Modeling i*, e3Value and Service Choreographies for Virtual Organization Formation

Mohammad Hossein Danesh
Department of Computer Science, University of Toronto
40 St George Street
danesh@cs.toronto.edu

ABSTRACT
Collaborations among organizations is gaining more attraction as businesses need to respond to more dynamic requirements faster and more efficiently. In order to facilitate agile response to market demands, organizations are using partnerships and collaborations to share their resources and competitive advantages. Therefore computer supported organized gathering of organizations which is known as Collaborative Networked Organizations (CNO) and the ability to form such initiatives is an asset. As business opportunities arise multiple organizations form temporal collaborative settings called Virtual Organizations (VO). The ability to setup and negotiate collaboration in a VO is called VO formation and is considered an asset in today's business environment. In this paper we discuss the support of three level modeling approache that facilitate VO negotiation and setup. The three levels consists of (1) a strategic goal model that illustrates intentional strategic dependencies with the i* modeling notation, (2) a value network that presents value offering of different partners and is considered as the business model of the collaboration, and (3) a choreography process model that focuses on interaction between different participant of the VO. In this paper the concentration is on how these modeling types relate to each other and why they are needed in different steps of VO formation. The logic of transforming the three model types in a collaborative context is discussed and is implemented as operators in Model Management Tool Framework (MMTF). Two case studies are provided to illustrate how the modeling steps and transformations are performed.

Keywords
Virtual Organization – Goal Modeling – Value Network – Service Choreography

1. INTRODUCTION
Maintaining competitive advantage in today's business environment is more difficult. Customer needs and technology is changing more often and organizations need more dynamic business models, business strategies, governance principles, business processes and technological capabilities [1, 2]. As a driver to create advantage, organizations have been motivated to share their resources and advantages to create new opportunities. Virtual Organization (VO) is defined as "a dynamic, temporal consortium of autonomous legally independent organizations which collaborate with each other to attend a business opportunity or cope with a specific need, where partners share risks, costs and benefits, and whose operation is achieved by coordinated sharing of skills, resources and competencies and whose interactions are supported by computer networks" [3].

As Service Oriented Architecture (SOA) has gained more attention and organizations have adopted SOA infrastructure, in [4] the authors have proposed a federated solution based on SOA that facilitates execution and synchronization of business processes between organizations. The infrastructure is proposed to support VO formation, process and performance management while supporting different collaboration topologies. The service zone interaction model proposed by the authors facilitates policy enforcement by each partner and allows them to share their processes, business rules, performance indicators, tasks and infrastructure resources. In their research on Service Oriented Virtual Organization (SOVO) the authors propose how to design collaborative business processes in [3] and how to measure performance of such business processes in [5]. From the SOVO perspective, everything within a collaborative setting such as virtual organization is a service and is distributed through partners service zones. The use of value network and service choreographies to form and manage this collaborative environment is described in their work as service oriented process and performance management frameworks.

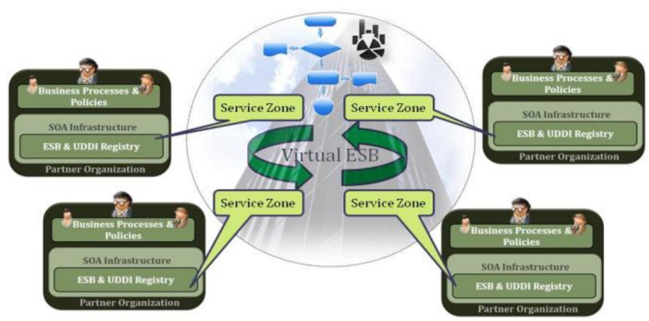

Figure 1; SOVO Service Zone Interaction Model [3]

1.1 The i* Goal Models
Goal models have gained attention to model strategic intentions of enterprises. The i* modeling framework focuses on modeling the socio technical aspect of organizations. The idea behind i* goal modeling is to capture actors goals and dependencies in the early stages of requirement engineering to explore alternative development and design more aligned solutions to organizational needs [6]. The i* technique enables modeling of causal relationships among involved actors. The strategic dependency model of the i* framework focuses on alliances of actors and their dependencies on each other. The strategic rational model focuses on analyzing the means to a goal within the boundary of an actor and facilitates the identification of the cause of a dependency. The i* framework has 8 major components; Actors, Goals, Tasks, Resources, Soft Goals, Dependencies, Means and Decompositions. An Actor is an entity that has certain intentionality and expresses the intentionality by its goals. Goals can be further drilled down to other elements known as, tasks that are activities with known specification, resources that are entities that need to be provided and sub goals. A task can also be

decomposed further to sub tasks, goals and underlying resources. Dependency in i* refers to an actor relying on another actor to perform a task, provision a resource or satisfy a goal. The main difference between a goal and a tasks is that goals in i* do not have further specification and therefore can be satisfied in any way but tasks are activities with specific specification and outcome. A soft goal refers to non-functional attributes and dependencies [6]. Figure 2 shows the i* modeling elements graphical representations.

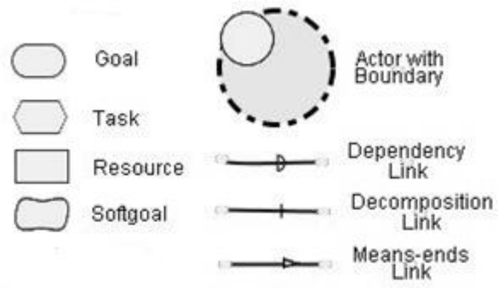

Figure 2; The i* Modeling Elements

1.2 Value Networks and e3Value

There are different ways to model and design collaboration between organizations. Traditionally when organizations gather together to produce value added services, they started by engineering their processes using function or process oriented models. However these methods usually have shortcomings to align with the overall value co-production of collaborative networked organizations [7]. Therefore in this research we have used value networks to model business value creation and tracking. Business Value Networks "are ways in which organizations interact and share values forming complex chains including multiple providers and administrators to derive increased business value" [8]. This helps the collaborative organizations to identify service participants and their value expectations and value exchange rationales.

e3Value is an ontology defined for modeling value networks which is further discussed in [9]. We have used the e3Value notation for modeling value networks. A glossary of e3Value notations and a sample value networks is depicted in Figure 3.

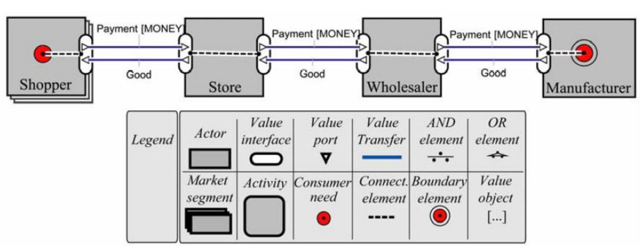

Figure 3: A Sample Value Network modeled by e3Value [9]

1.3 Collaborative Processes and Service Choreographies

Choreography "formalizes the way business participants coordinate their interactions. The focus is not on orchestrations of the work performed within these participants, but rather on the exchange of information (messages) between these participants" [10]. Choreography consists of a set of activities that involves more than one participant and focuses on the sequence of messaging and communication between the participants. In [3] the authors have proposed the use of choreography to specify each party's role and activities, and the sequences of service invocation for virtual organizations. The choreography in this setting serves as an agreement between the participating business partners in their collaboration. Modeling and negotiating choreographies between parties in a VO is used for setting contracts and service level agreements between partners. In Figure 4 the notation for choreography modeling in BPMN v2 is depicted. In this research due to SOVOs extensive use of services we choreograph partner services and therefore the term service choreography is also used.

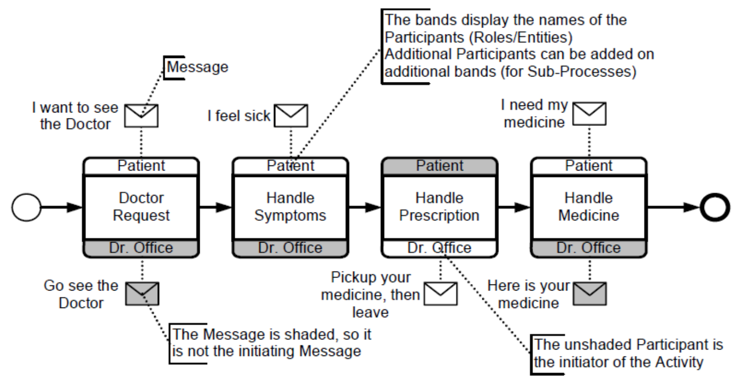

Figure 4: BPMN Choreography Modeling Notation [10]

2. USE OF i* AND e3VALUE FOR VO FORMATION

When modeling collaborative network organizations and more specifically virtual organizations, it is not recommended to focus on long term strategic alliances and market analysis due the temporal nature of these setups. Therefore Value Networks are proposed for modeling business collaboration and networked organizations business models [1,2]. A Value Network is a network of enterprises that jointly creates and distributes objects of economic value [12]. Use of value network as a business model for collaborative organizations, helps to identify each partners offering, competencies and their contribution in the overall value creation process [13]. In this research we have used e3Value as the modeling language for value networks and we perform a feasibility study (A study that shows if a certain project or business model is feasible) on the model based on e3Value modeling tool described in [9]. Value networks provide a high-level view of the activities taking place in and between organizations by identifying actors, resources, information and services being exchanged between actors and the value creation chain [11]. An analysis on each actor in a value network and the overall value offering and receiving of the actor should be a positive summation in order for the collaboration to be feasible [9]. In other words, a value network focuses on the ''what'' of a collaborative business. When modeling a VO formation not only modeling what the business is and if it is a feasible collaboration for all parties involved or not, is not enough [13]. We also need to model how the value exchange happens and how it is produced. Process models deal with operational and procedural aspects of business communication, including control flow, dataflow and message passing [11]. In this research we use service choreographies and BPMN notation to illustrate collaborative business processes. It is important that we can align the business collaboration to the higher level business model which in this research is the value network. In [13] the service choreography is an important artifact of the process and performance management framework for service oriented virtual organizations. Final participating partner negotiations, lower level business processes and services design, performance indicator hierarchy definition, and Service Level Agreements (SLAs) aggregation rely on the service choreography [2,6]. Therefore it is important to align the service choreography to the business model and have traceability between the two business models. A third aspect of a business collaboration which illustrates why a business is beneficial to each partner is also critical. We need to know why a certain configuration of partners is beneficial, why a certain value constellation is proposed, what alternatives in producing collaborative values exists, and finally what is the intention of each collaborative party in participating to the collaborative organization [7,8]. In this research we use i*, to help in clarifying interests, intentions, and strategies of different stakeholders in a collaborative organization. Figure 5 shows how these three levels of modeling artifacts interact with each other to illustrate collaborations in collaborative networked organizations. This work is closely related to the work presented in [3] and [5] and complements those frameworks.

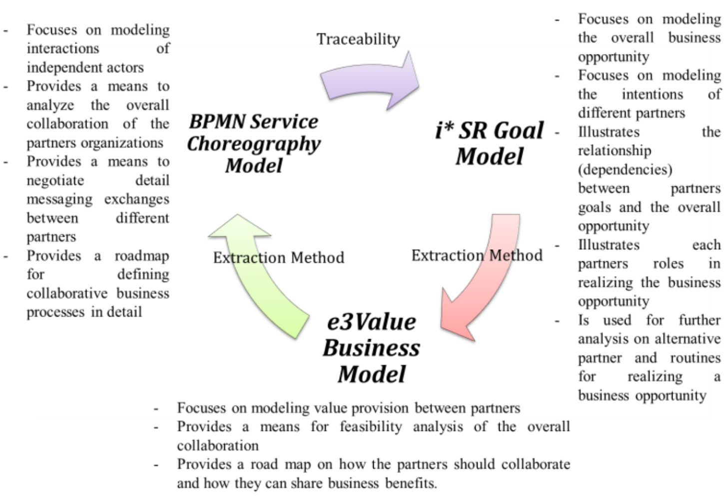

Figure 5: Three Level of Modeling collaborations for a CNO

When modeling collaborations of a CNO you might go through the cycle presented in Figure 5 a few times. Using goal models to identify potential partners, building a value network and analyzing each partners roles and benefits in the collaboration and finally negotiating a choreography model that satisfy all partners in the collaboration will not happen overnight. This modeling cycle will help CNOs achieve this goal through multiple cycles of goal matching, value constellation and choreography negotiation. This cycle can also be used when analyzing a change within each of these levels. For example if a price of a service offering of a specific partner changes it effects the value network feasibility study and each partners share in the overall collaboration. The trace of the changes back to the goal model can help develop alternatives. As another example a changing one of the partners in the goal model may result into different value network and choreography. These relations between different layers of the models will help analyze these changes and their effects to help decision making for a CNO.

3. OPERATORS AND THEIR CHARECTERESTICS

This section describes how relations between the three modeling levels presented in Figure 5 are defined. Two operators are three relations are defined to achieve this goal. The first one focuses on extracting a value network from the dependencies defined between different actors of an i* goal model. The second operator generates a graph that illustrates modeling steps of the service choreography and the relation of those steps to the value network and i*goal model.

3.1 Creating 3Value Business Models from i* Goal Model

There are a few works that focus on transforming goal models to e3Value. The authors in [16] describe how designing a collaborative e-services can be facilitated by e3Value business model and an i* goal model. They focus on the relationship between the two models and how to extract value networks from goal model. This paper also discusses how i* can be used to generate alternatives to make a certain collaboration profitable and uses e3Value to do the feasibility study and determine whether a specific collaboration is profitable or not. In [11] the authors provide a method to transform a strategic goal model to a value oriented business model and extract a hierarchy of implementation services (top level and low level services) from the value network. The main difference between the work of [11] and what we need in this research is that their work has a general view on service decomposition from value networks and we focus on collaboration and interaction of different entities. To be more specific, Andersson and his colleagues define a set of transformation rules that focus on balancing each value interface input and output while we only care the overall stability of each node and the value network. In this research we are concerned about the flow of values between different partners and how they choreograph their services while in [11] the focus is on breaking down the values to lower level services. The main idea in both the mentioned approaches in transforming an i* goal model to e3Value is the same. The transformation has two basic rules which we implement and at this stage we keep the operator as simple and general as possible.

Rule1: Each actor in the i* goal model is transformed into an actor in the e3value model

Rule2: Each dependency in the i* goal model refers to a value transfer in the e3value model. Each i* dependency has a model element as the dependum and the owner of that model element which is an i* actor is called the dependee. The actor that receives the dependum in a dependency is called the depender. However dependencies can happen in lower levels which means a specific model element of the depender actor can be the receiver of the dependum and therefore that element is the depender (examples of this can be found in the i* goal model presented in [11]). However when transforming a dependency from i* to e3value, a value transfer is created with the dependee actor in e3value as the source and the depender actor in e3value as the destination and the dependum as the value being exchanged.

Condition1: If the value transfer with the same depedum and the same depender actor (this happens when more than one element of a specific actor depend on the same depundum) exists do not create a duplicate value transfer.

Relation1: Each Actor in i* has a one to one relationship with an actor in e3Value.

Relation 2: As a result of this operator a relationship is defined that will be later used to trace elements between the two models. Consider the set of dependencies in the i* model as $D=\{d_1, d_2, \dots, d_n\}$ and the resulted value transfers as $V=\{v_1, v_2, \dots, v_n\}$, then each value transfer has a relation to a subset of D defined as $DofV_i$ where each member d_k has the same depender actor, dependee and dependum to v_i but can have different depender elements.

3.2 Providing a Roadmap for Defining Service Choreographies

There have been different attempts to derive service choreographies from value networks. Among these attempts authors in [6] and [7] based the service choreography description on value inter-dependencies in the value networks. In [6], the authors start by decoupling the value network into value chains with loose or no relation to each other. The service choreographies are then extracted from sets of values and finally they connect different sets of service choreographies together. The downside in this method is when we have a peer-to-peer network where decoupling will not be an option due to inter-dependencies between values. In this research we use a similar approach based on value dependencies but we do not base our choreographies on sets of decoupled value chains. We propose the following steps for extracting choreographies from value networks:

1. Assign each value transfer in the network an ID therefore a set of values which can be defined as $V=\{v_1, v_2, \dots, v_n\}$ will be resulted.

2. Create the following matrix must be formed. In the presented matrix v_i's are values of the set V. p_{ij} is 1 if v_j has a dependency on v_i in a sense that v_j cannot be performed as it should, unless v_i is performed otherwise p_{ij} is 0. Note that this dependency needs to be a direct dependency which means if $p_{ij}=1$ and $p_{jk}=1$ but there is no direct relation between v_i and v_k then $p_{ik}=0$.

$$M = \begin{array}{c} \\ v_1 \\ v_2 \\ \vdots \\ v_n \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & \dots & v_n \end{array} \\ \begin{bmatrix} 0 & p_{12} & \cdots & p_{1n} \\ p_{21} & 0 & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} & \cdots & 0 \end{bmatrix} \end{array}$$

.

3. The presented matrix can be extracted from the dependency relation of the previous operator. p_{ij} is 1 if there exists $D_k \in DofV_i$ and $D_h \in DofV_j$ such that depender element of d_k is the same as dependum of d_h.

4. For each value in V, count its successive values as (SV_i):

$$SV_i = \sum_{k=1}^{n} p_{ik}$$

5. For each value in V, Calculate its depth of influence (DF_i) which is equal to the following formula:

$$DF_i = \left\{ SV_i + \sum_{j=0}^{n} DF_j \,\middle|\, where\ p_{ij}\ in\ matrix\ M\ is\ equal\ to\ 1 \right\}$$

Where $DF_i = 0$ if SV_i=0

6. Rank the values based on DF_i.

This depth of influence gives us an idea of how important each value transfer is in a sense that it effects the other exchanges the most. The idea of this kind of analysis comes from critical path analysis in project management. According to this ranking we build a directed graph which we have named dependency graph. The nodes of the graph are the value transfer and are shown by their ID and the links are all the dependencies between the values as presented in the matrix. An example of the dependency graph is presented in Figure 6. The case study is presented in [5]. The red numbers in this graph are showing the service choreography modeling priorities.

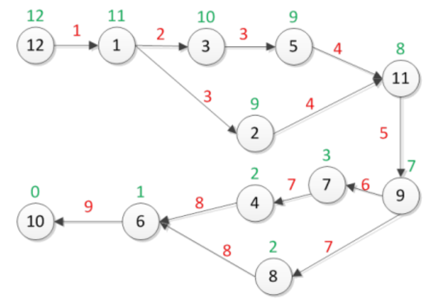

Figure 6: An Example of Dependency Graph [5]

Depending on the context and the level of the values defined the modeler can use these priorities. For example if the virtual organizations node in the value network is present to the final costumer the choreography components would be the same as the following graph. The first case study presented in this paper is as such. But if the virtual organization node is not present to the costumer and at each stage one of the parties in the collaboration interacts with the costumer, the components in the choreography are not the same as the ones in the dependency graph as is the case in the example provided in [17]. Another factor that affects the use of this dependency graph is the level of detail of the value modeling. In [5] the authors recommend that value transfer should represent business processes of partners. In that case this nodes can represent a service choreography component but if the values are modeled at a higher level as in the example provided in [14], the choreography components are not the same as the value transfers. However in most cases this graph will give us a roadmap on the sequence of negotiation. The limitation to this approach is that it assumes that all these values have the same effect and a value is not more valuable than others while this may not always be the case. Although this could be a nature of focus when negotiating a service choreography, in a service oriented environment where you build new services and processes using existing resources, the focus is on how to assemble these services and the service choreography focuses on this aspect therefore the authors have not focused on the degree of influence of each value.

4. IMPLEMENTATION IN MMTF

In this section I am going to present my implementation of the proposed modeling approach and the defined transformations with Model Management Tool Framework (MMTF) presented in [18]. I have defined three model types that I am going to describe in the first part of the section. In the second part, I am going to present the implementation of the theory I presented in section 3 on the mentioned transformations.

4.1 Meta Models

As discussed in previous sections, in this research we deal with three types of models: I* Goal models that facilitate strategic reasoning, e3Value that provide a value network business model and the BPMN service choreography. However in my implementation I have not implemented all the elements of these modeling notations. I have defined a subset of the i* and e3Value that is needed to implement the transformation and not all the details of the modeling notation are included. I also do not provide a service choreography model. This implementation only supports the modeling approach up to the generation of dependency graph.

In Figure 7the meta model of the implemented i* is presented. The "SRGoalModel" is the root element. The root contains a set of actors. Each actor contains a set of elements which are one of the three types of Goal, Resource or Task. Each goal can have sub elements as means to achieve the overall goal and each task can be decomposed to sub elements. A model also contains a set of dependency that has an element as a dependee and an element as a depender. We recommend the dependencies to be directed at the leaf nodes of each actor. A dependency at a higher level cannot be interpreted by the operators defined in this implementation.

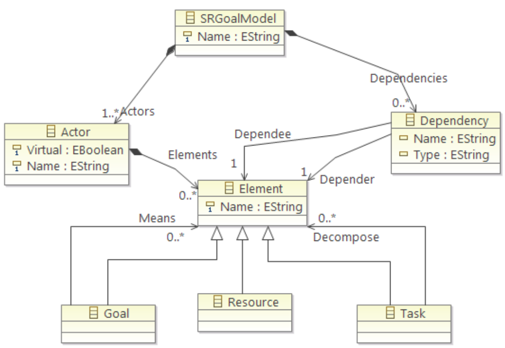

Figure 7: Implemented i* Meta Model

In Figure 8 the meta model of the implemented e3Value models is presented. Each model contains a root element named as "ValueNetwork". From the e3Value object we have modeled the actors and value transfer (or sometimes called value exchange). Each model contains a set of actors and a set of value transfers. Each value transfer refers to a source as the actor providing the value and a destination as the actor receiving a value. We have focused on this subset of e3Value because these are the objects that the research on SOVO [17] focused on.

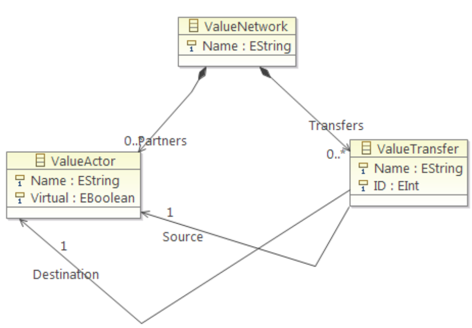

Figure 8: Implemented e3Value Meta Model

In Figure 9 we have provided the meta model for the dependency graph introduced in section 3.2. The graph consists of nodes that refer to a value transfer and have a depth of influence which is DF_i as defined in section 3.2. The links present the connection between the node and the step attribute indicates when it should be modeled.

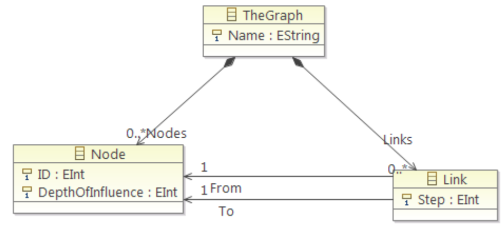

Figure 9: Implemented Dependency Graph Meta Model

4.2 Operators

4.2.1 Plain Value Network Creation

The implementation of this operator takes two models as an input, an i* goal model according to the meta model in Figure 7 and an e3Value model according to the meta model in Figure 8. The order of the inputs is not important. The operator clears the e3value model and generates the value network according to the i* goal model and also creates two relations, one which describe matching of actors between the two models types which will be referred to as the "Actor Relation", and the other for tracing the origin of value transfers to the dependencies which will be referred to as "Dependency Relation". This operator and the resulted model and relations are based on the logic discussed in section 3.1 and apply the same conditions.

For describing the characteristics of the proposed operator I am going to use the classification presented in [19]. To describe the transformation rules I would start with its domain. The source domain is an i* goal model and the meta model is presented in Figure 7. The target domain is an e3Value model with the meta model presented in Figure 8. The transformation rule does not have syntactic separation. The transformation rule is also not multi directional and does not support updating the target model. There is one application condition for this rule that is, it cannot have a duplicate value transfer with the same source and target actors and same value exchange. Therefore in such cases the operator only adds the relation between the dependency and the value transfer. The only intermediate structure created in this transformation is the relation between the dependencies and value transfers which is later transformed to the model relation "Dependency Relation". There is no use of parameterization in this transformation. This operator does not explore the whole goal model but only determines the dependency elements in the model and visits all instances but it does check the target model to identify if a value transfer with the same characteristics exists. In this case we apply deterministic location recognition. The scheduling of this operator is fairly simple. There are two rules that are applied. First it creates the actors and the actor relation, and then it moves to dependencies and value transfers. Therefore the scheduling is implicit, selects rules explicitly, iterates recursively and has no phasing. Overall the implementation of this transformation on top of MMTF is considered an operational approach. In future this operator should support soft goal and contribution elements of i* models. It should also facilitate updates on the value network and should not create the whole model from scratch

4.2.2 Create Dependency Graph with Relation

The implementation of this operator takes a dependency relation and a dependency graph model as inputs. It generates the nodes and links of the dependency graph model and creates a one-to-one relation between the nodes of the dependency graph and value transfers of e3Value model. In MMTF if you have a relation you can access the models that the relation refers to. In other implementations the operator would need the value network as an input. The current operator uses name matching to detect if a dependee of a dependency is the same as the dependum of another to declare a value dependency (refer to section 3.2 for more information). This operator assumes that elements name in i* goal model are the only means of identifying an element. Therefore the operator relies on a name match to declare if a value dependency exists (determination of value dependency is discussed in step 3 of section 3.2). This operator has two phases. In the first phase, it creates the graph nodes and links and in the second phase, it calculates the DF_i with a recursive implementation and it grantees that all nodes have been visited.

We characterize this operator according to the classification presented in [19]. The first character is the source and target domain model. The input parameters are the dependency relation and e3value model which are both results of execution of the operator presented in section 4.2.1. The target domain is a dependency graph which the meta model was presented in Figure 9. This operator is not syntactically separated and acts on both source and target domain at the same time and is implemented as a java function. The implemented operator is not multidirectional and only support dependency graph generation. This transformation only works if there is no loop in the value dependency. This means that the values dependency cannot represent a cycle. Imagine a transportation case where value one is transporting, value two is legal document preparation and the third value is providing tracking information. Now the first value is dependent to on time preparation of the second value and the second one is dependent on constantly receiving tracking information. The cycle shows a continual offering until the product is delivered. When implementing this loop the DF_i (refer to section 4 for definition) value be infinite and therefore the dependency graph cannot be sorted. This operator creates a couple of intermediate structures: the dependency matrix, an array containing SV_i and an array containing DF_i (for the definition of these constructs refer to section 3.2). There is no parameterization implemented in this operator. The operator uses a deterministic location approach. The rule scheduling is implicit, selects rules explicitly, the first phase follows a recursive pattern and the second phase follows a loop, and the rule executes in two phases. Overall the implementation of this transformation on top of MMTF is considered an operational approach

As discussed in the previous paragraph the major limitation of this operator is that it cannot handle a loop in the value dependencies. There are some ways to accommodate it. Usually dealing with this kind of loops, the modeler according to the application domain knows which value is the one at the end of the cycle and which value is the initiator of the cycle. Therefore the infinite problem can be resolved by cutting the relation between the end node of the cycle and the start node. The DF_i value calculation should start from the end node. The implementation of this requires an interactive loop detection and resolution which is left for future work. Another limitation with the current operator is that it does not track dependencies in different levels of the goal model hierarchy and assumes that all dependencies are modeled at the leaf node. Therefore if a dependency to a parent node exists, the modeler needs to create a dependency for each of the children. In future the implementation could be expanded so that the child nodes inherit the dependency of their parent.

4.3 Lessons Learned

MMTF and the way it handles model instances and model relations, facilitates the implementation of models and transformations. Creating new model relations and tracing model elements in these relations were fairly simple. The fact that MMTF treats the model relations as model types and allows further use of them beside traceability gives the user a great ability. The graphical interface of MMTF especially when presenting relations is not intelligent and requires more attention.

In designing transformations, focusing on basic transformation is powerful and provides a good general tool. But implementing more flexible options and extended support, analysis and reasoning is much more complicated. For example creating a basic transformation of an i* to e3Value requires less effort than providing a transformation that updates an existing e3Value model based on an i* or a transformation that is multidirectional. Providing support for interactive transformation that is intelligent enough to ask the modeler about a certain context or application of a model requires more effort (for example handling loops with the modeler's knowledge of the context). This could be a point for further expansion of MMTF.

5. CASE STUDIES

5.1 Mobile Operator

The first case study is developed for clarifying how this modeling approach can support virtual organization formation. It assumes a certain mobile operator and cell phone producer initiate a virtual organization to sell lees attractive phones and lower quality communication plans at a very competitive price.

In Figure 10 the goal model of this virtual organization is presented. The high level goal of the cell phone producer is to 'Balance inventory by selling less attractive phone' and it provides the underlying processes to support this goal. In return it expects to be paid for the phones sold but it also has a flexible scheduling in receiving the payments. On the other side the mobile operator wants to "Make Money out of Unused Bandwidth without Commitment" and again the operator is flexible in receiving payments. The important factor for the operator is to be able to preserve its brand image and it does not want to provide the service with the same quality therefore the new initiative (virtual organization) would have a presence to the costumer. The customer is interested in having "Access to Cheap Data Plans with Smart Phone". In Figure 10 the elements that are not associated with any of the actors' boundaries belong to the virtual organization and represent collaborative processes. The leaf elements in this example are considered as business processes.

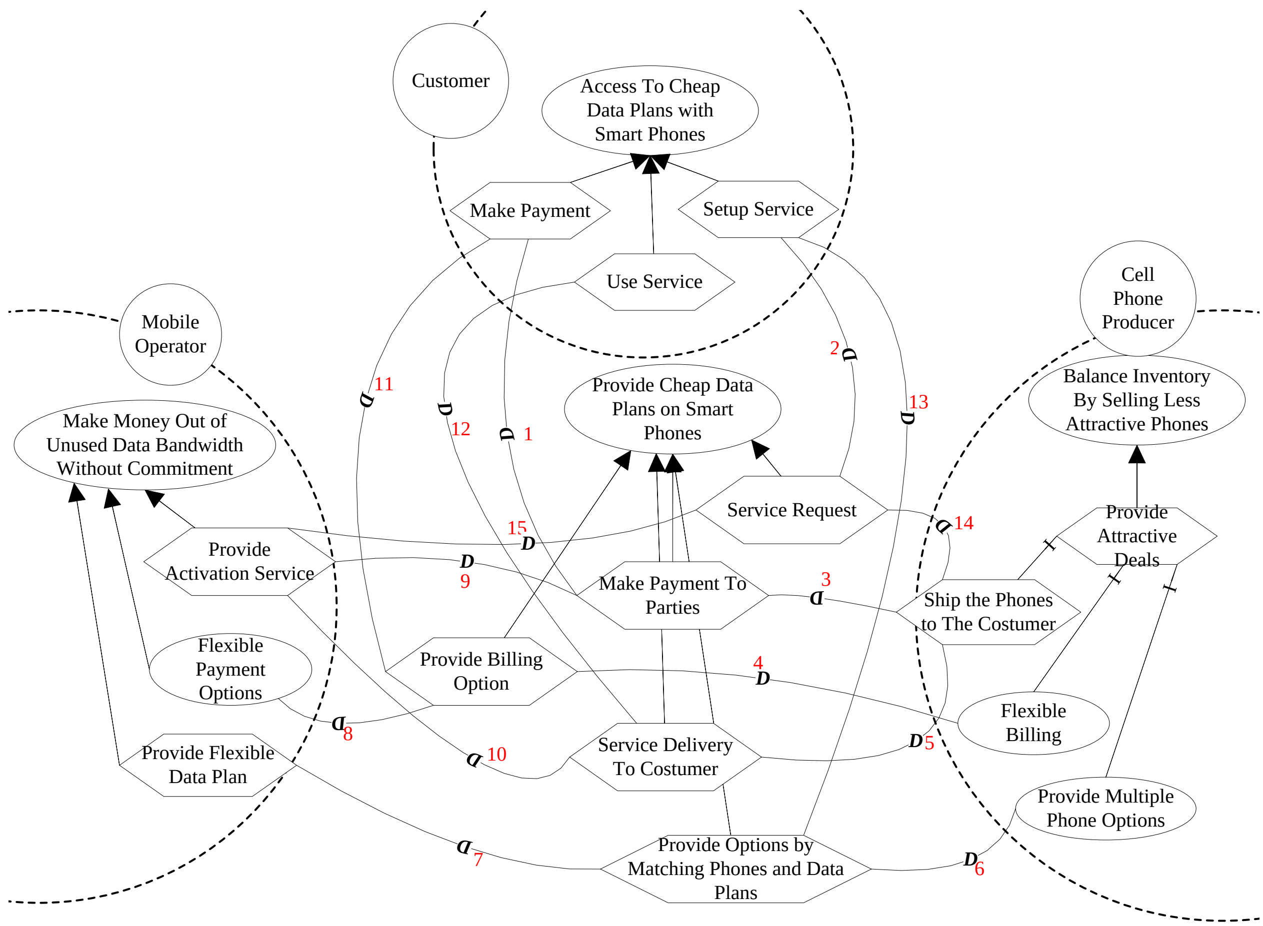

Figure 10: Mobile Operator i* Goal Model

Figure 11 shows the implementation of the model presented in Figure 10. Note that the dependency numbers presented in Figure 10 are the same as the numbers that appear in from of the dependency elements in Figure 11.

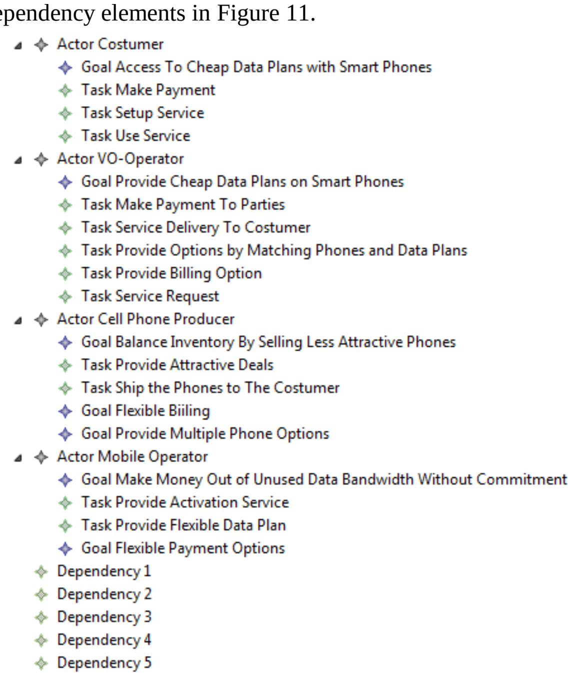

Figure 11: Part of the Mobile Operator i* Model

When executing the operator discussed in section 4.2.1 the e3Value presented in Figure 12 is resulted. Note that the numbers in this figure are added so the reader can trace the values by their ID in future steps of the case study.

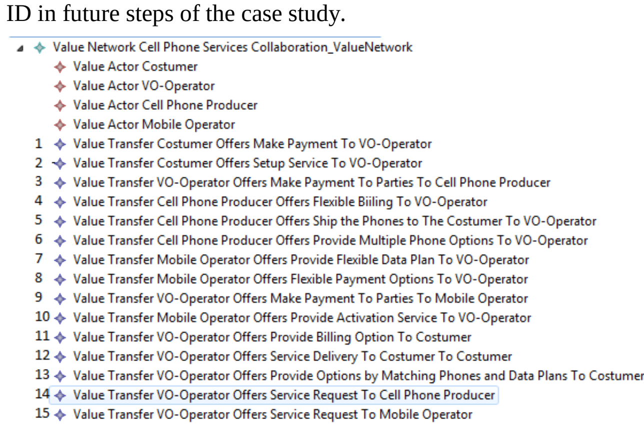

Figure 12: Mobile Operator e3Value Model

At this stage, I execute the operator presented in section 4.2.2. The result is a dependency graph presented in Figure 13. Note that the graphical presentation is added to the image to present a better understanding of the model.

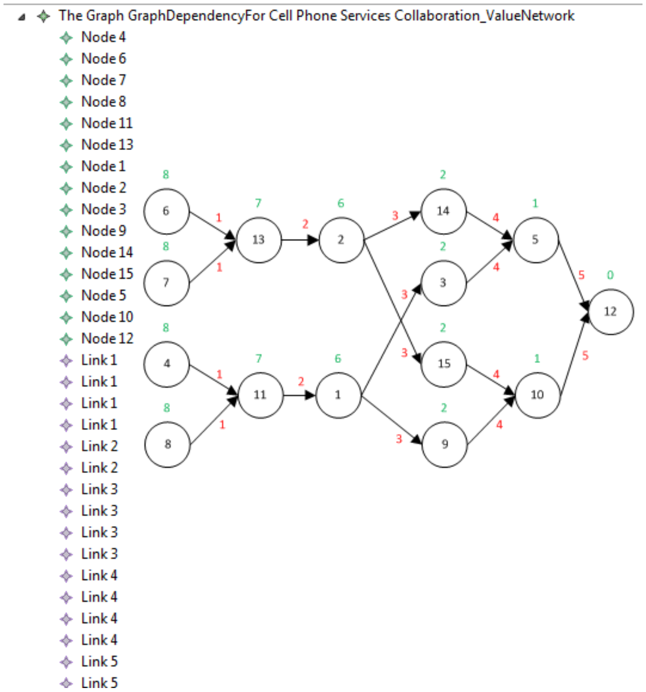

Figure 13: Mobile Operator Dependency Graph Model

This is the final stage of the implementation. As discussed earlier, if value transfers and i* elements are on the process level the dependency graph should represent the overall service choreography with minor addition of context information. Therefore in Figure 14, I present a service choreography that is resulted from this dependency graph. The red numbers in the image refer to the red umbers in Figure 13 that show the modeling steps.

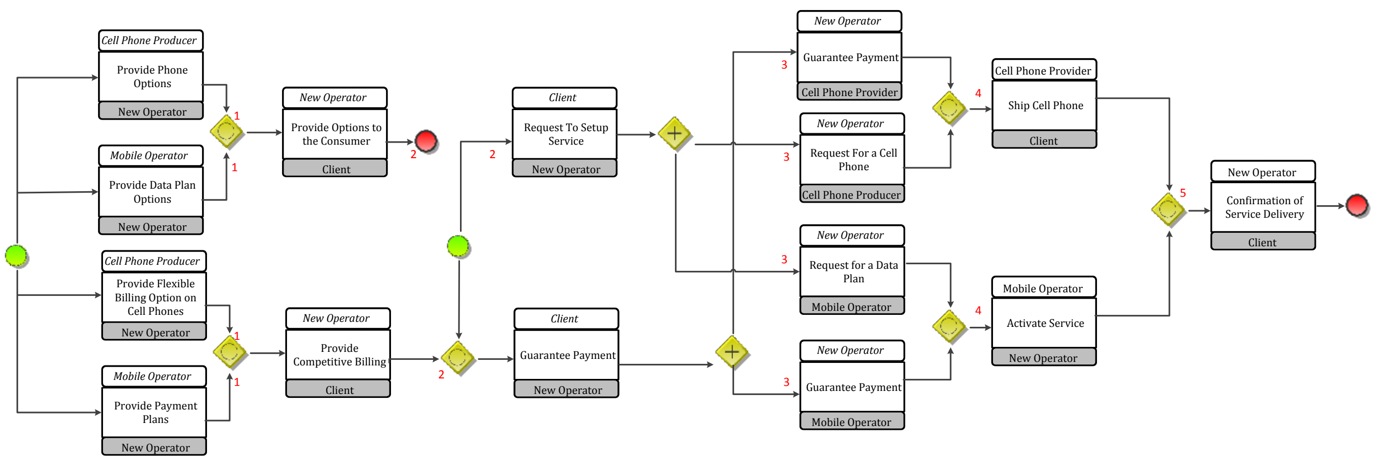

Figure 14: Service Choreography for the mobile Operator Case Study

5.2 Online Radio Station

In this section I am going to discuss a second case study. The e-services for an online radio station case is derived from the case proposed in [14]. It focuses on initiating an international radio service that provides online radio streaming to its users. It has 6 actors involved as it can be seen in Figure 15. Musicians and producers are those who invest in music production and hold the right to certain album or music track. The right society handles distribution of online listeners' contributions to the music producers and musicians. The clearing organization coordinates music publication rights. The internet radio station handles streaming of the content. The listener is the audience listening to the content provider. In Figure 15 there are uses of roles and soft goal which our version of goal models does not support. The tasks performed by the roles in this example are moved to their parent. The soft goals are not considered in the implementation of the example. Note that the soft goals do not have a visible effect on the value network and intent to model the non-functional attributes of the value transfers.

The implemented goal model for this case in this implementation is presented in Figure 16. The goals and levels of tasks in this example are at a high level. The tasks represent duties and overall operations of the actors and not participants processes. Therefore the resulted value network is at the same level of abstraction. As a result the graph dependency can only recommend the overall steps of the interactions and what the participating partners need to agree on and it cannot be used for detail choreography modeling.

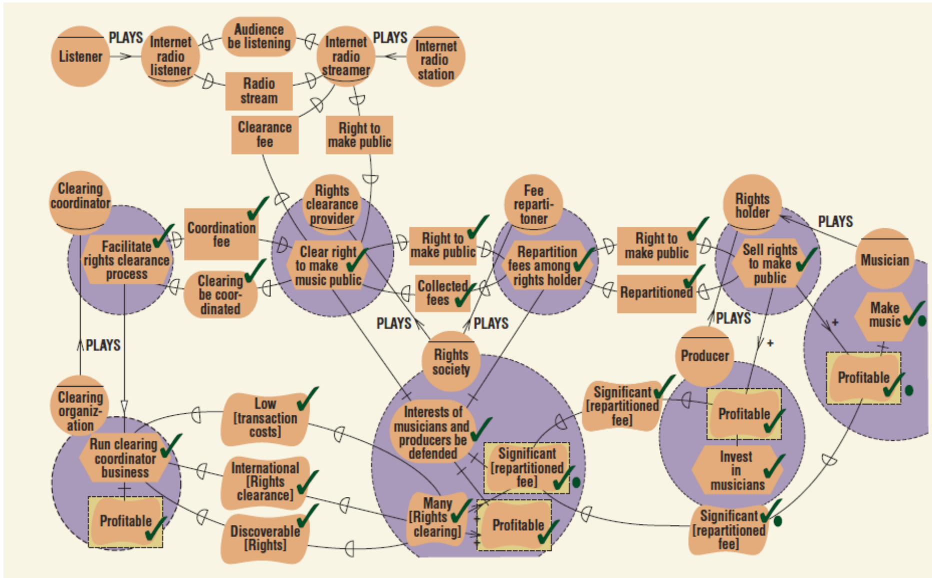

Figure 15: International Radio Service Goal Model [14]

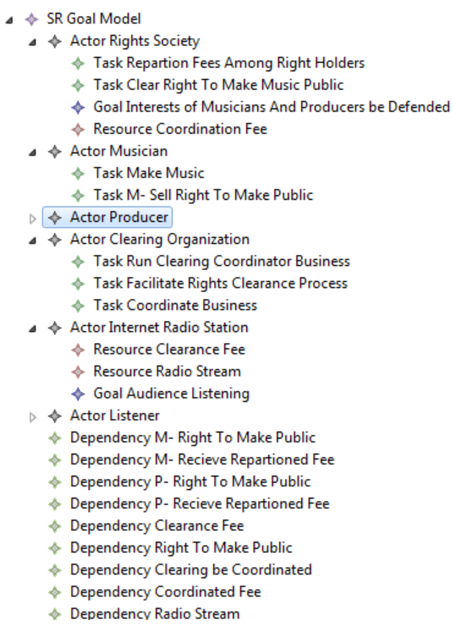

Figure 16: Radio Station i* Goal Model

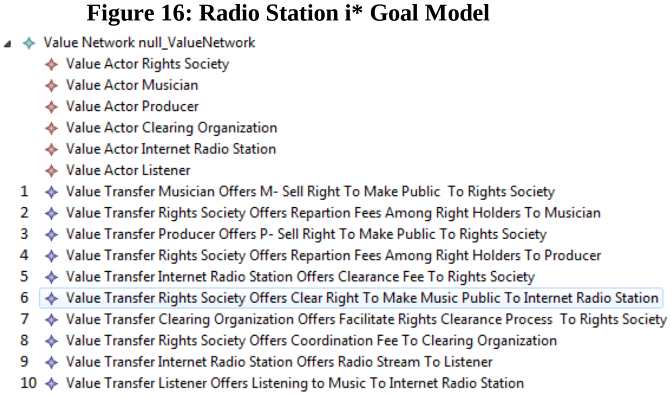

Figure 17: International Radio Station e3Value Model

In Figure 17 the value network resulted from execution of the implemented transformation is shown. The resulted value network represents exactly the same value network presented in [14] by the authors of the case study.

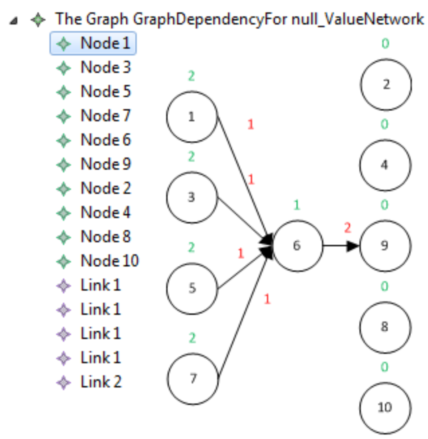

Figure 18: International Radio Station Dependency Graph

Finally the resulted dependency graph is presented in Figure 18. As you can see value numbers 2, 4, 8 and 10 (see what they are in Figure 17) that refer to financial transaction do not require setup negotiation (the analysis of how much money should be transferred is already covered in the feasibility study done in the value network). The dependency graph is showing interactions among partner and their information exchange and the negotiation at this stage focuses on how to synchronize their interactions. Value 1, 3, 5 and 7 refer to negotiation on publication right clearance which needs to be performed in step one. The second step refers to playing music publicly through the online streaming application which the diagram is depicting that must be synchronized with the clearance right process. So the focus of the collaborative task is on providing such process for clearing the rights to make music public.

6. Conclusion

As collaborative organizations have gained more attention, the ability to form and initiate organized collaboration in time to address dynamic opportunities is an asset. In order to realize such goal participating partners need to synchronize their goals and processes while preserving their competitive advantage and autonomy. Therefore forming collaborative organizations such as virtual organization requires analysis and synchronization of strategic alliances, financial gain and process interactions. In this paper I proposed the use of three levels of models to facilitate virtual organization formations. An i* goal model that reflects partners strategic goal and intentions when participating in a collaboration. An e3Value, value network that depicts partners value offering and facilitates feasibility analysis on the overall collaborations. Finally a service choreography that models partners interactions and facilities process synchronization. As part of the modeling approach, two transformations are proposed that illustrate a relation between the three modeling techniques. These models and their transformation are implemented using MMTF. The characteristic of this transformations and its specification were discussed in this paper. These transformation and the resulted relations provide alignment and traceability to the users of this approach and facilitates VO formation and negotiation.

There were a few limitations with the current implementation. In future, the transformation should support the notion of soft goals and make use of different types (committed, open and critical) of dependencies in the i* model. These different levels could be mapped to a dependency matrix that handles weighted dependencies instead of binary values. In future implementations, the dependency inheritance should be included in building the dependency matrix. Finally the transformation could provide more powerful choreography modeling if it can interact with the modeler to facilitate inclusion of context information. The context information will facilitate resolution of loops in value dependencies, identification of starting and ending points of a process and sub processes.

7. References

[1] E. Ermilova and H. Afsarmanesh, "Competency and Profiling Management in Virtual Organization Breeding Environments," in *Network-Centric Collaboration and Supporting Frameworks*, vol. 224, Boston, MA: Springer US, 2006, pp. 131–142.

[2] L. M. Camarinha-Matos, H. Afsarmanesh, N. Galeano, and A. Molina, "Collaborative networked organizations - Concepts and practice in manufacturing enterprises," *Computers & Industrial Engineering*, vol. 57, no. 1, pp. 46–60, Aug. 2009.

[3] M. H. Danesh, B. Raahemi, and M. A. Kamali, "A framework for process management in service oriented virtual organizations," in *2011 7th International Conference on Next Generation Web Services Practices (NWeSP)*, 2011, pp. 12–17.

[4] M. H. Danesh, B. Raahemi, S. M. A. Kamali, and G. Richards, "A Distributed Service Oriented Infrastructure for Business Process Management in Virtual Organizations," presented at the IEEE 25th Canadian Conference on Electrical and Computer Engineering, Montreal, Quebec, 2012.

[5] S. M. A. Kamali, G. Richards, M. H. Danesh, and B. Raahemi, "A framework for performance measurement in service oriented virtual organizations: A value network approach to collaborative performance measurement," in *7th International Conference on e-Business, ICE-B 2012 July*,2012, pp. 263–271.

[6] E. S. Yu, "Social Modeling and i*," in *Conceptual Modeling: Foundations and Applications*, A. T. Borgida, V. K. Chaudhri, P. Giorgini, and E. S. Yu, Eds. Springer Berlin Heidelberg, 2009, pp. 99–121.

[7] Zhongjie Wang, Dianhui Chu, and Xiaofei Xu, "Value Network Based Service Choreography Design and Evolution," in *2010 IEEE 7th International Conference on e-Business Engineering (ICEBE)*, 2010, pp. 495–500.

[8] I. Ul Haq and E. Schikuta, "Aggregation patterns of service level agreements," in *Proceedings of the 8th International Conference on Frontiers of Information Technology*, New York, NY, USA, 2010, pp. 40:1–40:6.

[9] C. Kort and J. Gordijn, "Modeling Strategic Partnerships Using the E3value Ontology: A Field Study in the Banking Industry," *Handbook of ontologies for business interaction*, 2008.

[10] T. Allweyer, *BPMN 2.0*. BoD, 2010.

[11] B. Andersson, P. Johannesson, and J. Zdravkovic, "Aligning goals and services through goal and business modelling," *Information Systems and E-Business Management*, vol. 7, no. 2, pp. 143–169, 2009.

[12] C. Huemer, P. Liegl, R. Schuster, H. Werthner, and M. Zapletal, "Inter-organizational systems: From business values over business processes to deployment," in *2nd IEEE International Conference on Digital Ecosystems and Technologies, 2008. DEST 2008*, 2008, pp. 294–299.

[13] M. H. Danesh, B. Raahemi, S. M. A. Kamali, and G. Richards, "A Framework for Process and Performance Management in Service Oriented Virtual Organizations," *International Journal of Computer Information Systems and Industrial Management Applications.*, vol. 5, pp. 203–215, 2012.

[14] J. Gordijn, E. Yu, and B. van der Raadt, "E-service design using i* and e3value modeling," *IEEE Software*, vol. 23, no. 3, pp. 26 –33, Jun. 2006.

[15] E. Yu, "MODELLING STRATEGIC RELATIONSHIPS FOR PROCESS REENGINEERING," University of Toronto, Toronto, Canada, 1995.

[16] B. van der Raadt, J. Gordijn, and E. Yu, "Exploring Web services from a business value perspective," in *13th IEEE International Conference on Requirements Engineering, 2005. Proceedings*, 2005, pp. 53 – 62.

[17] M. H. Ahmadi Danesh Ashtiani, "Design and Implementation of a Framework for Process Management in Service Oriented Virtual Organizations Using Service Zones," Unversity of Ottawa, 2012.

[18] R. Salay, M. Chechik, S. Easterbrook, Z. Diskin, P. McCormick, S. Nejati, M. Sabetzadeh, and P. Viriyakattiyaporn, "An Eclipse-based tool framework for software model management," in *Proceedings of the 2007 OOPSLA workshop on eclipse technology eXchange*, New York, NY, USA, 2007, pp. 55–59.

[19] K. Czarnecki and S. Helsen, "Feature-based survey of model transformation approaches," *IBM Systems Journal*, vol. 45, no. 3, pp. 621 –645, 2006.