, stability; –: flexibility

Table 1. Effects of moving PEs along the variability dimensions.

Reconfiguring BPAs. Our method's objective is to help organizations find the right balance between flexibility and stability given their preferences/priorities and the level of change in their business domains. Change can have two types of effect on a BPA. Once designed, a BPA has some amount of flexibility to support certain types of domain changes (e.g., varying trip distances are supported by distance-based fare payments). If flexibility in a BPA can accommodate the change (i.e., the changing trip lengths), a BPA does not need a reconfiguration. However, if the assumption that passenger trip lengths are quite different (thus justifying the complexity/cost of distance-based fares) no longer holds, distance-based fares become *too flexible* for the domain and the BPA may need to be changed. Hence, a BPA is modified when the domain dynamics (the rate/range of change) is changes (not when *some* change happens).

5 Related Work

Over the years, BPAs were researched (e.g., [3]) and various sets of relationships among their BPs were identified [2,3], such as sequence, hierarchy, trigger, etc. In the EA area, BPAs are viewed as BP cooperation and ArchiMate [8] includes causal, mapping, realization, etc. relationships between BPs and services. Other relevant domains are BP variability [1], which focuses on the realization relationships among BPs and the binding time of such processes, and software product lines [12]. The ability to postpone design decisions to the latest feasible point is a major reason to support variability [12]. Advancing decisions in processes improves their efficiency [13]. In goal-based methods for BPs (e.g., [5]), the many ways of achieving business goals are utilized to develop customizable, adaptive, and evolving systems. Methods weaving requirements and BPs have also emerged. E.g., [14] uses NFRs and contexts for runtime BP configuration. In process-aware information systems, Weber et al. [16] identify four dimensions of change and point to the new BP relationships: *creation*, where a BP creates another one, and *recurrence*, where a BP is followed by another BP multiple times.

Overall, existing approaches focus on process-level variability while we look at the BPA-level variability and goal-based trade-off analysis (also missing from most other approaches). Also, for PEs that are decisions themselves, the phases/stages where they can be placed provide options for *when* (temporal dimension) and *how often* (recurrence dimension) to bind them. Hence, we obtain domain-specific variability binding options, which are much richer than those usually discussed in variability research (e.g., [4]).

6 Discussion & Conclusions

A good BPA must reflect its domain. Assumptions about the domain (e.g., its volatility and data availability within it) need to be analyzed to justify the flexibility in the BPA. Such assumptions are not yet captured in our approach. We are working on their formal modeling, which will help specify the conditions that trigger BPA adaptations.

The analysis in Section 4 favours local optimization (at the level of individual PEs), which is detrimental to the global optimality. We are developing ways to integrate multiple goal models, each similar to those in Fig. 4, to alleviate this. Also, identification of relevant data and its availability and volatility play a vital role in positioning PEs within a BPA and we are currently working on integrating data into this approach.

We support some model incompleteness to reduce modeling and analysis complexity and to focus on the important domain aspects. GMs only have to be utilized in volatile portions of the domain, where BPA changes might be needed. Some portions of the BPA can be seen as stable and thus not requiring an analysis. Also, most BP modeling details are below our threshold of interest, which is a phase. For analyzing BPA choices, we plan to utilize goal reasoning algorithms already successfully used in a variety of applications (e.g., in [5] for BP configuration).

In summary, we presented an approach for modeling/analysis of BPAs to help organizations be better aligned with their business domain dynamics and their desired level of flexibility. We introduced two BPA design dimensions to identify BPA choices and adopted GMs to analyze these w.r.t. important NFRs. We have identified a research agenda to further refine the approach and are applying the method in several domains that experience significant business/technological volatility in order to evaluate it.

References

1. V. Chakravarthy and E. Eide. Binding Time Flexibility for Managing Variability, In Proc. MVCDC 2 at OOPLSA 2005, San Diego, CA, USA, 2005.
2. R. Dijkman, I. Vanderfeesten, and H. Reijers. Business process architectures: overview, comparison and framework. Enterprise Information Systems, DOI: 10.1080/17517575.2014.928951.
3. M. Dumas, M. La Rosa, J. Mendling and H. Reijers. Fundamentals of Business Process Management, Ch.2. Springer-Verlag, Berlin-Heidelberg, 2013.
4. M. Galster, D. Weyns, D. Tofan, B. Michalik, and P. Avgeriou. Variability in Software Systems – A Systematic Literature Review. IEEE TSE, 40(3), pp. 282–306, 2014.
5. A. Lapouchnian, Y. Yu and J. Mylopoulos. Requirements-Driven Design and Configuration Management of Business Processes. In Proc. BPM 2007, Brisbane, Australia, 2007.
6. A. Lapouchnian, Y. Yu, S. Liaskos and J. Mylopoulos. Requirements-Driven Design of Autonomic Application Software. In Proc. CASCON 2006, Toronto, Canada, Oct 16-19, 2006.
7. A. Lapouchnian and E. Yu. Exploiting Emergent Technologies to Create Systems that Meet Shifting Expectations. In Proc. ET Track at CASCON 2014, Toronto, Canada, 2014.
8. A. Lapouchnian, E. Yu, A. Sturm. Re-Designing Process Architectures: Towards a Framework of Design Dimensions. In Proc. RCIS 2015, Athens, Greece, May 13-15, 2015.
9. A. Lapouchnian, E. Yu, A. Sturm. Re-Designing Process Architectures. Technical Report CSRG-625, University of Toronto, 2015. Available at: ftp://ftp.cs.toronto.edu/csrg-technical-reports/625
10. Open Group, The. ArchiMate 2.1 Specification, 2013. Retrieved from http://pubs.opengroup.org/architecture/archimate2-doc/
11. J. Pagh, and M. Cooper. Supply Chain Postponement and Speculation Strategies: How to Choose the Right Strategy. Journal of Business Logistics, 19(2):13-33. 1998.
12. M. Svahnberg, J. van Gurp and J. Bosch. A taxonomy of variability realization techniques: Research Articles, Software—Practice & Experience, v.35 n.8, p.705-754, July 2005.
13. S. Subramaniam, et al. Improving process models by discovering decision points, Information Systems, 32(7), 2007, pp. 1037–1055.
14. E. Santos, J. Pimentel, T. Pereira, K. Oliveira, J. Castro. Business Process Configuration with NFRs and Context-Awareness. ER@BR, 2013.
15. TOGAF Version 9.1. 2011. Retrieved from: http://pubs.opengroup.org/architecture/togaf9-doc/arch/
16. B. Weber, M. Reichert, and S. Rinderle-Ma. Change Patterns and Change Support Features – Enhancing Flexibility in Process-Aware Information Systems. Data and Knowledge Engineering 66(3):438–466, 2008.