

Maintaining Motivation Models (in BMM) in the Context of a (WSDL-S) Service Landscape

Konstantin Hoesch-Klohe, Aditya K. Ghose, and Hoa Khanh Dam

Decision Systems Lab (DSL),
School of Computer Science and Software Engineering,
University of Wollongong.
{khk789,aditya,hoa}@uow.edu.au

Abstract. The ever-changing business context requires organisations to constantly adapt their motivation and service representations. While there has been work focusing on the relation between the motivation- and service level, very little work has been done in providing machinery for handling (propagating) changes at the motivation level and identifying the resulting impact on the service landscape. In this paper, we propose a novel framework which addresses this problem.

1 Introduction

A motivation model (e.g. represented as a Business Motivation Model[1]) is an important artefact in an organizational context, as it encodes organizational intent and guides the maintenance of its service capabilities and ultimately all service capabilities of an organisation should be traceable back to (and justified by) elements in the motivation model.

The ever-changing business context requires organisations to constantly adapt their motivation and service representations. For example, an organisation may change its vision in terms of modifying its goals (which may render existing service capabilities superfluous) or may have to give up (or adopt) services, e.g. to remain compliant with changing regulations (which may result in unrealized organisation motivation). A *manual* adaption can be a time-consuming and error-prone exercise. For example, we might overlook inconsistencies or commit to sub-optimal modifications (often simply because of human cognitive limits on the space of alternative modifications we can explore). Formally (semi-)automated machinery is therefore desirable.

Existing work however focuses on identifying and verifying the functional relationship between the motivation- and service level (e.g. [2], [3], [4]), or techniques for handling changes at the service level (e.g. [5], [6]). Little work has been done in providing means to identify the consequent impact of changes made at the motivation level to the service level. Previous work (e.g. [7], [8]) has been done in providing formal machinery for dealing with changes at the motivation level. Formal requirement engineering approaches such as KAOS [9] and FormalTropos[10] focus on verification. Zowghi et al. [7] provides an operator which maps one goal model (represented by a default theory) to another, but does not consider

hierarchical relationships among goals nor their relation to services. Ernst et al. [8] address the above, but in their formalization goals are required to be atomic statements (as opposed to formulas in an arbitrary language, as is the case in our work¹). which does not permit the kind of analysis we provide.

In this paper, we propose a novel framework for handling changes at the motivation level and highlighting consequent impacts to the service level. Section 2 covers preliminaries and introduces a running example. Section 3 introduces a motivation-service (F-hyper-) graph and introduces a hierarchical entailment relation with the graph. Section 4 formalize a change operator which is recommended for (but is not restricted to) changes driven by compliance, i.e. where a motivation model needs to be adapted to meet compliance obligations. This is important in the service context, since non-compliant motivation models give rise to non-compliance at the service level. Finally, we conclude and outline future work in section 5.

2 Preliminaries

Business Motivation Model: A Business Motivation Model (BMM)[1], as standardized by OMG, is a hierarchical representation of organisational *motivation*. In a BMM model the *vision* is the most abstract element and attained via a set of goals. A *goal* is a statement about a state or motivation the organisation seeks to maintain or bring about. A *strategy* represents an accepted course of action to achieve the ends (i.e. the goals and vision). Strategies, are implemented via *tactics*, which are narrower in scope. Figure 1 (left side) shows some examples of a vision, strategy, goal and tactic and their hierarchical representation in a BMM (right side). In the BMM of Figure 1 (right side), the

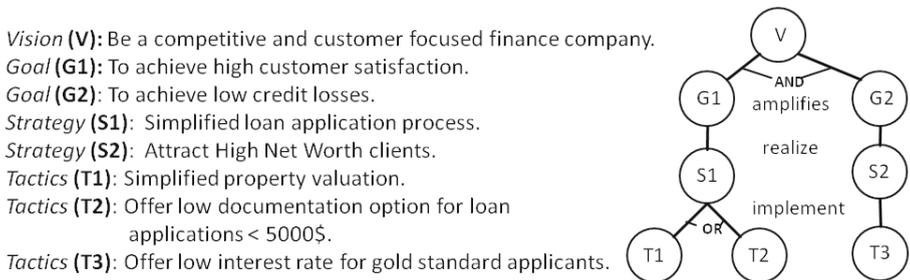


Fig. 1. A Business Motivation Model (right) and its description (left)

“AND” connection between goals “G1” and “G2” relating them to parent vision “V” indicates that *both* goals need to be realized to bring about the vision; the “OR” connection between tactics “T1” and “T2”, and strategy “S1” denotes that either of the tactics can be pursued to implement the strategy.

Service Representation: Semantic service description standards like WSDL-S and OWL-S enforce a degree of rigour and detail in how services are modelled,

¹ We only require a finitely generated formal language with an associated entailment relation (\models) relating sentences in the language.

which in turn supports more sophisticated analysis. Most of these standards support the specification of pre- and post-conditions, which we leverage in our framework. Our current work leverages WSDL-S, but, as noted above, other semantic service standards could equally well be used. For simplicity of exposition, we use the syntax of classical logic in representing pre- and post-conditions (effects) in the following. Service descriptions are maintained in a *service catalogue*. In our running example will refer to the following services in the service catalogue of a hypothetical organisation (for brevity we only provide their name and effect): *Simplified Property Valuation* (Se1): $\text{PropertyValued} \wedge \text{SimplifiedPropertyValuation}$; *Small Loan Service* (Se2): $\text{LoanApplicationHandled} \wedge \text{LowDocumentation}$; *Premium Loan Service* (Se3): $\text{IncomeStatementReceived} \wedge \text{LoanApplicationHandled} \wedge \text{GoalStandardDiscount}$.

3 Formal Representation

We now describe a formal representation of BMM models and service catalogues that permits the application of (semi-) automated machinery for managing change of various kinds. The formal representation is based on a *motivation library* and *Motivation-Service graph*, which are described in detail as follows.

Motivation Library: A motivation library is a domain specific collection of feasible motivations, i.e., motivational elements which are feasible to realize². All elements of an organisation's BMM are part of the library, but the library may contain additional elements. We require each motivational element to be represented by a formal assertion of the language \mathcal{L} . In other words, the library is simply a set *LIB* of assertions, such that $LIB \subseteq \mathcal{L}$. The assertions provide a (machine understandable) description of the respective motivation element. Within the motivation library, we refer to the motivational elements that an organisation desires to (i.e. would like to) adopt to its BMM as $LIB^{des} \subseteq LIB$. The motivational elements the organisation is actually *committed to*, is given by the elements (denoted by their formal assertion) of the BMM model. Naturally, an organisation should be committed to all desired motivations. However, this may not always be possible. An organisation may not commit to bring about a desired motivation m , because m may be *inconsistent* with other (desired) motivations, which have been accorded higher preference, or m may be *infeasible* to pursue with other desired motivations (e.g. doing so would cause a compliance violation). Maintaining a motivation library has several advantages: it supports reuse and avoids unnecessary loss of information (this has also been pointed out by e.g. [7]). In other words, motivations which are currently infeasible, inconsistent, not desired or justified (by a desired element), are retained in the library in anticipation of future situations where these might be adopted (e.g. when they become feasible again, or the source of inconsistency goes away due to other changes). The list below provides the assertions for the respective BMM

² Note that some states of affairs might be highly desirable for an organization, but not feasible to achieve.

elements of Figure 1, as well as the strategy S3 and the tactic T4 (which do not appear in the current BMM, but might have appeared in a previous one), as part of the library. Note that the assertions may be arbitrarily detailed, but are kept simple for ease of exposition.

V: Competitive \wedge CustomerFocused	G1: HighCustomerSatisfaction
G2: LowCreditLoss	S1: SimplifiedLoanProcessing
S2: HighNumberOfHighNetWorthClients	T1: SimplifiedPropertyValuation
T2: LowDocumentation	T3: GoldStandardDiscount
S3: StrictCreditAssessment	T4: IncomeStatementReceived

Motivation-Service Graph: We use an acyclic F-hypergraph to represent an Motivation-Service Graph. An *F-hypergraph* (given by a set of vertices V and edges E) is a generalization of a simple (directed) graph, which allows edges with a single source vertex but (potentially) more than one target vertices. An F-hypergraph is acyclic, iff there does not exist a path in the graph for which the “start”- and “end”- vertex are the same. A formal definition can be found in [11]. A *Motivation-Service-graph* (MS-graph) is a *labelled* acyclic and F-hypergraph representation of the BMM elements, services (from a service catalogue) and their hierarchical relationships. In an *MS-graph*, each vertex is either associated with an element of the motivation library or the service catalogue. Hereafter, we will also assume the ability to refer to the type of each element of the motivation library (via the type of associated vertex). Vertices associated with services will refer to the service post-conditions. We use edges $e = (x, Y)$ to denote an AND-relation between a vertex x and a set of sub-vertices Y (note that edges represent *refinement* relationships between BMM elements or *realization* relationships between BMM tactics and services). For example, the edge “e1” of the MS-graph in Figure 2 (left hand side) is an AND-relation. In an OR-relation we use edges $e = (x, Y)$ and $e' = (x, Y')$ to denote that the sets of vertices $Y \subset V$ and $Y' \subset V$ are distinct refinements or realizations of x . For example, the edge “e4” and “e5” of the MS-graph in Figure 2 (left hand side) are two distinct relations. Note that, if we had defined the relationship between a goal and each of its refinements (realizations) individually (as we would be obliged to do in a simple graph), we would not have been able to distinguish between goals belonging to alternative refinements (realizations). We believe that our formalization addresses many of the deficiencies in the way AND/OR (goal) graphs are formalized (most ignore the fact that such graphs are in fact hypergraphs and that AND edges are in fact F-hyperedges).

In addition, a *background knowledge base* $\mathcal{KB} \subseteq \mathcal{L}$ is used as an encoding of domain and organisation specific knowledge (which for example could be represented in RuleML). The \mathcal{KB} may contain the knowledge that a low loan processing time results in high customer satisfaction (e.g. $\text{LowLoanProcessingTime} \rightarrow \text{HighCustomerSatisfaction}$), or that accepting credit applications with low documentation (i.e. without an income statement, etc.) is considered not to be a strict credit assessment (e.g. $\text{LowDocumentation} \rightarrow \neg \text{StrictCreditAssessment}$). The following rules are also considered in our example: $\text{HighCustomerSatisfaction}$

→ CustomerFocused, StrictCreditAssessment → LowCreditLoss, LowCreditLoss → Competitive, LowDocumentation → SimplifiedLoanProcessing, SimplifiedPropertyValuation → SimplifiedLoanProcessing, HighNumberOfHighNetWorthClients → LowCreditLoss, GoldStandardDiscount → HighNumberOfHighNetWorthClients, IncomeStatementReceived → StrictCreditAssessment.

Although we acknowledge that maintaining a formal representation of domain knowledge can be laborious, it should be emphasized that such an exercise has advantages beyond this paper (e.g. it forces stakeholders to make precise their knowledge, assumptions and terminology and thereby helps to highlight potential inconsistencies and their resolution by coming to a shared understanding).

Wellformed Motivation-Service Graph: We refer to an MS-graph that does not contain alternative refinements (or decompositions) as an *AND-MS-graph*. Any *general MS-graph* (which may contain alternative refinements), can be represented as a set of distinct AND-MS-graphs. We use the function $\Delta_{AND}(G)$ to denote all maximal AND-MS-graphs that are sub-graphs of a given MS-graph G . For example, the MS-graph of Figure 2 (left hand side) has *two* AND-MS-(sub) graphs (one includes T1 and Se1, the other includes T2 and Se2). In the following, we refer to a vertex v as a root vertex, iff there does not exist a (hyper-) edge e in the MS-graph, such that v is an element of the targets of e . An AND-MS-graph as wellformed if it satisfies the following properties.

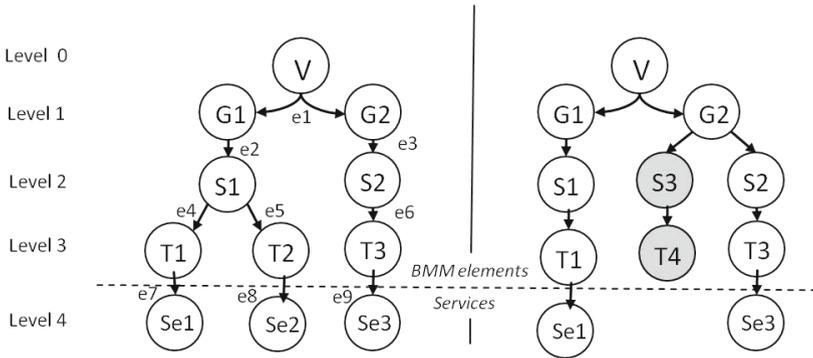


Fig. 2. The original MS-graph (left side) and its modification (right side)

(1) All root vertices are *desired*, of type “vision” and can only point to vertices of type “goal”, which can only point to vertices of type “goal” or “strategy”. A vertex of type “strategy” can only point to vertices of types “strategy” and “tactic”, which can only point to vertices of types “tactic” and “service”.

(2) The conjunction of the assertions associated with all vertices must be consistent with the domain knowledge base, since the organisation intends to concurrently realize all of them (i.e. there is no sequencing knowledge encoded that might suggest that one should be achieved before another).

(3) All target vertices of any edge must *minimally entail* the source vertex of the respective edge. We have adapted this property from [9]. Observe that in our running example the effects of service Se1, Se1 and Se3 respectively entail the tactics T1, T2 and T3 and are hence part of the MS-graph.

A *general MS-graph* is well-formed iff all $\mathcal{G} \in \Delta_{AND}(G)$ are well-formed. In our example, the strategy S3 and tactic T4 (part of the motivation library) do not participate in the MS-graph of Figure 2 (left hand side), as doing so would result in a derivable AND-MS-graph which is not wellformed (i.e. inconsistent) and the organisation has given precedence to the other elements.

k-Level Entailment: We now define a hierarchical entailment relation for MS-graphs, called *k-level entailment*. *k-level entailment* permits us to answer whether a particular assertion is derivable (or not derivable) from a particular level onwards in the MS-graph. We define *k-level vertices* as a subset of MS-graph vertices, such that for each element v , there exists a path of length k from a root vertex to v , or v is a leaf vertex and there exists a path from a root vertex to v with a length less than k . The root vertices themselves are 0-level vertices. For example, Figure 2 indicates the level of each vertex.

For an *AND-MS-graph* \mathcal{G} , an assertion α is *k-entailed*, iff all *k-level vertices* together with the \mathcal{KB} entail the assertion and there does not exist a lower level (lower in terms of the value of k) in the MS-graph for which this is the case, i.e. k is the “earliest” (i.e. lowest) level at which the assertion is entailed. We use $\mathcal{G} \models_k \alpha$ to say that \mathcal{G} *k-entails* an assertion α . For example the assertion **LowDocumentation** is entailed at level 3. From the definition of a wellformed MS-graph it follows that, if an assertion α is *k-level entailed*, then α is entailed in all levels i that follow after k . We refer to this as α being *i-consequence*. For a general MS-graph G an assertion α is *strongly k-entailed* (denoted by $G \models_k^{str} \alpha$) iff all AND-MS-graphs derivable from G have α as an *i-consequence* and $k \leq i$. Conversely, we say that a general MS-graph G *does not entail* an assertion α (denoted by $G \not\models^{str} \alpha$) iff there does not exist an AND-MS-graph derivable from G that entails α at any k .

4 Maintaining Motivation-Service Graphs

In this section, we show how to maintain an organisation’s motivation model (i.e. the BMM) and service capabilities (i.e. the catalogue) in their conjoint MS-graph representation. There can be many change drivers. Organizations change their motivations to respond to dynamic business contexts, e.g., by the addition of motivation elements (via an *add-vertex* operator) or the removal of motivations (via a *remove-vertex* operator³). An MS-graph may have to be modified to meet compliance obligations, e.g., to ensure that an assertion is entailed by the graph (via an *entail-assertion* operator) or isn’t (via a *not-entail-assertion* operator). Due to space constraints, we only describe the “not-entail-assertion”

³ The modification of a motivation can be viewed as a removal of the prior version followed by the addition of its modified version.

operator, which is also the most complex of the four types mentioned above). The properties of the *not-entail-assertion* operator are inspired by the AGM logic of theory change [12] and belief contraction postulates. A key insight from this framework is the need to minimize change to a body of knowledge encoded in an artefact such as an MS-graph. We therefore require means to assess the extent of change, which might be encoded in a notion of proximity between two MS-graphs.

Motivation-Service Graph Proximity: Given MS-graphs G , G' and G'' we say that $G' <_G G''$ if G' is “closer” to G than G'' . This relation $<_G$ may be defined in various ways. From a *graph theoretic* perspective, we may view G' to be closer to G if it has more vertices and edges in common with G . This intuition places an equal weighting on all elements of the graph. However, this is not sensible for MS-graphs, since some vertices justify others and hence should be given precedence (e.g., a strategy is justified by the goal it aims to bring about). We prefer an MS-graph that preserves the original graph up to level $k + 1$ over one that preserves the original graph up to level k . On the basis of this argument, one possible intuition for defining $<_G$ could be *k level set intersection cardinality*. MS-graph G' is preferred over G'' if G' shares more vertices with the original MS-graph G at level 0. In case of a tie, G' is preferred over G'' if it shares more vertices with G at level $k + 1$, and so on.

“Not Entail Assertion” Operator: The “not entail assertion” operator (denoted by \ominus) minimally modifies an MS-graph to obtain one where an assertion is strongly not derivable. A key application is in compliance management. In the running example, if new regulations require that each loan must be backed up by sufficient documentation, the organisation must ensure that the assertion $\alpha = \text{LowDocumentation}$ is not derivable from the MS-graph (in the example α corresponds to a single assertion but may in general be a consequence of a set of assertions). Our formalization can handle this. We offer the following normative properties for the “not entail assertion” operator.

1. *Wellformedness:* For a wellformed MS-graph G , $G \ominus \alpha$ is wellformed.
2. *Success:* For a wellformed MS-graph G where $G \ominus \alpha = G'$, $G' \not\vdash^{str} \alpha$.
3. *Vacuity:* For a wellformed MS-graph G where $G \not\vdash^{str} \alpha$, $G \ominus \alpha = G$.
4. *Desire-inclusion:* For a wellformed MS-graph G where $G \not\vdash^{str} \alpha$, there does not exist a desired goal in the library, such that if it were to be added to G the resulting MS-graph would satisfy the properties 1-3 (i.e. the resulting MS-graph should include as many desired goals as possible)
5. *Minimal-change:* For a wellformed MS-graph G where $G \ominus \alpha = G'$, there does not exist a G'' which satisfies the 4 properties above and $G'' <_G G'$. In other words, the operation should return the MS-graph which satisfies the above conditions and is closest to the original graph.

Definition 1. *Given a wellformed MS-graph G and an assertion α , let the operation $G - \alpha$ return a set, such that each $G' \in (G - \alpha)$ satisfies the properties (1)-(5). We define $G \ominus \alpha = s(G - \alpha)$, where s is a selection function which returns an element in $G - \alpha$.*

\ominus represents a class of operators, parameterized by the proximity relation $<_G$ and the selection function s . The removal of a element from G might permit the addition of previously inconsistent desired motivations from the library to G . In our example, the vertices T2 and Se2 of the MS-graph make α derivable, which might lead to their removal. This may permit the inclusion of the previously inconsistent library elements S3 and T4 to the MS-graph.

The MS-graph in Figure 2 (right side) denotes the outcome of the “not entail assertion” (with $\alpha = \text{LowDocumentation}$) operator (instantiated by k -level intersection cardinality) with respect to the original MS-graph (right side)⁴. Tactic T4 can be included in the MS-graph but it is not realized by a service (or combination of services). This would have to be flagged to the user, who could then decide to adopt the required functionality, drop the unrealized options from MS-graph, or leave the model unchanged. Furthermore, the service Se2 is not part of the MS-graph any more (and flagged as such to the user) since it is part of the service catalogue, but does not contribute to bringing about any organisational motivation.

5 Conclusion and Future Work

In this paper, we have proposed a formal framework to deal with changes at the motivation level and assessing their impact to the service landscape. An operator class (the “not entail assertion” operator) for compliance driven changes was introduced. We have developed a prototype implementation of the operator (using A* search technology) for propositional logic, which suggests the feasibility of our approach. Future work would involve investigating the scalability of our approach and the deployment of local search versus global search techniques in real world scenarios. Formalizing additional operator classes which capture other situations of change is also part of our future work.

References

1. OMG: Bmm, <http://www.omg.org/spec/BMM/1.1/PDF> (2010)
2. Jokhio, M.S.: Goal-based testing of semantic web services. In: ASE, pp. 707–711 (2009)
3. Chopra, A.K., Dalpiaz, F., Giorgini, P., Mylopoulos, J.: Modeling and Reasoning about Service-Oriented Applications via Goals and Commitments. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 113–128. Springer, Heidelberg (2010)
4. Lu, Q., Tasic, V., Bannerman, P.L.: Support for the Business Motivation Model in the WS-Policy4MASC Language and MiniZnMASC Middleware. In: Kappel, G., Maamar, Z., Motahari-Nezhad, H.R. (eds.) ICSOC 2011. LNCS, vol. 7084, pp. 265–279. Springer, Heidelberg (2011)
5. Andrikopoulos, V., Benbernou, S., Papazoglou, M.: On the evolution of services. IEEE Transactions on Software Engineering (2011)

⁴ In our example all elements in the motivation library are desired, and recall that the desire inclusion property has primacy over the minimal change property.

6. Hibner, A., Zielinski, K.: Semantic-based dynamic service composition and adaptation. In: 2007 IEEE Congress on Services, pp. 213–220 (2007)
7. Zowghi, D., Ghose, A.K., Peppas, P.: A Framework for Reasoning About Requirements Evolution. In: Foo, N.Y., Göbel, R. (eds.) PRICAI 1996. LNCS, vol. 1114, pp. 157–168. Springer, Heidelberg (1996)
8. Ernst, N., Borgida, A., Jureta, I.: Finding incremental solutions for evolving requirements. In: RE (2011)
9. Darimont, R., Van Lamsweerde, A.: Formal refinement patterns for goal-driven requirements elaboration. ACM SIGSOFT Software Engineering Notes 21(6) (1996)
10. Fuxman, A., Liu, L., Mylopoulos, J., Pistore, M., Roveri, M., Traverso, P.: Specifying and analyzing early requirements in Tropos. Requirements Engineering 9(2) (2004)
11. Gallo, G., Longo, G., Pallottino, S., Nguyen, S.: Directed hypergraphs and applications. Discrete Applied Mathematics 42(2-3), 177–201 (1993)
12. Alchourrón, C., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. The Journal of Symbolic Logic 50(2) (1985)