Contract Enactment in Virtual Organizations: A Commitment-Based Approach*

Yathiraj B. Udupi and **Munindar P. Singh**
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
{ybudupi, singh}@ncsu.edu

Abstract

A virtual organization (VO) is a dynamic collection of entities (individuals, enterprises, and information resources) collaborating on some computational activity. VOs are an emerging means to model, enact, and manage large-scale computations. VOs consist of autonomous, heterogeneous members, often dynamic exhibiting complex behaviors. Thus, VOs are best modeled via multiagent systems. An agent can be an individual such as a person, business partner, or a resource. An agent may also be a VO. A VO is an agent that comprises other agents.

Contracts provide a natural arms-length abstraction for modeling interaction among autonomous and heterogeneous agents. The interplay of contracts and VOs is the subject of this paper. The core of this paper is an approach to formalize VOs and contracts based on commitments.

Our main contributions are (1) a formalization of VOs, (2) a discussion of certain key properties of VOs, and (3) an identification of a variety of VO structures and an analysis of how they support contract enactment. We evaluate our approach with an analysis of several scenarios involving the handling of exceptions and conflicts in contracts.

Introduction

Virtual organizations (VOs) are dynamic collaborative collections of individuals, enterprises, and information resources (Foster, Kesselman, & Tuecke 2001). Traditionally such collaborative activities are focused on data sharing and computation. This paper emphasizes VOs in business settings, especially where processes support delivery of real-world (not just IT) services. Production grids employed for scientific or business computing are excellent examples of settings where this approach can be applied. Because of legal and economic pressures, business environments provide richer policies than the more common scientific computing environments. VOs, whether business or scientific, have key properties that distinguish them from traditional IT architectures:

Autonomy. The members of a VO behave independently, constrained only by their contracts.

Heterogeneity. The members of a VO are independently designed and constructed, constrained only by the applicable interface descriptions.

Dynamism. The configuration of a VO changes at runtime as members join and leave.

Structure. VOs have complex internal structures, reflected in the relationships among their members.

The above properties of VOs closely match the properties of multiagent systems. Agents are persistent computations representing independent principals: they are *autonomous* and *heterogeneous* as a result. Multiagent systems are motivated from flexible human organizations and consequently exhibit *dynamism* and *structure*. Thus the distinguishing properties of VOs are mirrored in multiagent systems.

Collaborations among agents are structured via contracts. A contract is modeled as a set of commitments. A VO is formed between the contracting agents if it does not exist already. VOs can have complex nested structures and hence contracts may be formed at multiple levels. More than one contract may simultaneously exist among a set of contracting agents. Here, the VOs within which the contracts are formed may overlap resulting in situations where an agent belongs to two or more VOs, neither of which is an ancestor of the other. Several other factors come into play while creating contracts in addition to the consequences of the VO structures. VOs have key properties that are essential for handling contracts, commitments, and the various operations on commitments. This paper identifies several different VO structures and their implications on contract enactment and vice versa.

A VO not only encapsulates some relationships among its members, but also functions as an agent that engages in potentially complex relationships with its members. In our approach, these relationships are expressed in terms of goals, policies, and commitments. For example, the goals of a VO can be propagated to its members as goals, or may become the commitments of its members. Likewise the policies of a VO would normally be propagated to its members. The policies of a VO might control how the commitments among its members evolve. Consequently, as an important example, if two agents enter into a contract, besides the commitments that are explicitly part of the contract, their behavior would be constrained by the goals, policies, and commitments of

*This research was supported by the National Science Foundation under grant ITR-0081742.

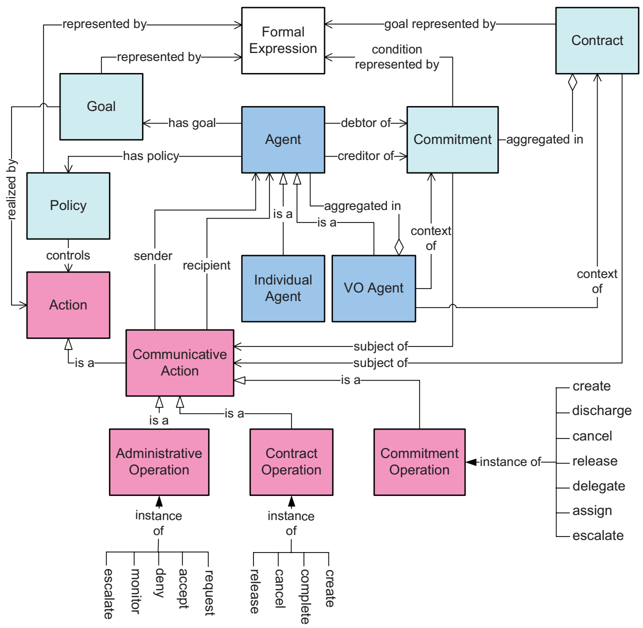

Figure 1: Conceptual model of multiagent VO approach

their ancestor VOs. For instance, the parent VO might declare a contract invalid or completed, or might release one of the agents from its commitments according to the contract. Such flexibility is essential for a VO to handle exceptions and accommodate opportunities.

The main contributions of this paper are (1) a formalization of VOs, (2) a discussion of certain key properties of VOs, and (3) an identification of a variety of VO structures and an analysis of the different ways of contract enactment. We evaluate our approach with an analytical study involving the handling of exceptions and conflicts in contracts.

A Multiagent Approach for VOs

The proposed approach is centered on the notion of *agents*. An agent is a computational entity with a persistent identity that is proactive and interactive. As a base case, an agent may be an individual, such as a person, business partner, or a resource. An agent may also be a VO. That is, a VO is an agent that comprises other agents, including, other VOs. A member of a VO is referred to as its *child*, so that kinship terminology such as parent, descendant, ancestor, can be used. The key features of our approach are that it: (1) models and handles recursively formulated VOs, and (2) allows for agents supporting each VO's autonomy and local policy compliance.

Agents (individual or VO) collaborate by sharing goals, forming contracts, and creating commitments among one another. They apply various operations on these commitments and contracts. The agents' interactions are best understood as *communicative acts*, which involve their contracts or their constituent commitments. Figure 1 presents our conceptual model. The following describe the concepts relevant to this paper in detail.

Commitments are the central primitive for expressing organizational interactions among agents (Singh 1999). A commitment functions like a directed obligation from its *debtor* to its *creditor*: it specifies the condition that the debtor is obliged to the creditor to bring about. Importantly, unlike traditional obligations, commitments are defined within an organizational *context*. Conditional commitments associate a condition with a precondition. Conditional commitments provide a natural basis for contracts.

Definition 1 A commitment is defined as $C(A, B, F, U, \mathit{Id})$, where A is the debtor, B the creditor, U the context, F the condition, and Id the Id of the commitment.

In essence, commitments reify aspects of agent interactions and enable interactions to be treated as first-class citizens in our representations. Six operations are defined on commitments (Singh 1999). Commitments are *created* among agents in the context of a common VO to accomplish certain goals. The debtor of a commitment *discharges* it by bringing about the stated condition. A debtor can *cancel* a commitment. The context VO or the creditor of a commitment can *release* it relieving the debtor. A debtor of a commitment may *delegate* it to another agent: the outcome of the delegation is that the delegatee becomes the debtor of a commitment with the same condition and creditor as the original commitment. Similarly a commitment can be *assigned* to another creditor within the same context VO. These operations appear in the policies of an agent.

We introduce an operation *escalate*. Intuitively, *escalate* conveys a meaning that is the reverse of *delegate*. An agent may *escalate* a commitment to a superior if it is not able to discharge it or if it is the creditor and the debtor has failed to discharge the commitment.

The motivation for the context of a commitment is to delimit the scope of a commitment, so as to enable the proper treatment of exceptions and opportunities. In particular, commitments in real life are not irrevocable. Often, an agent has no choice but to revoke a commitment because of problems that may be, for instance, physical (factory burned down), economic (oil prices shot up unexpectedly), or legal (cannot ship pharmaceuticals across national boundaries). The context of a commitment provides a way to revoke or otherwise manipulate commitments.

Goals capture the states of the world that the agent desires to bring about. Goals are ends, but function as means to other goals. In connection with VOs, the goals (ends) of some agents may cause them to enter into a contract or form a VO. Conversely, the contracts that a VO enters into may cause it to adopt goals (means), which could potentially yield additional goals for its children and other descendants. Goals are (potentially) realized by agent actions.

Contracts comprise commitments involving two or more agents. Typically, each agent participating in a contract would be the creditor of some commitments and the debtor of some commitments in the contract. Also, typically, the commitments would be conditional and may refer to the conditions of other commitments in the contract.

Contracting agents become children of the same VO. To simplify our model, we require that each contract corresponds to a (new) VO created when that contract is established. As noted above, each commitment exists within the scope of a context VO. This leads to a coherence requirement for contracts: *Each of the commitments in the set that constitutes a single contract must have the same VO as their context.*

Definition 2 A contract is a tuple $\langle M', G', S', U \rangle$, where $M' = \{A_1, \ldots A_n\}$ is a set of agents, G' is a set of goals (each a goal of some A_i), and S' is a set of commitments. Each commitment in S' involves a debtor and a creditor drawn from $M' \cup \{U\}$. All the commitments in S' have the same context U which maps to a VO.

The following operations are defined on contracts. A contract is *created* when a set of agents collaborate to accomplish certain goals. Thus typically when a contract is established the conditions of each commitment in it would correspond to a goal of the commitment's creditor. A contract is *completed* by the context VO if all the commitments specified in the contract are successfully executed. A contract can be *released* by the context VO, which essentially eliminates all the commitments for that contract. A contract is *canceled* by an agent to cancel the constituent commitments: typically, if it cannot discharge its commitments without violating its context VO's policies.

Interesting scenarios arise when an agent is responsible for two contracts simultaneously. The commitments from one contract may need to be released in order for the agent to be involved in a second contract. But the release can be guided by the context VO within which the contract is enacted. The structure of a VO and the different contractual requirements on its descendants has interesting ramifications on how contracts are enacted.

Commitment operations extended. In VO settings, commitment operations, especially delegation, can differ in some respects from the generic operations on commitments. The actions and interactions required for a contract may be carried out by the descendants of the contracting VOs. A contracting VO would delegate its commitments to its children. The outcome of a delegation (or assignment) is that the delegatee (or assignee) takes the place of the delegater (or assigner). However, we leave a "trace" of the commitment with the delegater (or assigner) to handle any exceptions, e.g., by monitoring its enactment and responding to any incoming escalates from the delegatee (or assignee).

Communicative acts correspond to operations on contracts and commitments. We identify three kinds of communicative acts: *administrative* (not emphasized here), *commitment-based*, and *contractual*. A communicative act is expressed using the language L_χ described below:

Definition 3 $L_\chi = \{\mathit{operationName}\,(A, B, \Theta)\}$ A is the source of the communicative act and B its destination. Θ is the content of the communicative act.

Above, Θ may be the commitment or the contract on which the operation is made. For example, a commitment $C(A_1, B_1, F_1, U_1, i_1)$ can be created by A_1 and communicated to B_1 using the communicative act of $\mathit{create}(A_1, B_1, C(A_1, B_1, F_1, U_1, i_1))$.

Policies are rules to control operations on commitments, contracts, and organizational relationships in which an agent is involved. Each agent has a policy engine that responds to incoming communicative acts. An agent would attempt to enforce its policies (including those received from ancestor VOs) by monitoring for any exceptions or conflicts among its commitments.

A Contractual VO Scenario

In business settings, delegation is routine. For example, a university (say, Univ) contracts with the IDA Agency to have the Engineering building trimming painted. The contract specifies the stated service in the form of goals and commitments from both the contracting parties. IDA creates a commitment with Univ as its creditor to discharge the stated service. While Univ creates a commitment with IDA as its creditor to ensure the goals of proper scheduling, facilitating the paint job, and timely payment for the service. Univ delegates the tasks of scheduling, facilitating, and judging the paint job to its Facilities department. IDA delegates the job to its city division (IDA-City) where the university is located, which would deal with Univ Facilities.

Each member acts in accordance with its policies. Because of delegation, children adopt the contractual restrictions determined by parent organizations, which might potentially cause some local policies to be overridden. For example, facilities may allow a building to be painted only on student holidays and IDA-City may only paint the outside if the ambient temperature is below 80° Fahrenheit.

Similarly, the creditor of a commitment may assign it to a descendant. In the example above, consider the commitment of painting involving IDA as the debtor and Univ as the creditor. Here, Univ being the creditor can assign the commitment to its Engineering department. If IDA realizes that it cannot satisfy its commitment, it can cancel its commitment. If the Engineering department is not happy with the paint job done by IDA-City, it can escalate the commitments to Univ. And, if Univ decides not to do business with IDA for the poor quality of its job, it can release IDA from its commitment (and hence the contract). However, the commitments are discharged only upon successful completion of the specified tasks.

Virtual Organizations Formalized

Based on the foregoing motivations, we formalize a VO as follows. Recall that a VO is an agent; an agent must be a VO or an individual.

Definition 4 A VO A is an agent defined via a tuple $\langle M, G, P, S \rangle$. Here $M = \{A_1, \ldots, A_n\}$ is a set of agents. G, P, and S, are sets of formulas. The A_i are the children, G are the goals, and P are the policies of A. S is a set of commitments, each of which has a creditor and a debtor drawn from $M \cup \{A\}$ and a context equal to A.

As discussed earlier in connection with contracts, contract formation results in the formation of a VO that includes all the contracting parties. The following clause describes a constraint on the contract formation.

Clause 1 $\textit{contract}(M', G', S', U) \Rightarrow (\forall A_i : A_i \in M' \Rightarrow A_i \in U.M) \wedge (G' \subseteq U.G) \wedge (S' \subseteq U.S)$

A contract among a set of entities ($M' = \{A_1, \ldots, A_n\}$) with a parent VO U as its context implies that all A_i are children of U, and the goals and commitments of the contract are part of U's goals and commitments, respectively.

Let us introduce the notation used in Figure 2 with reference to Part (a). The root node is the VO X, with child VOs A and B. Double edges, such as the one between A and B indicate a contract (of two or more parties). Single edges relate a parent (upper) and child (lower). For example, the edge from A to A_1 indicates that A_1 is a child of A. The leaves are individual agents. Nonleaf nodes are VOs.

In Figure 2(a), a contract C_1 is formed between A and B in VO X, which serves as their parent. In this scenario, A accomplishes its part of the contract by delegating its commitments to a child A_i. Each such A_i should discharge the commitments delegated to it with the goal of B being satisfied. In case of conflicts with A_i's previous commitments, A can delegate a commitment to A_i if A can successfully release A_i from all previous commitments that have A as their context, and which could have arisen because of other contracts (for example, C_2 between A_1 and A_2 in Figure 2(a)).

Duality of Contracts and Virtual Organizations

It is important to understand the relationship between contracts and VOs. These concepts are closely related but not identical. They are best understood as duals of each other.

A contract is a static entity capturing relationships among two or more agents. The context VO for a contract, which is created when the contract is instantiated, represents just the contract. Typically the contract (and its context VO V) would arise within a VO V' that already exists and where the contracting agents are peers. There would presumably have been another contract that led to the formation of V', which in turn would have been created within another existing VO, V'', and so on. The process can terminate at any "social" organization or institution that is not modeled computationally, e.g., the state of North Carolina, the European Union, or a professional body such as AAAI.

A VO is a dynamic entity, whose membership or structure might evolve, and within which commitments and contracts are manipulated via communicative acts. A VO would have been created through a contract (or simply postulated as a starting point). Simply put, on the one hand, a VO is created through a contract and, on the other hand, provides a basis for creating, manipulating, and enacting further contracts.

In the Univ-IDA example, agents Univ and IDA both happen to be VOs formed (in prehistory) by structuring their members via commitments, contracts, and policies. When Univ and IDA enter into a contract, a new VO is created for this contract. Based on their structures, Univ and IDA can delegate and assign commitments from this contract as appropriate. The contracts that might have formed Univ and IDA are not being manipulated, but provide a basis for enacting the new contract between Univ and IDA.

Key Properties of a Virtual Organization

The following are some important properties generally (but not always) supported by VOs in our framework. The next section presents cases that illustrate these properties.

Pr 1 An agent is restricted to communicate with its children, parents, and siblings.

- $\textit{operationName}(A, B, \_) \Rightarrow (B \in A.M) \vee (A \in B.M) \vee (\exists Z A, B \in Z.M)$

This property supports distributed management of VOs by simplifying the responsibility of a VO to check for compliance and handle exceptions in contracts among its children. The children must collaborate with their parents for enforcing the policies of their parents.

Pr 2 An agent exhibits dynamic behavior and may engage in multiple contracts simultaneously. However, for each contract entered by any agent, there is a different VO.

- $\textit{contract}(M'_x, \_, \_, U_x) \wedge \textit{contract}(M'_y, \_, \_, U_y) \wedge (M'_x \bigcap M'_y \neq \phi) \Rightarrow (U_x \neq U_y)$

This property restricts a VO from having two different contracts involving a common child. An agent may autonomously decide the contracts in which to participate.

Pr 3 A VO can delegate or assign its commitments (e.g., those resulting from a contract) to its children.

- $\textit{delegate}(A, A_i, \_) \Rightarrow A_i \in A.M$
- $\textit{assign}(A, A_i, \_) \Rightarrow A_i \in A.M$

This property enables distributed enactment of contracts.

Pr 4 A VO can release the commitments that are formed either by a contract entered into by its children, or by delegation.

- $\textit{release}(A, A_x, C(A_x, A_y, \_, A, \_)) \Rightarrow A_x, A_y \in A.M$

This property says that a context VO may exercise the release operation only over its children's contracts and associated commitments. For example, in Figure 2(b), Y can release B only from the commitments of the contract C_4, and not of contract C_3 because both B and D are children of Y, whereas A is not a child of Y.

Pr 5 An agent can escalate a complaint regarding a commitment or a contract to its parent VO.

- $\textit{escalate}(A_x, A, Q(\ldots)) \Rightarrow A_x \in A.M$

This property restricts escalates to propagate only up in the VO structure from a child to a parent.

Formal Enactment of Contracts in a VO

We evaluate our approach via a study of several VO schemas involving the handling of exceptions and conflicts in contracts. We identify some important cases based on the structure of the VOs and the contracts formed at different levels. These cases mostly satisfy the properties enumerated above. We study these cases with the help of Figure 2.

Exception handling in contracts. An exception occurs when one or more of the parties involved do not behave according to the contract terms. Conflicting contracts cause

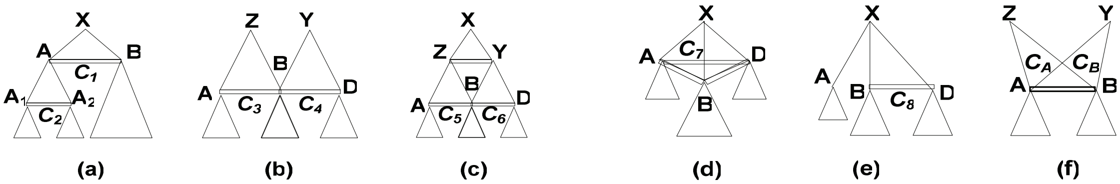

Figure 2: Contract creation and enactment scenarios in important VO schemas

exceptions. Exceptions are caught by the parent VOs of the children involved in the responsible contracts when one or more of the contracting children send an escalate communicative act to the parent VO. For example, in Figure 2(b), if an exception is created in contract C_3 between A and B, the parent VO Z handles this exception either by forcing the responsible parties A and B to behave appropriately (i.e., by enforcing its policies), or by replacing the problematic agent, in effect releasing it from the contract and delegating the commitment to another agent.

Case 1: In Figure 2(a), a contract C_1 exists between A and B. Here, A can delegate its commitments from the contract to child A_1. If A_1 were previously committed to another child of A, say A_2, then A can release A_1 from that commitment. *Applicable properties:* Pr1, Pr3, and Pr4.

Case 2: Figure 2(b) shows a situation of overlapping VOs with one VO having a contract C_3 between A and B, and another VO having a contract C_4 between B and D. Under these circumstances, A is not within the context of the $B \Leftrightarrow D$ contract that exists in a different VO. B may be involved in both contracts if they do not conflict or decide which to enact. *Applicable properties:* Pr1 and Pr2. Pr4 and Pr5 become applicable if A requests B to enact contract C_3, and B in the case of a conflict, escalates a commitment to its VO Y, who can release B from contract C_4.

Case 3: Figure 2(c) modifies Figure 2(b) to include a common ancestor VO X of A, B, and D. Here, A is not in the context of the $B \Leftrightarrow D$ contract C_6, but A can escalate a request via its parent Z to X. B and D are in the scope of X through their parent Y. X can tell Y to release B from contract C_6. *Applicable properties:* Pr1, Pr2, Pr4, and Pr5.

Case 4: In Figure 2(d), a three-party contract C_7 is formed between A, B, and D. Here, in the case of any conflicts, one of A, B, or D can escalate a request to the parent X who is the context VO of C_7. *Applicable Properties:* Pr1, Pr4, and Pr5.

Case 5: In Figure 2(e), a contract C_8 exists between B and D, with A being a child of the same VO as B and D. Here, A cannot influence B or D in the $B \Leftrightarrow D$ contract, but A can escalate a request to its parent X if it wants to contract with B or D, and X can release B or D from the contract. This is different from Case 2, but is like Case 3, because here A, B and D all have a common parent X. *Applicable Properties:* Pr1, Pr4, and Pr5.

Case 6: Figure 2(f) illustrates a scenario where A and B participate in two contracts with each other. This results in the formation of VOs Z and Y for the contracts C_A and C_B, respectively. Under these conditions, conflicts may occur if A or B cannot simultaneously discharge the commitments formed within the two contracts. If another VO, say, W includes the VOs Z and Y, then W can determine which contract is given precedence. This would match Case 3. Otherwise, A and B must negotiate which of the conflicting contract to pursue. *Applicable Properties:* Pr1, Pr2, Pr4, and Pr5.

Literature

We compare our approach with some existing approaches dealing with organizations.

Communities are based on members having similar objectives and resources (to be shared). Feeney *et al.*'s architecture supports delegation of authority across communities to manage distributed resources (Feeney, Lewis, & Wade 2004). They support a nested community architecture with hierarchical policy enforcement, especially with respect to conflict resolution. New policies are checked for conflicts and recursively propagated to parent communities until they reach the community owning the target resource, where they are deployed. Policy decisions of a community are made via consensus among its members. Our approach can model a community as a VO whose policies reflect the consensus of its children. The VO agent provides a locus for compliance with and enforcement of the community policies as well as for the conflict resolution among community members.

Normative systems specify norms for their members and can be modeled using agent-based VO architectures. Norms constrain an agent's behavior. Boella *et al.* propose a conceptual model of VOs as normative multiagent systems (Boella, Hulstijn, & van der Torre 2005). Here, they recursively model normative systems and demonstrate the dynamic aspects of VOs using different types of interactions between the normative systems and the agents who have assigned roles. Dastani *et al.* show how organizational roles and norms influence an agent's goals (Dastani, Dignum, & Dignum 2002). Our VO architecture can be applied to a normative system where the policies can specify the norms of a society which eventually form the local policies of an agent. Our approach provides a commitment-based reasoning to the interactions between agents and VOs.

Contracts in organizations can be understood as systems of legal rules that change the regulative and constitutive norms of an organization (Boella & van der Torre 2004). Boella *et al.* formalize contracts as having constitutive rules that demonstrate how the creation of a contract relates to the mental attitudes (beliefs, desires, and goals) of an organization. Our approach formalizes contracts in terms of commitments, and demonstrates the enactment of the contracts based on operations on commitments. The commitments in our approach are similar in some respects to Boella *et al.*'s notions of obligations. However, our approach relates contracts to commitment operations, and thus yields a more precise model of the enactment of contracts.

Organizational designs are patterns of organizing multiagent system with a view to classifying their performance characteristics (Horling & Lesser 2005). Horling *et al.* present a distributed algorithm that uses an underlying organization to guide coalition formation. Brooks and Durfee (2003) demonstrate how "congregations" of agents can benefit multiagent systems in searching other agents and minimizing the interactions and search costs.

Teams are formed when a number of cooperative agents get together to accomplish a common goal (Tambe 2003). Agents coordinate their actions in a way that is consistent and supportive of their team's goal. Our approach considers commitment-based contracts that bring together agents to form VOs. Here, the contracting agents collaborate in the context of a common VO to accomplish the contract goals. Our approach can be thought of as addressing the same basic problem, robust teamwork, but specialized to VOs where contracts capture the essence of the interactions and the commitment operations support responses to various exception conditions.

Social reasoning mechanisms and relationships enable an agent to evaluate and reason about others using its dependencies with others (Sichman & Conte 2002). Several relationships can exist among the agents and these relationships can influence the actions taken by them. This idea can be enhanced and combined with our approach as follows. In the VO context, relationships can exist between a VO and its children, as well as among the siblings. Relationships here are dynamic, because they can be formed and revoked at run time. Such relationships can form the basis of policies, and can influence the operations on commitments and contracts.

Conclusion

This paper proposes a commitment-based architecture for VOs. This architecture treats VOs as consisting of agents, potentially VOs in their own right. The nesting structure of the VOs highlights the freedoms and constraints on the VOs at each level. Each VO is associated with a set of goals, commitments, and policies. The key advantages of this architecture are as follows.

Relationships. The proposed architecture naturally supports complex structures. It enables VOs that are nested or partially overlapping. It supports managing the complementary properties of two VOs being unaware of each other's structure but gaining requisite visibility to interact effectively.

Policy Management. The proposed architecture recognizes that VOs are distributed. It supports two complementary perspectives. One is that there is a single locus of policy enforcement. The other is that a distributed organization must have parts that collaborate to enforce a given policy.

Contract Enactment. The proposed architecture provides a commitment-based argument for contract enactment. Our hierarchical and distributed VO setup facilitates handling of various scenarios of conflicts and exceptions in contract enactment.

The effect of relationships between agents on contract enactment in a VO and other enhancements to our formal VO definitions will be considered as near future work. The relationships captured while describing VOs can dynamically change and becomes crucial for describing VO behaviors.

References

Boella, G., and van der Torre, L. 2004. Contracts as legal institutions in organizations of autonomous agents. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, 948–955.

Boella, G.; Hulstijn, J.; and van der Torre, L. 2005. Virtual organizations as normative multiagent systems. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 192–201.

Brooks, C. H., and Durfee, E. H. 2003. Congregation Formation in Multiagent Systems. *Autonomous Agents and Multi-Agent Systems* 7(1-2):145–170.

Dastani, M.; Dignum, V.; and Dignum, F. 2002. Organizations and normative agents. In *Proceedings of the First EurAsian Conference on Information and Communication Technology*, 982–989. Springer-Verlag.

Feeney, K. C.; Lewis, D.; and Wade, V. P. 2004. Policy based management for Internet communities. In *Proceedings of 5th International IEEE Workshop on Policies for Distributed Systems and Network (POLICY)*, 23–32.

Foster, I.; Kesselman, C.; and Tuecke, S. 2001. The anatomy of the grid: Enabling scalable virtual organizations. *The International Journal of High Performance Computing Applications* 15(3):200–222.

Horling, B., and Lesser, V. 2005. A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review* 19(4):281–316.

Sichman, J. S., and Conte, R. 2002. Multi-agent dependence by dependence graphs. In *Proceedings of 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, 483–490.

Singh, M. P. 1999. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law* 7:97–113.

Tambe, M. 2003. Towards flexible teamwork. *Journal of Artificial Intelligence Research* 7:83–124.