Support for the Business Motivation Model in the WS-Policy4MASC Language and MiniZnMASC Middleware

Qinghua Lu[1,2], Vladimir Tosic[1,2], and Paul L. Bannerman[1,2]

[1] NICTA, Australian Technology Park, Sydney, NSW, Australia
[2] University of New South Wales, Sydney, NSW, Australia
{Qinghua.Lu,Vladimir.Tosic,Paul.Bannerman}@nicta.com.au

Abstract. The WS-Policy4MASC language and MiniZnMASC middleware for policy-driven management of service-oriented systems enable making IT system management decisions that maximize diverse business value metrics (e.g., profit, customer satisfaction). However, their past support for alignment with high-level business considerations was weak. Therefore, we introduce a new extension of WS-Policy4MASC that specifies the key concepts from the Business Motivation Model (BMM) industrial standard for modeling business intent. These concepts include hierarchies of ends (e.g., goals) and means (e.g., strategies). We also present and illustrate new decision making algorithms that leverage information in the extended WS-Policy4MASC to align run-time IT system management decisions with business considerations.

Keywords: Business-driven IT management, business motivation model, dynamic adaptation, policy-driven management, self-management, service-oriented computing, Web service management.

1 Introduction

Information technology (IT) systems are rarely ends in themselves. They usually execute in support of actions to fulfill operational and strategic objectives of an organization. Therefore, organizational performance includes a measure of the effectiveness of IT systems in meeting business drivers and expectations. This paper illustrates how the high-level business constructs in the Object Management Group's (OMG) Business Motivation Model (BMM) can guide the execution of service-oriented systems, through extensions to the WS-Policy4MASC language and MiniZnMASC middleware for policy-driven self-management. While the implementation of the presented system is for management of service-oriented systems and business processes, the underlying conceptual solutions can also be generalized to policy-driven self-management of other IT systems.

Due to the high complexity of management tasks and the cost of experienced human system administrators, it is more efficient and cost-effective for IT systems to be self-managing, directed by high-level policies at run-time. Self-management has been a research goal for several decades, but was made prominent by the vision of

G. Kappel, Z. Maamar, H.R. Motahari-Nezhad (Eds.): ICSOC 2011, LNCS 7084, pp. 265–279, 2011.

autonomic computing [1]. Additionally, while business users are typically interested in maximizing business value, prior IT system management solutions have mostly focused on optimizing technical quality of service (QoS) metrics and not directly on maximizing business value. The goal of business-driven IT management research (BDIM) is to determine mappings between technical and business metrics and leverage these mappings to make run-time IT system management decisions that maximize business value metrics [2]. Autonomic BDIM is the intersection area between autonomic computing and BDIM, where processing of business value metrics is added to the decision making components of autonomic computing. There are many open research challenges in this intersection area [2].

Service-oriented computing has become the dominant way of building distributed computing systems. While business-driven management of service-oriented systems is mentioned in prior research [3-5], there are still many open issues. One of the limitations of the past BDIM research is that most works have focused on maximizing profit. However, human business managers can have diverse business objectives. Maximizing short-term profit may not always be the only or best approach to achieve long-term or high-level business goals [4-5]. Business motivation (such as goals, strategies) is a major differentiator of companies in a market, so it is an ideal mechanism to incorporate in self-managing BDIM solutions to direct decision making in controlling and adapting IT systems in response to run-time changes.

BMM is an OMG standard for specification of high-level business motivation and intent as input into design, development and execution of IT systems [6]. Our WS-Policy4MASC language [4] for specification of policies for management of IT (particularly service-oriented) systems provides unique support for implementing autonomic BDIM solutions. Our algorithms using WS-Policy4MASC information, implemented in the MiniZnMASC middleware [7], make decisions for dynamic adaptation of service-oriented systems that maximize diverse financial and non-financial business value metrics. However, the alignment with high-level business value metrics was weak in our previous work. Therefore, we now present extensions to WS-Policy4MASC incorporating key BMM constructs and new run-time self-management algorithms to leverage these additional metrics.

In the next section, we present background information on WS-Policy4MASC, MiniZnMASC and BMM, and overview other major related work. The main section of the paper details our WS-Policy4MASC extensions with the key BMM constructs and our new BDIM algorithms that use these metrics. In the final section, we summarize conclusions and future work.

2 Background and Related Work

2.1 WS-Policy4MASC and MiniZnMASC

WS-Policy4MASC [4], our extension of the WS-Policy industry standard, is a policy language that can describe various adaptations and all information necessary for decision making. WSPolicy4MASC defines five types of WS-Policy policy assertions: 1) goal policy assertions (GPAs) prescribe conditions to be met; 2) action policy assertions (APAs) list adaptation actions; 3) utility policy assertions (UPAs) contain business metrics for particular situations; 4) probability policy assertions (PPAs) specify probabilities of occurrence, and; 5) meta-policy assertions (MPAs) describe which values are important for adaptation decisions.

The specification of diverse business value metrics in UPAs and MPA specification of strategies for choosing among alternative adaptation actions are the main original contributions of WS-Policy4MASC and differentiators from the other WS-Policy extensions. Each adaptation approach is modeled as an APA. WS-Policy4MASC enables specification of both financial and non-financial business value metrics (BVMs) in UPAs. A number of UPAs can correspond to consequences of executing a particular APA or meeting a particular GPA. WS-Policy4MASC enables dealing with uncertainty through PPAs, e.g. by assigning probabilities that different estimates of the same BVM will be correct. It enables specification of business strategies in MPAs, as a means of deciding which among alternative adaptation approaches to take in policy conflict situations when several APAs could be applied, but only one can be chosen. In addition to policy assertions, WS-Policy4MASC also specifies details necessary for run-time management in auxiliary constructs: ontological meaning, monitored QoS metrics, monitored context properties, states, state transitions, events, schedules, applicability scopes, and various types of expression.

Our MiniZnMASC middleware [7] is a comprehensive framework for autonomic management of service-oriented systems and business processes. It implements novel decision-making algorithms that, at runtime, concurrently provide adaptation decisions, depending on different business strategies and operational circumstances, in a way that achieves maximum overall business value while satisfying all given constraints. These decision-making algorithms use information specified in WS-Policy4MASC policy assertions. They are triggered by monitored events, such as not meeting a GPA. When adaptation decisions are needed at the same time for a number of business process instances, MiniZnMASC uses the constraint programming language MiniZinc [8] to make such decisions. Although our past MiniZnMASC publications focus on autonomic BDIM support, this middleware can be used also for traditional decision-making that maximizes technical metrics. This is because WS-Policy4MASC can describe all information necessary for adaptation decision-making. Our evaluation using several prototype implementations showed that the proposed MiniZnMASC architecture and decision-making algorithms are feasible and easy to modify. Furthermore, our performance and scalability tests showed that MiniZnMASC does not introduce unforeseen performance or scalability problems.

2.2 Business Motivation Model (BMM)

BMM [6] comprises a set of abstractions that define elements of business plans integrated with high-level processes to accommodate business change. It provides the business motivational intelligence to frame operational system services within a context of ongoing business change. As depicted in Figure 1, BMM integrates four primary motivational elements in the model: end, means, influencers and assessment. *End* defines the organization's aspirations – what it wants to be or become. *Means* specify the actions the organization will undertake to achieve the desired ends. *Influencers* are internal or external causes of change that may influence the organization's business motivation. An a*ssessment* is a judgment about the impact of an influencer on the organization's current end and/or means. Assessments may employ existing analysis techniques such as SWOT (strength, weakness, opportunity, threat) and consider *potential impacts* of influencers in terms of risks and *potential rewards*. Assessment decisions may result in changes to the current end and/or means.

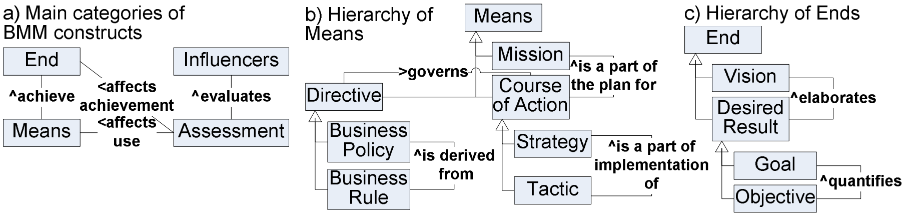

Fig. 1. Main concepts and interrelationships in BMM (based on [6])

The end and means comprise component elements. An end comprises a vision and desired result; a desired result comprises a goal and objective. *Vision* is "an overall image" of the aspiration (that is, it may not be fully or explicitly defined). Goals and objectives are more specific. A *Goal* is a qualitatively defined desired result, while an *Objective* is a quantitatively defined discrete step towards achieving a goal. Objectives provide metrics for measuring progress towards the stated goal. Means comprise mission, course of action and directive. *Mission* describes the broad ongoing operational activity of the organization. *Course of Action* define the actions the organization will undertake in terms of *Strategies* (broadly scoped actions) and *Tactics* (narrowly scoped actions). The BMM hierarchy of means also includes *Directives* and the overall model includes *Influencers* and *Assessment. Directives* specify the business policies and rules that frame courses of action. *Business Policies* define what can and cannot be done within means in a general sense, but are not directly actionable. By contrast, *Business Rules* provide specific actionable guidance to implement or fulfill business policies but are defined externally to BMM. Similarly, organization unit and business process have roles in BMM but are defined externally, in other OMG standards. *Influencers* are anything that can impact employment of means or achievement of ends; while an *Assessment* is a judgment about the impact of an influence.

The major value of BMM is in providing business plan-based structured 'intelligence' to guide the implementation and execution of software-based business services. It provides an abstraction of why businesses pursue particular ends through particular means and includes a basic assessment mechanism to consider emergent change impacts over time. However, BMM also has limitations, derived from simplifying assumptions that limit its application in practice. First, it assumes a predominantly stable operating environment. This may not be possible in organizations in highly dynamic and volatile environments. Second, it assumes that strategies and tactics are and can be made explicit. This is usually not possible in the incremental/emergent approach to strategy. Third, while the standard acknowledges that BMM may be applied at different organizational levels, it does not explicitly accommodate multi-unit modeling. Finally, several elements in the model, such as vision, do not lend themselves to explicit definition or necessarily translate into tangible entities. The presence of intangibles can be a significant barrier in the implementation of autonomously managed systems. Notwithstanding these limitations, BMM does provide a standard definition of high-level business value constructs that can be mapped as drivers for IT systems management.

2.3 Other Related Work

The need to frame and integrate service-oriented systems with business level drivers, such as goals and strategies, is increasingly recognized in the literature. A common rationale for this is alignment of the organization's IT-enabled operations with its strategic motivations and directions. However, design-time integration is studied much more often than execution time integration.

A potential alternative implementation model is the balanced scorecard (BSC). Conceptually, the balanced scorecard has been influential in motivating alignment of operations with organizational strategies. Proposed as a mechanism to map strategy to implementation, a balanced scorecard can represent the cause and effect linkages through which specific improvement objectives can be realized [9]. Strategic maps reflect how business strategies can be achieved through supporting financial, customer, internal business process, and learning and growth implementation perspectives. While the balanced scorecard has been explicitly applied to IT (the *IT balanced scorecard*) to support alignment through IT governance [10], it has not yet been fully implemented at the systems level. A key difficulty in implementing the balanced scorecard in self-managed service-oriented systems is its high dependency on intangibles [11]. Consequently, its application has been primarily a manual process. Nevertheless, the balanced scorecard concepts inspire and influence researchers to align service-level operations to business strategies.

In particular, [12] proposes an approach for IT service management by business objectives called IT Management by Business Objectives (MBO). Their information model contains objectives, key performance indicators (KPIs), and perspectives, inspired by BSC. Decision support is provided through deployment of a reasoning engine, Aline, which computes the alignment of alternative courses of action to objectives, providing a measure of utility and basis for ranking the decision options and returning a recommendation. However, this novel approach is limited by the vagaries of assessing intangibles inherent in the perspectives, as in BSC. While there is other BDIM-related work, MBO is the one most closely related to our research.

Others have focused on design-time alignment of organizational goals, objectives, strategies or business requirements with process models. These include, for example, the *Tropos* development methodology based on *i** organizational modeling for requirements and design [13]; a framework for representing organizational strategies and goals in terms of business requirements using *Tropos* and *Formal Tropos* and implemented by activities in business processes through Web services using an early version of WSBPPEL [14]; co-evolution of operational business process models using the Business Process Modeling Notation (BPMN) and organizational models using the *i** modeling notation [15]; relating BPMN-based business process models to high level stakeholder goals modeled using KAOS [16]; an approach to process design and configuration management using requirements goal models to capture alternative process configurations [17]; and a map-driven process modeling approach based on intentions and strategies abstracted from organizational tasks [18].

Furthermore, many languages and supporting tools have been developed by academia and industry for specification of policies, SLAs, and related IT system management constructs for service-oriented systems. They are often accompanied by corresponding run-time management middleware. While significant, these approaches tend to focus on monitoring of technical QoS metrics, providing limited business value metric-based control capabilities. The WS-Policy extensions WS-QoSPolicy [19] and WS-CoL [20], and the corresponding middleware, are closest to our research but they do not address self-managing BDIM.

3 Extension of WS-Policy4AMSC and MiniZnMASC with Key BMM Constructs

When changes (e.g., in system performance) occur during runtime, the affected Web service compositions should be adapted. If there is more than one available adaptation option, a decision is needed to resolve the 'conflict' to determine which adaptation option should be executed. It is often appropriate to maximize business value in such adaptation decision-making. To improve the support for business value considerations in adaptation of Web service compositions, we extended WS-Policy4MASC and MiniZnMASC with key constructs from the BMM industrial standard for modeling business intent. These constructs are from the End and Means hierarchies (but Directives are not included). In the BMM Means hierarchy (see Figure 1.b), a Mission is implemented through Strategies that, in turn, are implemented via Tactics. In a BMM End hierarchy (see Figure 1.c), a Vision is composed of Goals that are measured through Objectives. There are analogies between the BMM End hierarchy and the BMM Means hierarchy: Vision is the final end of the BMM hierarchy and is at the same abstraction level as Mission, Goal is at the same level as Strategy, and Objective is at the same level as Tactic.

A Vision can be achieved via a bottom-up approach by starting from Tactics. Contribution of particular lower-level constructs to the achievement of higher-level constructs (e.g., contribution of a Tactic to the achievement of a Strategy) and contribution of particular means to the achievement of ends (e.g., contribution of a Strategy to the achievement of a Goal) can be shown as directed arcs between nodes representing BMM constructs. These arcs can be in AND/OR relationships. The OR relationship means that any of the lower-level constructs can achieve the higher-level construct. The AND relationship means that all lower-level constructs must be achieved for achievement of the higher-level construct. To denote the strength of a particular contribution, we specify a Utility Contribution Weight (UCW) for each arc. A UCW is a relative value (between 0 and 1) that denotes how much of the business value produced at the higher-level BMM construct is due to the achievement of the lower-level BMM construct. In principle, the sum of all UCWs of arcs incoming into the same node should be 1. To denote probabilities in OR relationships, we specify Occurrence Probabilities (OPs). The sum of all OPs in an OR relationship should be 1. If an arc is not in any OR relationship, its OP is 1. If values for some of the UCWs or OPs are not given, default values are calculated based on the assumption that arcs with missing information have mutually equal contributions and probabilities. For example, if there are 2 arcs in an OR relationship and OPs are not given, the default values are OP=0.5 for both these arcs.

To illustrate how adaptation decisions are made using the BMM End and Means hierarchies, we provide an example of a business scenario. A loan broker company provides a loan brokering Web service composition shown in Figure 2. The BMM constructs and their values for the example are listed in Table 1. This company classifies its consumers into three classes according to the customers' previous loans record and credit history with the company: gold, silver, and bronze. The company provides different classes of consumer with different technical QoS guarantees, prices per year, and penalties if the guarantees are not met.

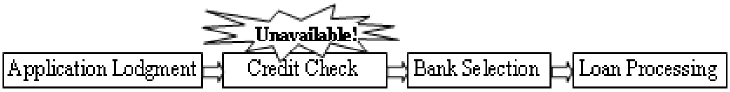

Fig. 2. A loan brokering Web service composition

Table 1. BMM constructs in the loan broker company

BMM Construct	**Construct Value**
Vision	To be a world-class company, helping people and organizations to grow
Mission	To be the provider of first choice across loan brokering services
Goal1	To increase product sales
Goal2	To improve customer satisfaction
Objective1	To have $300,000,000 turnover in this year of operation
Objective2	To build a reputation for quality, error-free services and products
Strategy1	Ensure that the service composition has a relatively high availability
Strategy2	Ensure that the completion time of the service composition is short
Tactic1	Use a replacement external credit service when the credit check service becomes unavailable
Tactic2	Skip the credit check service
Tactic3	Use an internal credit check service

The credit check service is a third-party service provided by an external credit check agency. During runtime, this service suddenly becomes unavailable for some reason. The adaptation system finds a replacement, service S, for the credit check service. It takes some time to set up the replacement service. Therefore, in addition to having an alternative tactic "use the replacement external service", the adaptation system also provides two other alternative tactics: "skip the credit check service" and "use an internal credit check service".

In this context, different tactics have different value contributions towards the business vision. To determine which Tactic to execute, business value contributions towards the business Vision have to be calculated according to policies. Figure 3 illustrates an example of BMM End and Means hierarchies of the loan-broking company. In this example, we identify three alternative tactics in the BMM hierarchies. The selected Tactics have to satisfy various constraints including cost limit constraints (in this example: $100 total cost) and other constraints.

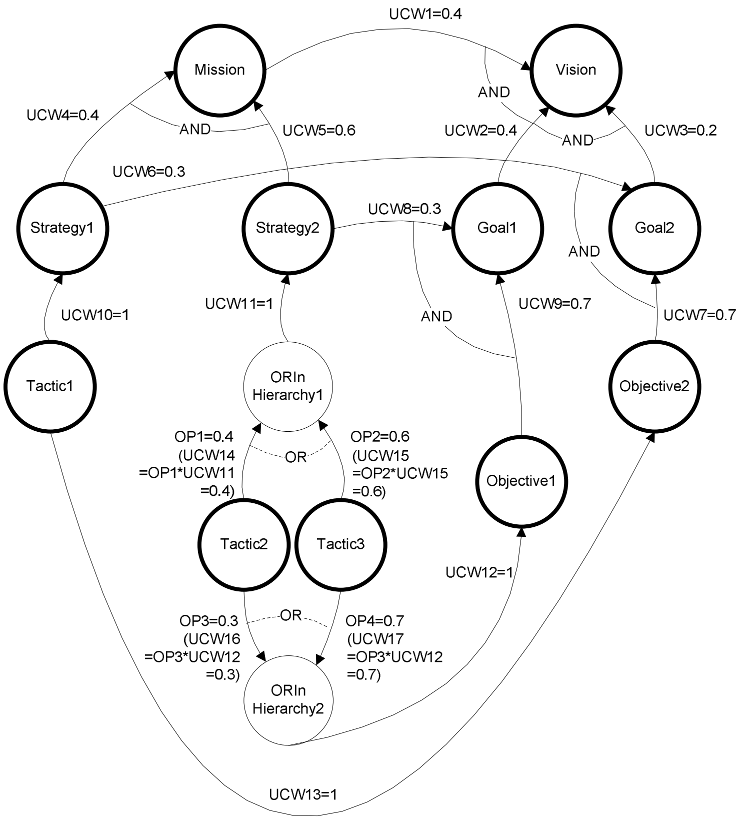

Fig. 3. BMM hierarchy in the loan broker company

Table 2. Business value for bronze consumers of each node in the BMM hierarchy

BMM hierarchy node	Business value
Vision	$10,000
Mission	$1,000
Goal1	$4,000
Goal2	$5,000
Strategy1	$200
Strategy2	$100
Objective1	$400
Objective2	$500
Tactic1	$10
Tactic2	$30
Tactic3	$20

In Figure 3, we illustrate how to use the BMM End and Means hierarchies to select one Tactic for each class of consumer, among several options. We assume that the same Tactic has different business values to different classes of consumers. To simplify the discussion, we only discuss how to determine which Tactic to select for bronze consumers. Table 2 shows the business value of each node for bronze consumers in the BMM End and Means hierarchies.

Starting from Tactic1, we have two options: Strategy1 and Objective2. Let us meet Strategy1 first (there is no difference in starting from Strategy1 or Objective2). We have two options to go after Strategy1: Mission and Goal2. Let’s select Mission to meet first. After meeting Mission, we achieve Vision, which is the end point. After meeting Vision, we calculate the total business value of meeting each hierarchy node in the return path. The calculation for the loan broking example is shown in Figure 4. As it is the same way for other paths, we do not discuss repetitively here. V represents the business value of directly achieving a BMM hierarchy node. V’ represents the total business value contribution a BMM hierarchy node makes and it includes the business value of directly achieving this hierarchy node and the sum of contributions this node makes to achieving each of the related higher-level BMM hierarchy nodes. As shown in Figure 4, Tactic3 brings the maximum high-level business value. Therefore, the algorithm selects Tactic3 as the adaptation action for bronze consumers.

1) **V**’(Vision)=V(Vision)=$10,000
2) V’(Mission)=V(Mission)+ UCW1*OP*V’(Vision)=$1,000+0.4*1*$10,000 = $5,000
3) V’(Goal1)=V(Goal1)+UCW2*OP*V’(Vision)=$4,000+0.4*1*$10,000=$8,000
4) V’(Goal2)=V(Goal2)+UCW3*OP*V’(Vision)=$5,000+0.2*1*$10,000=$7,000
5) V’(Strategy1)=V(Strategy1)+UCW4*OP*V’(Mission)+UCW6*OP*V’(Goal2)=$200 +0.4*1*$5,000+0.3*1*$7,000=$4,300
6) V’(Strategy2)=V(Strategy2)+UCW5*OP*V’(Mission)+UCW8*OP*V’(Goal1)=$100 +0.6*1*$5,000+0.3*1*$8,000=$5,500
7) V’(Objective1)=V(Objective1)+UCW9*OP*V’(Goal1)=$400+0.7*1*$8,000=$6,000
8) V’(Objective2)=V(Objective2)+UCW7*OP*V’(Goal2)=$500+0.7*1*$7,000=$5,400
9) V’(Tactic1)=V(Tactic1)+UCW10*OP*V’(Strategy1)+UCW13*OP*V’(Objective2) =$10+1*1*$4,300+1*1*$5,400=**$9,710**
10) V’(Tactic2)=V(Tactic2)+UCW11*OP1*V’(Strategy2)+UCW12*OP3*V’(Objective1) =$30+1*0.4*$5,500+1*0.3*$6,000=**$4,310**
11) V’(Tactic3)=V(Tactic3)+UCW11*OP2*V’(Strategy2)+UCW12*OP4*V’(Objective1) =$20+1*0.6*$5,500+1*0.7*$6,000=$**7,520**

Fig. 4. The process of calculating high-level business value in the loan-broker example

We extended our WS-Policy4MASC language and MiniZnMASC middleware with the key BMM concepts in four steps:

1) conceptually extended WS-Policy4MASC with all key BMM constructs;

2) coded the new extensions into the WS-Policy4MASC language schema and tested this on examples;

3) conceptually extended and improved the MiniZnMASC BDIM algorithms to use BMM-related features supported by WS-Policy4MASC, and;

4) implemented and tested the BDIM algorithm in Java.

Note that our solution is for IT system management of one operational business process; not the whole organization. The solution has to be modified and extended to be applied to the IT system management across the whole organization.

The starting point for our integration of BMM and WS-Policy4MASC was an examination of mappings between the BMM concepts and the WS-Policy4MASC concepts. Due to the space limitations, we focus in this paper only on the main aspects crucial for our algorithms that use BMM constructs in MiniZnMASC. The full mappings table and its discussion are available in [21].

```
<!-- Definition of the BMMHierarchy -->
<masc-bh:BMMHierarchy MASCID="BMMHierarchy1">
    <masc-bh:ToBeAchieved>
        <masc-bh:ActionPolicyAssertionRef To="Strategy1" />
    </masc-bh:ToBeAchieved>
    <masc-bh:ANDInHierarchy MASCID="ANDInHierarchy1">
        <masc-bh:ORInHierarchy MASCID="ORInHierarchy1">
            <masc-bh:UCWRef To="UCW7" />
            <masc-bh:AchievedBy>
                <masc-bh:OPRef To="OP3" />
                <masc-bh:ActionPolicyAssertionRef To="Tactic1" />
            </masc-bh:AchievedBy>
        </masc-bh:ORInHierarchy>
        <masc-bh:ORInHierarchy MASCID="ORInHierarchy2">
            <masc-bh:UCWRef To="UCW8" />
            <masc-bh:AchievedBy>
                <masc-bh:OPRef To="OP1" />
                <masc-bh:ActionPolicyAssertionRef To="Tactic2" />
            </masc-bh:AchievedBy>
            <masc-bh:AchievedBy>
                <masc-bh:OPRef To="OP2" />
                <masc-bh:ActionPolicyAssertionRef To="Tactic3" />
            </masc-bh:AchievedBy>
        </masc-bh:ORInHierarchy>
    </masc-bh:ANDInHierarchy>
</masc-bh:BMMHierarchy>
```

Fig. 5. An example of the BMMHierarchy definition

The corresponding concept for BMM Means in WS-Policy4MASC is the action policy assertion, while the corresponding concept for BMM End in WS-Policy4MASC is the goal policy assertions. In order to specify relationships between hierarchies of BMM ends and means, we added the new “BMMHierarchy” construct to WS-Policy4MASC. The sub-elements under “BMMHierarchy” are “ToBeAchieved” and “ANDInHierarchy”. “ToBeAchieved” is what the arc arrow between two nodes in the BMM hierarchy points to. The elements under “ToBeAchieved” can be “GoalPolicyAssertionRef” or “ActionPolicyAssertionRef”. “ANDInHierarchy” has the sub-element “ORInHierarchy”. There should be at least one “ORInHierarchy” bounded to an “ANDInHiearchy”. “ORInHierarchy” has two sub-elements: “UCWRef” and “AchievedBy”. “UCWRef” refers to utility contribution weights for the “AchievedBy”. The sub-elements under “AchievedBy” are “ActionPolicyAssertionRef” or “GoalPolicyAssertionRef”, and “OPRef”. “ActionPolicyAssertionRef” or “GoalPolicyAssertionRef” refers to what the arrow between two nodes in the BMM hierarchy points from. “OPRef” refers to the occurrence probability of “AchievedBy”. Figure 5 shows an example of the BMMHierarchy definition.

The extended algorithm for selection of the best options among conflicting action policy assertions is outlined in Figures 6 and 7. Here, the “best” means “highest overall business value, while meeting all constraints” and depends on which combinations of the business value metric categories are used in the calculations as well as on the contribution of particular conflicting action policy assertions to long-term business goals and strategies. This algorithm enables choosing an adaptation option that might not be the best in the short-term, but is the best one when longer-term high-level business considerations are taken into account.

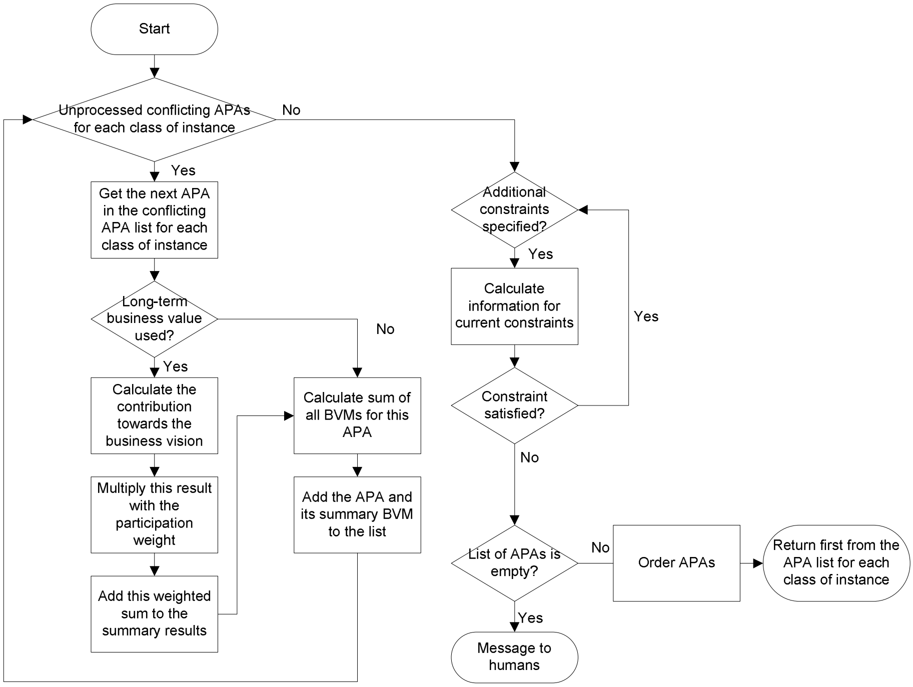
Start
Unprocessed conflicting APAs for each class of instance
No
Yes
Get the next APA in the conflicting APA list for each class of instance
Long-term business value used?
No
Yes
Calculate the contribution towards the business vision
Multiply this result with the participation weight
Add this weighted sum to the summary results
Calculate sum of all BVMs for this APA
Add the APA and its summary BVM to the list
Additional constraints specified?
Yes
Calculate information for current constraints
Yes
Constraint satisfied?
No
List of APAs is empty?
No
Order APAs
Return first from the APA list for each class of instance
Yes
Message to humans

Fig. 6. The conflict resolution algorithm

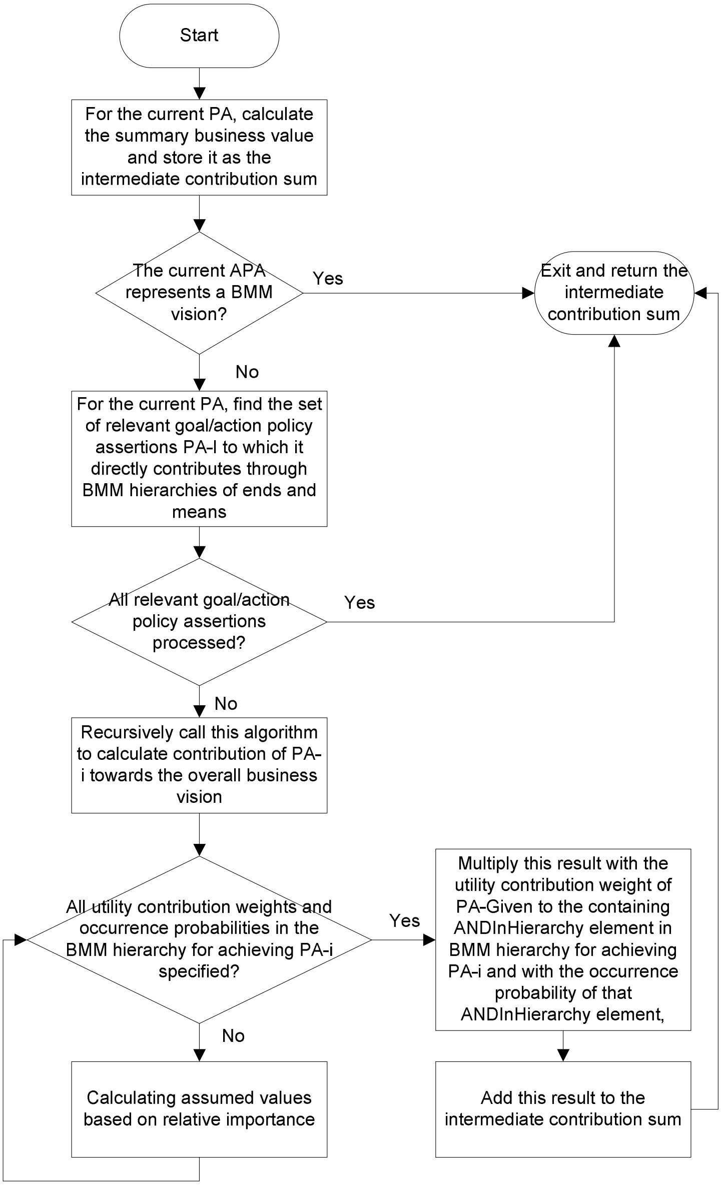
Start
For the current PA, calculate the summary business value and store it as the intermediate contribution sum
The current APA represents a BMM vision?
Yes
Exit and return the intermediate contribution sum
No
For the current PA, find the set of relevant goal/action policy assertions PA-I to which it directly contributes through BMM hierarchies of ends and means
All relevant goal/action policy assertions processed?
Yes
No
Recursively call this algorithm to calculate contribution of PA-i towards the overall business vision
All utility contribution weights and occurrence probabilities in the BMM hierarchy for achieving PA-i specified?
Yes
Multiply this result with the utility contribution weight of PA-Given to the containing ANDInHierarchy element in BMM hierarchy for achieving PA-i and with the occurrence probability of that ANDInHierarchy element,
No
Calculating assumed values based on relative importance
Add this result to the intermediate contribution sum

Fig. 7. Calculation of value contribution towards the business vision

The algorithm first loops through all conflicting action policy assertions (APAs) to check whether each of them satisfies all given constraints and, if yes, to calculate the sum of all business value metrics (BVMs) for these APAs. The APAs that satisfy all constraints are added to the list, along their summary BVMs. If none of the conflicting APAs satisfies the constraints, the resulting list will be empty and an exception is thrown (e.g., human administrators are notified to resolve the issue). The APAs in the list are ordered based on the decreasing value of their summary BVMs. The first APA in the list will have the highest summary BVM while satisfying all constraints. This APA is returned as the adaptation action (which could contain several sub-actions) to be executed.

Notice in Figure 6 that the algorithm checks whether the long-term business contribution is considered in the adaptation decision-making. If yes, the algorithm calculates the contribution towards the BMM business vision, multiplies the result with the participation weights, and then adds this weighted sum to the currently examined action policy assertion as its long-term business value. As shown in Figure 7, if contribution towards long-term business vision is considered in decision-making, then when the summary business value metric is calculated for each adaptation option, the extended decision making algorithm adds weighted contribution towards the overall BMM business vision. The algorithm for calculation of the contribution to the overall business vision traverses the AND/OR hierarchy of BMM ends and means, calculates the summary business value for each hierarchy node, and applies utility contribution weights and occurrence probabilities.

4 Evaluation

We evaluated MiniZnMASC and the developed algorithms for long-term business-driven decision-making on four aspects: feasibility, functional correctness, performance overhead, and scalability. We evaluated feasibility through implementation of several proof-of-concept prototypes. We found no problems with feasibility. We implemented the motivating example from Section 3 and evaluated the functional correctness of MiniZnMASC by comparing the results calculated by MiniZnMASC and by hand. We also developed several other examples for this evaluation. The results showed that MiniZnMASC had been built correctly.

For the performance and scalability tests, we used a Hewlett-Packard laptop model HP EliteBook6930p with Intel Core 2 Duo CPU T900 2.53GHz processor and 4.00 GB of RAM memory, running 32-bit Windows Vista operating system. We measured the performance with increasing number of conflicting action policy assertions (BMM Tactics) to be examined. We started with 3 action policy assertions, then increased to

Table 3. Performance measurement results with increasing number of conflicting action policy assertions (BMM Tactics)

Test case	Execution time of decision making	Execution time of the whole conflict resolution algorithm
3 conflicting action policy assertions	Average: 141 ms Range: 125-157 ms	Average: 47 ms Range: 46-63 ms
10 conflicting action policy assertions	Average: 218 ms Range: 203-343 ms	Average: 78 ms Range: 78-78 ms
100 conflicting action policy assertions	Average: 1186 ms Range: 1029-1389 ms	Average: 437 ms Range: 434-453 ms

10, and then to 100. Using the Java "System.currentTimeMillis()" call, we measured the execution time of autonomic business-driven decision making and the execution time of the whole conflict resolution algorithm in MiniZnMASC. We repeated 100s of tests at different times of day and averaged their results.

Table 3 shows the measured results of the range (min, max) and average of the execution time. The overall execution time of decision making in MiniZnMASC rises because the execution time of the summation of business values for each conflicting action policy assertion increases with increasing number of conflict action policy assertions. The last test case (100 conflicting action policy assertions) is much more complicated than realistic scenarios in practice, so 1.186 sec is not an issue. It is important to note that in realistic scenarios of MiniZnMASC the number of conflicting action policy assertions will be low (typically 2, maybe a few more) while the overall number of action policy assertions can be huge. We also checked that the number of additional non-conflicting action policy assertions in the MiniZnMASC Policy Repository had no significant effect on performance, even when there were hundreds of action policy assertions.

5 Conclusions and Future Work

The main contribution of our research is in aligning organization-level business intentions with run-time execution of service-oriented systems by extending existing language (WS-Policy4MASC) and middleware (MiniZnMASC) for autonomic business-driven IT management with key concepts from the industrial standard for modeling business intent (BMM). These key concepts include hierarchies of ends (e.g., goals) and means (e.g., strategies). We also developed new decision-making algorithms for our MiniZnMASC middleware. The new algorithms leverage information in the extended WS-Policy4MASC to align run-time IT system management decisions with business concerns.

Our ongoing work is on extending WS-Policy4MASC and MiniZnMASC with the remaining BMM concepts (e.g., business policies, business rules, directives, influencers, and assessment). We already have conceptual solutions and WS-Policy4MASC extensions, but we still have to complete the new MiniZnMASC prototype and experiment with it.

Acknowledgments. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. We thank Nahid Ebrahimi Nejad, Shyam Sunder Iyer, and Cheng Li for their contributions to implementation of the extended WS-Policy4MASC and MiniZnMASC prototypes.

References

1. Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing. Computer 36, 41–50 (2003)
2. Bartolini, C., Sahai, A., Sauve, J.P.: Proceedings of the Second IEEE/IFIP Workshop on Business-Driven IT Management (2007)

3. Casati, F., Shan, E., Dayal, U., Shan, M.C.: Business-oriented management of Web services. Communications of the ACM 46, 55–60 (2003)
4. Tosic, V.: Autonomic business-driven dynamic adaptation of service-oriented systems and the WS-Policy4MASC support for such adaptation. Intl. J. of Systems and Service-Oriented Eng (IJSSOE) 1, 79–95 (2010)
5. Tosic, V.: On Modeling and Maximizing Business Value for Autonomic Service-Oriented Systems. In: Ardagna, D., Mecella, M., Yang, J. (eds.) Business Process Management Workshops. LNBIP, vol. 17, pp. 422–433. Springer, Heidelberg (2009)
6. OMG-BMM: Business Motivation Model Version 1.0, `http://www.omg.org/spec/BMM/1.0/PDF`
7. Lu, Q., Tosic, V.: Support for Concurrent Adaptation of Multiple Web Service Compositions to Maximize Business Metrics. In: IM 2011. IEEE, Dublin (2011)
8. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: Minizinc: Towards a Starndard CP Modelling Language. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 529–543. Springer, Heidelberg (2007)
9. Kaplan, R.S., Norton, D.P.: Having trouble with your strategy? Then map it. Focusing Your Organization on Strategy—with the Balanced Scorecard. 49 (2000)
10. Van Grembergen, W.: The balanced scorecard and IT governance (2000)
11. Kaplan, R.S., Norton, D.P.: Measuring the strategic readiness of intangible assets. Harvard Business Review 82, 52–63 (2004)
12. Bartolini, C., Sallé, M., Trastour, D.: IT service management driven by business objectives An application to incident management, pp. 45–55. IEEE (2006)
13. Castro, J., Kolp, M., Mylopoulos, J.: Towards requirements-driven information systems engineering: the Tropos project. Information Systems 27, 365–389 (2002)
14. Kazhamiakin, R., Pistore, M., Roveri, M.: A framework for integrating business processes and business requirements (2004)
15. Koliadis, G., Vranesevic, A., Bhuiyan, M., Krishna, A., Ghose, A.: A combined approach for supporting the business process model lifecycle. Citeseer (2006)
16. Koliadis, G., Ghose, A.: Relating Business Process Models to Goal-Oriented Requirements Models in KAOS. In: Hoffmann, A., Kang, B.-h., Richards, D., Tsumoto, S. (eds.) PKAW 2006. LNCS (LNAI), vol. 4303, pp. 25–39. Springer, Heidelberg (2006)
17. Lapouchnian, A., Yu, Y., Mylopoulos, J.: Requirements-Driven Design and Configuration Management of Business Processes. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 246–261. Springer, Heidelberg (2007)
18. Nurcan, S., Etien, A., Kaabi, R., Zoukar, I., Rolland, C.: A strategy driven business process modelling approach. Business Process Management Journal 11, 628–649 (2005)
19. Rosenberg, F., Enzi, C., Michlmayr, A., Platzer, C., Dustdar, S.: Integrating quality of service aspects in top-down business process development using WS-CDL and WS-BPEL, p. 15. IEEE Computer Society (2007)
20. Baresi, L., Guinea, S., Plebani, P.: Policies and Aspects for the Supervision of BPEL Processes. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 340–354. Springer, Heidelberg (2007)
21. Iyer, S.S.: Analysis of Methods for Improving IT Support for Business. University of New South Wales (2009)