Modeling Requirements for Value Configuration Design

Eng Chew, Igor Hawryszkiewycz and Michael Soanes.

Department of Information Technology, University of Technology, Sydney, Australia.
{Eng.Chew, Igor.Hawryszkiewycz, Michael.G.Soanes}@uts.edu.au

Abstract. Breadth and depth complexity are key challenges in achieving business process fusion as the enabler for value configuration design. The PARM framework is proposed as the requirement to address breadth and depth complexity through the independent but integrated operation of the process, activity, resource and management viewpoints. The operational scenarios for each viewpoint result in varying process modeling extension requirements. Existing process modeling constructs have varying support for these requirements. The PARM framework solution is an extension and integration of existing modeling constructs rather than a solution in its own right. Using the MDA approach of abstracting a platform independent model from a platform specific implementation, it is the goal in future papers to define process modeling extensions to support the PARM framework and map these into existing implementation architectures.

Keywords: business process design, value configuration design, process architecture.

1 Introduction

Porter[1] introduced the concept of the value chain as a series of activities that add value in contributing to the delivery of customer requirements. The value chain concept was later extended by Stabell and Fjeldstad[2] into value configuration, *defined as a network of value chains*. Value configuration denotes the fact that in practice, an enterprise commonly networks with several partners in servicing its customers. The value configuration models the enterprise-wide business process as a *network* of interdependent core processes. Designing individual core business processes in isolation, without the enterprise-wide view, can lead to a sub-optimal process design when aggregated into the total value network.

Dynamic reconfiguration of the value configuration is gaining momentum as a new competitive advantage. Gartner Group [3] has labeled this trend “business process fusion” and defines it as “the transformation of business activities that is achieved by integrating previously autonomous business processes to create a new scope of management capabilities.” Gartner Group [4] says that through a new operating and management focus on enterprise wide processes and technology integration, business process fusion will enable an enterprise to increase its agility and improve efficiency.

There is a recent trend to advocate the revitalization of business process management (BPM) as a key enabler for business process fusion. This paper (as part of a series of papers) proposes value configuration design, as the next evolution of business process design, is the enabler for business process fusion.

Value configuration design requires a *process engineering methodology* which ensures the resultant value configuration will deliver the customer value (requirements) in line with the business strategy. Our research aims to develop the methodology, which will serve as a practical process innovation tool for process managers.

The first paper [5] in the series introduced the breadth / depth complexity matrix as the challenge in addressing value configuration design. The second paper [6] introduced the Process / Activity / Resource / Management (PARM) framework as the requirements framework to address breadth / depth complexity. This paper expands the PARM framework into a set of requirements of each viewpoint of the PARM framework as input to identifying the process modeling constructs extensions required to address these requirements. The next paper will propose extensions to existing modeling languages to support the PARM framework and map them to proposed architectural implementations. The final paper will propose a process engineering methodology that leverages the PARM framework solution to achieve the objectives of value configuration design.

2 Breadth / Depth Complexity as the Challenge

Soanes [5] introduced the concept of the breadth / depth complexity matrix to describe the inadequacy of individual business process designs.
Breadth complexity is defined as the range of activity types within a business process ranging from highly structured systemic to unstructured ad-hoc activities.
Depth complexity is defined as the abstraction levels of process logic within a business process ranging from very coarse process logic (e.g. work passing from one resource to the next) to very granular process logic (e.g. navigation between fields on a data capture screen).
The combination of breadth and depth complexity results in the following matrix:

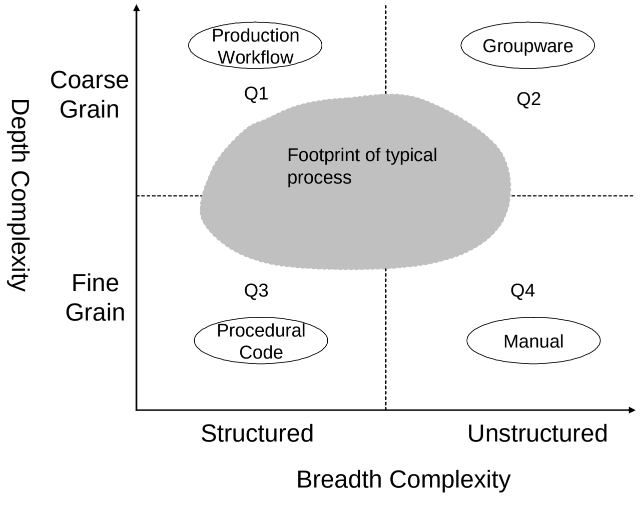

Fig. 1. Process Breadth / Depth Complexity Matrix

Soanes proposed that the footprint of typical processes crosses multiple breadth / depth quadrants of the above matrix. Soanes concluded that existing process design strategies and toolsets tend to specialise in one quadrant of the matrix. An example mapping of toolsets to quadrants is illustrated in Figure 1. Given individual processes can span multiple breadth/depth segments, this specialisation strategy can result in multiple process design strategies and toolsets being used within the one process. This fragmenting of business process logic across multiple toolsets results in a more complex task to maintain process logic as business requirements change – an obstacle to achieving business process fusion.

4 PARM Framework as the Requirement

Chew et al [6] introduced the PARM framework as the definition of the requirement for addressing breadth and depth complexity as per the following diagram:

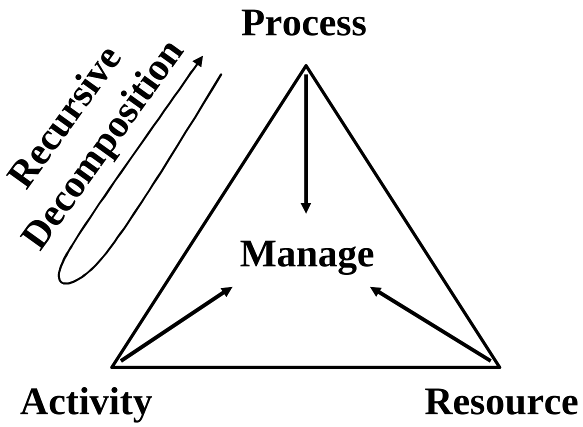

Fig. 2. Process / Activity / Resource / Management (PARM) framework.

The Process / Activity / Resource / Management (PARM) framework defines four viewpoints of business processing that need to be integrated and managed as part of the design considerations (in response to *stakeholder requirements*) for each core business process:

- The Process viewpoint focuses on the controlling, guiding and restricting of the flow of activities performed for specific process instances. Its measurable objective is to meet the *customer's* end to end service delivery expectations.
- The Activity viewpoint focuses on the facilitation of an environment to manage human activity with the recognition that human resources will prioritise their own

Appear in: A. ter Hofstede, B. Benatallah, and H.-Y. Paik (Eds.): BPM 2007 Workshops, LNCS 4928, pp. 169–178, 2008.
Springer-Verlag Berlin Heidelberg 2008

execution of multiple activities across multiple processes simultaneously based upon their own individual work practices. The execution sequence is not assumed to be deterministic – contrary to conventional process design which assumes that activities will be executed as prescribed by the design. Its measurable objective is to provide the most effective (both productivity and quality) environment for the completion of all work across all processes – reflecting the *process / knowledge worker's* cognitive decision making behaviour which is unstructured.

- The Resource viewpoint forecasts, plans, schedules and assigns resources to activities. Its measurable objective is to maximize the utilization and therefore the efficiency of the total resource pool. This viewpoint captures the *resource planner's* requirements.
- The Management viewpoint integrates the process, activity and resource viewpoints through balancing the tension between service, cost and quality expectations. It reflects the requirements of the *business owner* of the process.

Breadth complexity requirement is modeled by the alignment and integration of the process, activity and resource viewpoints, with the "breadth" being accentuated by the activity viewpoint which explicitly models both structured and unstructured behaviours.

The recursive decomposition of the framework parameters (an activity at one level of abstraction can be decomposed as a process at the next lower level of abstraction), enables breadth complexity to be managed at multiple levels of depth complexity.

Process design completeness is achieved through a consistent and integrated approach to modeling and managing the process, activity and resource viewpoints.

Chew et al's key observation was that existing process design strategies and BPMS toolsets assume that the process viewpoint is the driver of activities and resources. The PARM framework recognizes that in highly structured processes the process viewpoint can control the activity and resource viewpoints. However, the nature of unstructured processes implies that activity initiation and resource allocation can be initiated and controlled independently from the overall process.

In the following sections, the requirements of each viewpoint will be expanded.

4 Process Viewpoint Requirements

This reflects the scenario where the process viewpoint is the controller and is directly instantiating activity commencement and resource assignment. This scenario is particularly relevant to automated system processing but also reflects the traditional production line human centric business processing.

This is the traditional BPM viewpoint and thus no attempt is made in this paper to define the total requirements of BPM. The following are specific requirements relating to addressing the breadth / depth complexity challenge:

- Decomposition of the process design to multiple levels of abstraction is a critical enabler for depth complexity (a requirement that most BPMS address). There is some dispute whether decomposition process design is a good practice. For example Ould [7] in describing Riva as a process design methodology, specifically discourages decomposition. It is the view of this paper that decomposition is essential.
- Traditional process design will involve the use of highly structured graphical modeling constructs such as sequencing, concurrency branching, conditional branching etc. An additional requirement to address breadth complexity is the ability to define a set of activities with no sequencing control. By default this can be achieved with graphical constructs by a concurrency branch for each activity. However it by adding the capability for an activity to decompose at the next level to a set of declarative rules, provides a richer ability to define complex sequencing control. For the common scenario of wanting no sequencing control, the set of rules in effect become an action item list of actions that must be completed to achieve the activity been decomposed.
- The process viewpoint is the key communication viewpoint. To address different audiences, it is ideal to have the ability to produce graphical models at varying depth abstractions. Thus for senior management, the model will be filtered to a higher level whereas for process participants it may be at a detailed level. Ideally this depth filtering could also be applied to selective segments so that where appropriate some segments of the process could be shown in detail and others in summary.
- Where the process viewpoint is not the controller (e.g. a knowledge worker is performing activities based upon their own preferences), the process design needs to define wait states where the process awaits an external event (e.g. completion of an activity that the process was not in control of). Most modeling constructs already support this requirement through the use of defining states and then as a physical implementation of that state, tie the state to awaiting an event. Role Activity Diagrams for example provide the capability to define a state as well as event triggers.

5 Activity Viewpoint Requirements

This represents the scenario where the activity viewpoint is the controller whereby an individual resource based upon their own individual work practices, multi tasks across multiple activities across multiple processes utilising or interacting with multiple other resources. This scenario is particularly relevant to knowledge worker environments and exception handling within production processes.

The modeling requirements of this scenario are:

Appear in: A. ter Hofstede, B. Benatallah, and H.-Y. Paik (Eds.): BPM 2007 Workshops, LNCS 4928, pp. 169–178, 2008.
Springer-Verlag Berlin Heidelberg 2008

- The ability to define what cannot happen as constraints to the dynamic ordering of activities. These would be defined as declarative rules global to a process and all its sub processes.
- The ability to link activity events (start or completion of an activity) back to a process to trigger process state transition. This is to cater for activities that are performed outside of the process viewpoint control.
- Ideally, the ability to define suggested activity flow sequence to prompt resources to execute activities in the optimal order.

In addition to the above modeling requirements, the activity viewpoint would ideally be supported by an activity or work portal that is an integration of work sources (e.g. BPMS, email etc) and work tools (e.g. application systems, groupware etc) into an activity or work portal to facilitate the knowledge worker to multitask across multiple activities across multiple processes. The detailed requirements of a work portal are beyond the scope of this paper. However it is worth noting that portals is a very active research focus. Gartner [8] have defined six generations of portal evolution and call portals "the Swiss army knife of enterprise computing."

6 Resource Viewpoint Requirements

In this scenario, the resource viewpoint is the controller based upon a pool of outstanding and forecasted work (i.e. activities to be performed on specific process instances), a resource optimisation strategy (whether centralised or distributed) assigns resources to outstanding work.

The modeling requirements of this scenario are:

- To facilitate the most options for runtime optimisation, it is best to minimize unnecessary prescriptive activity flow definition allowing the optimisation algorithm to decide the best activity flow. Thus the declarative rule definition requirement described for the process viewpoint that defaults to a set of action items is encouraged to enable greater resource optimization flexibility.
- Resource requirements per activity per process type need to be defined to enable the resource optimizer to predict resource requirements.

The resource viewpoint is an area poorly supported in current BPM systems (BPMS). Most BPMS tools will support simulation as a means of identifying the optimisation of resource allocation. However as Reijers and van der Aalst [9] highlight, a simulation model typically focuses on a single process while the people involved distribute their time over multiple processes.

The benefit of applying resource management to BPM is subject to the accuracy of the defined resource requirements per activity per process type. This definition consists of two components (using labour hours as the example resource):

- The expected time each individual activity will take.
- The volume mix of the frequency of each activity within the total process based upon the percentage of cases that follow conditional routings within the process.

Both of these components need to measured over some sample size and sample timeframe. It would be sensible to have the BPMS track these components as the basis for calculating these on a regular snapshot basis for each process

On initial analysis, one could conclude that the resource management dimension has no bearing on the breadth / depth complexity challenge. However, in a practical implementation of this strategy, the abstraction level of the activity tracked is an important factor in ensuring the accuracy of the time estimate per activity. At too high an abstraction level, the time estimate is too broad an average with a high standard deviation. Too low an abstraction level, results in an onerous exercise to define a very fine grained process flow and the resulting tracking sample size may be too small to provide a statistically accurate result.

Thus, the depth of activity decomposition is an important driver for accurate resource management and as a result has an influence on the depth complexity of the process design.

A critique could also be made that this viewpoint is oriented towards production process management and not relevant to the less structured processes where activities are more difficult to estimate. As a business management culture, it is proposed that the practice of setting a plan, measuring an actual and identifying the variance (as represented in the Plan Do Check Act (PDCA) management philosophy originally proposed by Deming) is applicable to the whole continuum of activity structure and predictability. For less predictable activities, the abstraction level the plan is pitched may be higher, the timeframe projected forward may be shorter and the plan may be unique for each process / activity instance. A classical example within the normally highly structured manufacturing environment is repairing a machine. The PDCA philosophy would advocate that a plan with target time is made for assessment of the fault and then having identified the fault, a plan and estimate is made for the repair.
Thus resource management in this scenario needs to be more dynamic and emergent.

7 Management Viewpoint Requirements

A key driver of the PARM framework is the ability for each viewpoint to operate independently with integration both at the definition phase as well as based upon execution feedback. The Management viewpoint is the integrator of the other viewpoints through balancing the tension between service, cost and quality expectations.

The modeling requirements of this viewpoint are:

- A common meta-model that each viewpoint's requirements can be mapped to, facilitating the integration of the requirements before mapping to the desired implementation architecture.
- An event logging model that through process mining facilitates the execution feedback to each viewpoint.
- A measurement model that utilises the event logging information to evaluate operational performance in service, quality and cost as feedback to evolution of each viewpoint's design.

8 PARM Framework Implementation

The scope of this paper is to outline the PARM framework requirements. A detailed description of the PARM framework implementation is the scope of the next paper in this series. However it is appropriate to define an architectural context in which the requirements need to be implemented.

Model Driven Development (MDD) is a software engineering approach consisting of models and model technologies to raise the abstraction level at which software is created. Although there are many implementations of MDD (as per Hailpern et al. [10]), the most prevalent is the Object Management Group's (OMG's) implementation of MDD called Model Driven Architecture (MDA)[11].

MDA defines a viewpoint as abstracting to a selected set of parts, connectors and rules in order to focus on a particular concern. It defines three viewpoints:

- A computation independent viewpoint focuses on the context and requirements of the system hiding structure and processing of the system.
- A platform independent viewpoint focuses on the operation of the system while hiding the implementation details for a specific platform.
- A platform specific viewpoint focuses on the implementation of the platform independent viewpoint to a specific platform.

MDA advocates raising the level of application definition abstraction by producing models at each viewpoint where ideally the next level of abstraction is generated automatically from the higher level.

From a PARM framework implementation perspective, the goal is to provide extensions to existing techniques within each of the abstraction levels rather than mandating the use of specific techniques. Applying this to specific MDA abstraction levels: the goal is to provide modeling extensions to existing process modeling approaches as the platform independent viewpoint; that then map into existing implementation architectural styles as the platform specific viewpoint.

The next two sections will define the context within which the PARM framework requirements need to be mapped to process modeling approaches and implementation architectural styles.

9 Process Modeling Approaches

From a platform independent viewpoint, many techniques and standards have evolved for abstracting to a higher level the modeling of process logic.

There have been multiple schemes proposed for categorising modeling languages. For example Giaglis [12] proposes four perspectives: functional, behavioural, organisational, informational. Huff [13] identifies four different categories of process modeling language (PML) paradigms: Non-executable, State-based PMLs, Rule-based PMLs, Imperative PMLs.

For the purposes of defining how the PARM requirements impact process modeling approaches, this paper proposes a process modeling categorization that amalgamates characteristics of both categorisation schemes above.

Two fundamental logic expression paradigms can be leveraged to model process logic: procedural logic and declarative logic.

Procedural logic prescriptively defines a predetermined flow of activity control. Procedural logic techniques vary in their level of abstraction attained ranging from the low level of abstraction achieved by procedural code (using languages such as Java and Cobol for example) ranging up to the high level of abstraction of unstructured techniques like use case scenarios. In the middle of this range is the multitude of graphical techniques and their associated notations.

Given the PARM framework's de-emphasis of fine grained prescriptive activity flow definition, the capabilities of existing mainstream graphical process modeling languages are adequate to meet the prescriptive process modeling requirements of the PARM framework.

Declarative logic defines the constraints (via the use of rules) within which the process can execute and the actual activity flow sequence is determined dynamically at execution of the rules. As per procedural logic, the level of abstraction of declarative logic techniques varies ranging from the low level abstraction of expressing rules directly in a rules language to the mid level technique of decision tables (that are then translated into a rules language) and the high level abstraction of structured English. The rules language can be either a proprietary vendor rules language or desirably based upon a standards based rule language such as OMG's Semantics of Business Rules and Vocabulary (SBVR).

As defined by Ross [14], declarative logic can be used to define all process logic. Although declarative logic is more powerful in its ability to express more complex unstructured process logic, it's more difficult to interpret and communicate the process flow. Consequently procedural logic is a more describable technique for the predictable activity flow for structured processes.

The breadth complexity challenge advocates that procedural logic techniques by themselves lack the expressive power to handle the breadth of unstructured process logic. Consequently to achieve the PARM framework breadth requires integrating procedural logic techniques with declarative logic techniques.

The PARM framework requirements described above translate into two uses of declarative rules integrated within a prescriptive modeling approach:

- As global constraints that are defined for a specific process and its entire sub processes providing the "stop what must not happen" requirement.
- AS the basis for implementing the set of action items requirement representing the ability to define activities as an unstructured non deterministic sequence of actions.

10 Implementation Architectural Styles

From a platform specific viewpoint, many architectural styles have evolved as a means of implementing the operation of a system and specifically process logic.

A classification of architectural styles is proposed by Fielding [15]. Fielding bases his categorisation on the constraints inherent in the communication of components of the system. Fielding defines twenty-two styles on this basis with the recognition that there are further possible styles.

As zur Muehlen [16] documents in his standards landscape, there are a multitude of standards from multiple standards organizations targeted at varying objectives within the total implementation architecture domain. There is much debate over which abstraction level of standard is more important and whether one standard replaces another.

It is beyond the scope of this paper to define in detail the multitude of architectural styles and standards in the implementation domain.

However, given the role events are advocated to play in the PARM framework's requirement to integrate the process, activity, resource and management views, it is appropriate to propose the Event Based Architecture (EBA) style as a strong candidate for implementation of the proposed PARM framework approach. Seybold [17] defines EBA based upon business events occurring inside or outside an organization that is then notified to interested parties who then evaluate the event and optionally take action. .

Gartner in a research report on the role of events in enterprise applications [18] state that event driven processes is the key underlying factor that will enable the revolutionary improvements in business processes and application systems as advocated by the business process fusion concept.

11 Conclusion

Conceptually we want to support the mapping of any process modeling approach to any implementation architectural style. The goal of the PARM framework is not to edict a mandatory choice of either. The goal is to provide extensions to modeling approaches to address breadth and depth complexity and illustrate the ability to map these extensions into multiple architectural implantation styles. However from a realistic starting point, a subset of both must be chosen.

In a real implementation of the PARM framework, the choice of modeling and implementation styles will be influenced by specific business domain requirements and the existing legacy implementation environment.

The PARM framework solution is an extension and integration of existing solutions rather than a solution in its own right.

The next step is to propose the extensions required to existing modeling approaches to support the PARM framework requirements and map these to implementation architectural styles.

A subset of modeling approaches and implementation architectural styles will be chosen as a realistic scope for illustrating the PARM framework solution.

References

1. Porter, M.E.: Competitive Advantage (1985)
2. Stabell, Fjeldstad: Configuring value for competitive advantage. On chains, shops and networks. Strategic Management Journal 19(5), 413–417 (1998)
3. Hayward, S.: Gartner Group. Business Process Fusion: Enabling the Real-Time Enterprise (October 16, 2003)
4. McDonald, M., Rowsell-Jones, A.: Gartner Group. Agility and Efficiency: Business Process Fusion (April 2004)
5. Soanes, M.G.: Process Design Strategies to Address Breadth and Depth Complexity. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, Springer, Heidelberg (2006)

6. Chew, E., Hawryszkiewycz, I., Soanes, M.: Value Configuration Design – an evolution in adequate business process design. In: Krogstie, J., Opdahl, A., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, Springer, Heidelberg (2007)
7. Ould, M.: Business Process Management: A Rigorous Approach (2005)
8. Gootzit, D., Valdes, R., Phifer, G.: Gartner Group. RAS Core Research Note (December 13, 2006)
9. Reijers, H., Van Der Aalst, W.: The effectiveness of workflow management systems: Predictions and lessons learned. International Journal of Information Management (2005)
10. Hailpern, B., Tarr, P.: Model-driven development: The good, the bad, and the ugly. IBM Systems Journal 45(3), 451–462 (2006)
11. Object Management Group. MDA Guide V1.0.1, http://www.omg.org
12. Giaglis, G.M.: A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques. The International Journal of Flexible Manufacturing Systems 13, 209–228 (2001)
13. Huff, K.E.: Software Process Modeling. In: Fuggetta, A., Wolf, A. (eds.) Trends in Software Process, pp. 1–24. John Wiley & Sons, Chichester (1996)
14. Ross, R.G.: Principles of the Business Rules Approach (2003)
15. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine (2000)
16. zur Muehlen, M.: Standards Landscape, at http://www.workflow-research.de
17. Michelson, B.: Patricia Seybold Group. In: Event Driven Architecture Overview, 2nd edn. (February 2006)
18. Schulte, R.W.: Gartner Group. The Growing Role of Events in Enterprise Applications (July 9, 2003)