Applying Semantic Web Technologies that Automate Extended REA Typifications, Groupings and Associations

Tod Sedbrook, Computer Information Systems, Monfort College of Business Administration, University of Northern Colorado, Greeley, CO 80639

Richard Newmark, Accounting, Monfort College of Business Administration, University of Northern Colorado, Greeley, CO 80639

Charmayne Cullom, Computer Information Systems, Monfort College of Business Administration, University of Northern Colorado, Greeley, CO 80639

Corresponding author: Tod Sedbrook

Email: tod.sedbrook@unco.edu

Phone: 303:320-4515

Applying Semantic Web Technologies that Automate Extended REA Typifications, Groupings and Associations

Abstract

Enterprise modelers require tools and techniques that represent comprehensive domain knowledge in a formal and logically consistent manner. Current modeling approaches rely on entity relationship or unified modeling diagrams to represent semantic descriptions of business exchanges, but it remains difficult to transform the implicit metadata, ontologies and logic embedded in diagrams into a coherent and concise form that can be interpreted by machines. This study explores the effectives of uniting the machine processing capabilities of semantic web technologies with REA enterprise ontologies to model complex enterprises. We apply Web Ontology Language (OWL) and Semantic Web Rule Language (SWRL) to model REA policies for a distributed e-commerce partnership selling nearly new vehicles. We participated in a field study to assess the effectiveness of semantic technologies for sharing, improving and applying the extended REA model. OWL and SWRL provide an explicit machine executable representation that allows enterprise designers to formally express policies, experiment with updates and quickly incorporate changes as guided by the extended REA ontology.

Introduction

Enterprise designers continually adapt business policies—such as partner responsibilities, product promotions and supply chain flows—to cope within turbulent environments. Coherent and concise change models are needed to help enterprises continually adapt software systems to policy changes. Current modeling approaches rely on manual updates to entity relationship (ER) or unified modeling language (UML) diagrams to capture changing business policies and communicate changes to system developers. Updating diagrams and other design documents requires careful review by knowledgeable domain experts that understand the implicit metadata, semantics and logic embedded within diagrams. It is increasingly difficult and costly to coordinate model updates as distributed networks of enterprises adjust their policies to cope with continual change.

This study explores the effectives of joining the automated processing capabilities of semantic web technologies with McCarthy's (2006) extended Resource, Event, Agent (REA) ontology to better capture ongoing changes inherent in distributed enterprise collaborations. We report results of a field study that applies semantic tools directed by the REA ontology. The field study observes difficulties with standard modeling approaches and identifies benefits resulting from the application of semantic web technologies that incorporate the extended REA ontology.

The REA ontology defines a semantic vocabulary of economic Resources, Events, and Agents to define business exchanges. McCarthy addressed the semantic weaknesses of traditional accounting systems by creating an events accounting system based on the entity-relationship conceptual model (McCarthy 1982). REA applies economic theory to structure the economic events, exchange relationships and inflows and outflows of resources.

Incorporating REA extensions in evolving policy level e-business standards requires evaluating tools and techniques that apply typification, grouping and derived associations to support manager's intentions (Geerts and McCarthy 2006). The policy level refers to the abstractions constraining what should or could take place to satisfy partners participating in economic business exchanges.

Typifications provide a hierarchy of types to allow properties associated with general classifications to be inherited by more specific types. Typifications are used to connect operational-level entities (INVENTORY; individual vehicles) with policy-level generalizations (MODEL; Chevy Nova, Ford Taurus). Typifications model policy-level generalizations of types—e.g., CATEGORY (full-size, midsize, SUV) is a generalization of MODEL. Typifications also represent relationships between commitment events (e.g., SALES ORDER) and resource types (.e.g., MODEL; Chevy Nova, Ford Taurus), event types (SALE TYPE; U.S. sale, E.U. sale) and agent types (e.g., SALESPERSON; experienced salesperson; inexperienced salesperson) (Geerts and McCarthy 2006).

Groupings define bounded sets representing the union or intersection of possibly heterogeneous and dissimilar member types. For example, management may form the group "profitable" where members represent the union of selected customers and products. Derived associations further structure management's policy intentions by organizing and constraining interactions among generalized types and groupings.

Developments in semantic web technology such as Web Ontology Language (OWL) and Semantic Web Rule Language (SWRL) provide a new opportunity to apply extended REA ontologies and create REA-based applications (Dunn and Grabski 2000). We first address the problem of representing REA extensions related to typifications, membership groupings and associations within an OWL and SWRL ontology. We then consider the epistemological adequacy and maintainability of the resulting ontology (Hevner, March et al. 2004). The next section discusses related research. Subsequent sections outline our modeling approach and discuss the field study applications of OWL and SWRL to support inter-enterprise collaboration.

Related Research

Referred to as REA primitives, **R**esources, **E**vents, and **A**gents continue to prove effective in modeling and supporting activities across a wide-range of economic exchanges (Hruby 2006; McCarthy 2006). Associations such as such as stockflows, duality and participation relate primitives to express useful cognitive reference points for mirroring physical exchanges of resources and modeling interactions of employees, suppliers and customers. Yet as REA modeling evolves to support enterprise collaborations across supply chains, researchers recognize the need to extend the REA model to create richer models of economic transactions and to capture inter-enterprise business standards, policies and budgets (Geerts and McCarthy 2002).

REA modeling extensions have been proposed to support management planning and decision making, enterprise level value chains and workflow structures (Geerts and McCarthy 1999; Geerts and McCarthy 2002; Verdaasdonk 2003). Geerts and McCarthy described an extended REA model to provide a superordinate infrastructure that identifies additional ontological categories including typifications, membership groupings, and associations (Geerts and McCarthy 2006).

REA extensions are motivated by Sowa's (2000) ontological analysis that organized conceptual entities into three categories: independent, relative and mediating. The independent category contains REA's resource, event and agent primatives. The relative category relates independent level primitives to form dyadic associations. A sales event triggering a resource exchange is a relative association. Mediating forms provide additional associations that model the purpose or intention of a relative relation.

Mediating associations, for example, are applied to form judgments concerning what "ought to be" and what one "ought to do". For example, consider a situation where inexperienced sales people are helping important customers. A manager remarks that "experienced sales people ought to help important customers." The manager expresses a mediating policy intention that comments on the relative association between a salesperson and a customer. Such a policy could be expressed as an association policy between customer type (e.g., regular customers, preferred customers) and employee type (e.g., inexperienced salespeople, experienced salespeople).

Applying independent, relative and mediating categories to REA results in an explicit separation of operational and policy level concepts (Geerts and McCarthy 2002). Operational level, resources, events and agents relate to basic business concepts that are directly experienced, perceivable and physically recognizable. They are used to record information about events that have already happened. REA policy level extensions are, however, less intuitive since they require additional conceptual effort to classify REA primitives into generalized types hierarchies, membership groupings and policy-level associations. Disagreements continue concerning the semantics and application of these superordinate policy structures (Lampe 2002).

Sowa (2000) suggests logic, ontology and application of computer-based tools play complementary and supporting roles in supporting human understanding. This research maintains that progress in extending the REA ontology cannot be separated from its application.

The business environment is characterized by constant change. Supply chain partners must realign business processes, and negotiate policy changes among partners. Enterprise information must support managers as they explore impacts of change and manage policy disagreements among partners. Adaptation requires knowledge management systems to completely and consistently support policy change.

REA modeling began over twenty-five years ago by identifying basic-level categories of resources, events, and agents that form a basic ontology, where an ontology is an “explicit and formal specification of a conceptualization of a domain of interest” (Gruber 1993). The REA ontology provides a rich expressive language to assist managers in explicitly capturing and sharing enterprise knowledge.

Typifications, groupings, associations provide extensions to the basic REA model. These extensions represent organizing forms for expressing policies that apply types and membership groupings to primitive elements. The extensions support non normative business process including standards and budgets.

Common modeling approaches for structuring REA concepts include entity-relationship diagrams, and dynamic and static unified modeling language (UML) diagrams (Amer 1993; Sugumaran and Storey 2006). REA business patterns assist model development by supporting compact forms for presenting, integrating and inspecting policy additions (Hruby 2006; Sugumaran and Storey 2006). REA models capture requirements and assist in directing applications development (Adamson and Dilts; Reuber 1990; Bergholtz, Jayaweera et al. 2003).

Currently REA extensions are being integrated with e-commerce standards though bodies such as the United Nations Business Process group, and the ISO Open-EDI group and ebXML business process team (McCarthy 2006). These standards apply REA to specify meta-models that refine business requirements and direct the development of information systems for coordinating exchanges(ISO/IEC 2002).

The extended REA and proposed Open-EDI reference models define an ontology of key components required in a business transaction. Enterprise concepts are modeled through UML class, activity and sequence diagrams to capture both the static and dynamic abstracts of electronic business transactions. Although diagrams support semantic descriptions of business exchanges they are, however, of limited value in helping computers effectively manipulate the REA ontology(Oshri, Pan et al. 2005).

As REA modeling is applied to the complex and unpredictable world of inter-enterprise collaboration, new approaches are needed to better understand, test and evolve enterprise ontologies. New tools are needed to share and assess policy changes, and then automatically apply changes to semantic models of enterprise information systems (Chen, McLeod et al. 1995).

Providing computer-based support for enterprise managers requires explicit and shared artifacts that support operations and policy decisions. Formal and machine executable ontologies are needed to support the capture, store, retrieve and share process-based knowledge (Mentzas, Apostolou et al. 2006). Formal ontologies support applications that can access and apply machine readable ontology. (Nonaka and Toyama 2003).

Rules can flexibly express relations among ontology concepts and support change efforts (Brodie and Ridjanovic 1984). Business rules explicitly capture practical reasons for taking action and REA modeling efforts are improved with rule-base reasoning. Embedding rules within an ontology supports automatic inference within semantic hierarchies (Orriëns, Yang et al. 2003; Zhang, Lin et al. 2005).

Semantic rule systems, such as CREASY (Conceptualizing REA Systems), reason about entities that include resources, events and agents (Geerts and McCarthy 2000). Rules in CREASY are represented as simple predicates that express stock flows among agents, events and resources. Predicate structures provide consistent, flexible and straightforward representation schemes. Rules organize predicate structures to support inferences, where for example, a rule may express "a sale requires a number, date and an amount." Representing knowledge in simple predicate structures provides opportunities for analyzing inference chains and detecting inconsistencies.

Semantic Web Technologies

Semantic web standards promises to integrate ontologies with rule-base inference to provide a clear and consistent way capture and apply business policies. Research is ongoing to represent REA concepts with semantic web technologies such as Web Ontology Language (OWL) that provides an open and standard based language to define ontologies (Gailly and Poels 2006). REA is supporting advances in ontological engineering designed for improving supply chain collaboration and knowledge sharing for the semantic web (Hessellund 2006). Machine-readable ontologies provide common vocabulary to support systemization, sharing and application of knowledge.

The Semantic Web Rule Language (SWRL) was officially introduced in 2004 to extend OWL semantics to support machine reasoning. Rule engines apply SWRL rules to derive logical extensions to OWL classes (Horrocks, Patel-Schneider et al. 2004). SWRL rules check the consistency of ontologies, and provide automated support to deduce relationships and update ontologies.

An enterprise ontology, expressed in OWL and SWRL, must practically and faithfully express REA basic-level primitives, typifications, groupings and associations as they relate to concepts within a domain of interest (Geerts and McCarthy 2000). An epistemologically adequate ontology provides sufficient detail and yet is intuitively understandable to enterprise designers (Visser and Bench-Capon 1998).

In addition, the ontology must be extensible and maintainable to accommodate iterative development of facts, concepts and relations. Thus small changes, such as the addition or revision of a concept type, must not require large changes in the ontology's structure. Gruber (1993) suggests that the maintainability of an ontology is improved by encapsulating details and representing concepts within generalization and specialization hierarchies.

Finally, the evolving ontology must support the goals of the business enterprise. A domain specific ontology should help managers better understand and share external and internal policies to assist with business process integration.

Modeling REA extensions in OWL and SWRL

UML originated as a graphical software modeling language to help experts capture a domain's representations, but in order to facilitate processing by machines the implicit knowledge embedded in these diagrams had to be translated to procedural code. The translations introduced a semantic gap between the high level graphical models and the low level executable code.

In response, UML tools have evolved to better help domain experts capture semantics with logical rule structures expressed through the object constraint language (OCL). By using OCL designers can augment UML models by logically filling specification gaps, checking constraints and deriving better procedural code. UML and OCL are not designed to be distributed over the web, but instead they are optimized to represent constraints within individual UML models (Gasevic, Djuric et al. 2006).

OWL is designed for open sharing among multiple applications and groups. OWL extends the semantic facilities of UML and OCL tools to support declarative and self-contained statements more directly usable by software applications (Baclawski, Kokar et al. 2002). OWL represents declarative metadata, ontology and logic to help distributed enterprises better exchange and integrate semantic information to coordinate their tasks. As new OWL statements are introduced, they are interpreted by logical inference techniques to maintain deductive consistency and dynamically derive new sets of associations.

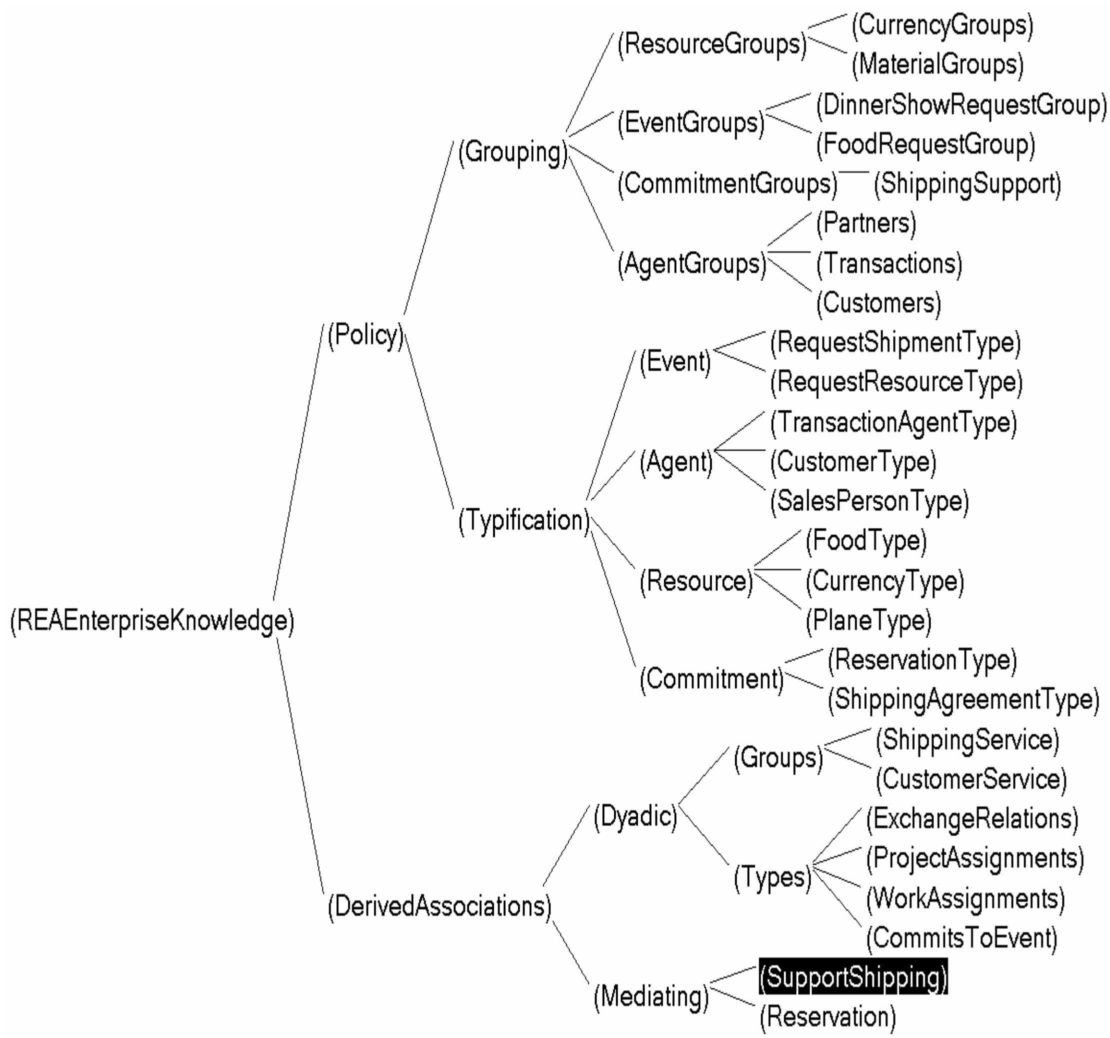
OWL's fundamental construct is a class representing collections of object instances. Generalization structures organize OWL classes to model similarities and differences between classes. Figure 1 presents a generalization hierarchy representing REA knowledge-level categories. All nodes represent subclasses except the *REA Knowledge* node which is the root class of the hierarchy.

The ontology expands from the root to capture two main subclass divisions: *Policy* and *Derived Associations*. *Policy* is the class that specializes into subclasses *Type*, expressing typification relationships and *Grouping* expresses membership relationships (Geerts and McCarthy 2006). *Type* nodes support REA categories of resources, events, agents and commitments that are used to classify homogenous collections of individual instances. *Grouping* nodes support REA categories that assign individual instances to one or more membership groupings.

The *Derived Association* node represents relations that describe semantic connections among *Groupings, Type,* and other *Derived Associations* subclasses. The subclass *Dyadic* includes subclasses *Types* and *Groupings.* The *Types* subclass organizes associations among policy types. *Dyadic Types* represents a pattern that relates collections of homogenous objects. For example*, Work Assignments* associate employee types (food service workers, cashiers) to resource types (food, cash).

Dyadic Groupings organize associations among and between *Policy Types* and *Policy Groupings.* *Groupings* represents a pattern that associates collections of heterogeneous objects (Hruby 2006). For example, the *Dyadic Groupings* relation *Customer Service* is a relationship among a heterogeneous set of *Customers* types categorized as members of either *Large* and/or *Profitable* groupings and the type classification *Experienced Sales Person*. Thus a *Customer Service* can model a manager's intent of assigning experienced sale people to important customers.

Figure 1. Below is an inheritance class hierarchy for organizing the extended REA ontology. The leaves of the graphic tree contain examples of domain specific subclass extension of *Derived Associations*, and *Policy* classes.



The *Derived Associations* subclass M*ediating* defines triadic relationships among *Policy* level *Types*, *Groupings* and other associations. Derived *Dyadic* and *Mediating* associations are formed by logical deductions that relate sets of resources, events and agents thorough business policies expressed as modular rules.

Rules provide a modular framework for expressing and evaluating a hierarchy of business requirements. Rules can represent generalized knowledge as well as specific business requirements and represents a formal specification that provides information for inference engines.

An inference engine fires a rule when its premises are satisfied and its consequent is added to working knowledge to form new associations. Table 1 presents a decision table that expresses the dyadic association rule *Commit Next Day*. This rule states that the assertion of a *Next Day Insured* shipping agreement requires a commitment to a bundle of specific *Shipping Support* services including afternoon delivery, morning pick-up and premium insurance. The inference engines fires this rule to deduce associations among commitments and resources.

Table 2 presents a decision table that expresses the mediating association rule *Executes Next Day Commitment.* When the *Commit Next Day* rule fires, the premise of the *Executes Next Day Commitment* rule is satisfied and it, in turn, asserts the need for *Shipping Provider Services* such air/ground transport, morning pickup, packaging and payment.

Deriving Associations with SWRL

An objective of the semantic modeling (Sowa 2000) web is to combine logic with ontology to formally reason and derive implicit knowledge. Rather than hard wiring dyadic and mediating associations to express instances rules and reasoning engines could automatically derive classifications. This allows managers to more easily express policies as rule structures that represent explicit, testable understandable policies.

Table 1. The *Commit Next Day* association as a dyadic business policy rule. The rule infers that an instantiation of a *Next Day Insured* shipping agreement requires a commitment to a bundle of specific *Shipping Support* services.

Dyadic Rule - **Commit Next Day**

IF (Premise) -> Next Day Insured*	
Next Day Insured -> asserted	✓
THEN (Consequent) -> Shipping Support	
Afternoon Delivery	✓
Export License	
Morning Pick-up	✓
Premium Insurance	✓
Special Payment	
Storage	

Table 2. The *Executes Next Day Commitment* as a mediating business policy rule. The rule infers that the firing of Table 1's *Commit Next Day* rule specifies a bundle of specific *Shipping Provider Services*

Mediating Rule – **Executes Next Day Commitment**

IF (Premise) -> Commit Next Day	
Commit Next Day -> rule fired	✓
THEN (Consequent) -> Shipping Services	
Air/Ground Transport Service	✓
Import/Export Service	
Morning Pick up Service	✓
Morning Packaging	✓
Payment Processor	✓

The following provides an example of SRWL rules to allow designers to logically express policy specifications to create and populate *Derived Associations*. SWRL rules support reasoning about *Dyadic* and *Mediating* relationships among REA instances, *Types* and *Groupings*.

Consider a manager that applies an ontology that provides the customer grouping categories of large, profitable, and potential. Individual customer instances are then assigned membership in one or more of the above groupings. The manager then develops a policy that guides decisions concerning assignment of sales people to customer accounts, for example a policy such as - "experienced sales people should handle large customers."

The intent of the above policy must be expressed in the ontology and allow automatic updating for any changes to individual instances, *Policy Types* or *Groupings*. As new customers are added or removed as members of the "large customer" membership group, the policy must be automatically applied and reflect new associations between sales people and customers. Also if the ontology's hierarchical structure changes, such as providing new subcategory "high revenue" as a subclass of "large" customers, then the policy must be applied regardless of a customer's assignment to a specialized typification subcategory, since "high revenue" customers are still "large" customers.

The above policy can be expressed more formally as:

IF
For the set (X) composed of every Sales Person instance of type "Experienced"
AND for the set (Y) composed of every Customer instance that is a member of the grouping category "Large"
AND there exists a *Dyadic Grouping* association category CustomizedQualityService
THEN
CustomizedQualityService's "hasRightHandParticipants" property has the set (X) of instances of "Experienced" Sales People
AND
CustomizedQualityService's "hasLeftHandParticipants" property has the set (Y) of instances of "Large" Customers.

The above rule is expressed succinctly in SWRL as:

ExperiencedType(?x) ∧ Large(?y) ∧ CustomizedQualityService(?z)

→

hasRightHandMembership(?z, ?x) ∧ hasLeftHandMembership(?z, ?y)

The above rule represents an IF -THEN structure where the symbol → separates the IF part, on the top, from the THEN part below. The variables in the above predicate pattern are prefixed with the symbol '?' and represent unbound placeholders representing instance sets. The symbol ∧ represents a conjunction operator linking predicates. The SWRL rules populate the inverse properties "hasRightHandParticipants" and "hasLeftHandParticipants" attributes to derive a dynamic set of associations among instances.

The OWL ontology combined with SWRL offers a new modeling formalism for representing extended REA relationships. SWRL provides explicit machine executable representations that allow enterprise designers to formally express policies and apply policy updates. The following explores an application of OWL and SWRL rules to capture and share evolving business policies.

OWL and SWRL models for Inter-Enterprise Collaboration

We participated in a field study of an enterprise project where we studied design artifacts and participated with managers as they adapted enterprise business policies to cope with ongoing and turbulent changes. The name and revealing details of the participating enterprises are fictionalized to assure anonymity.

Analysis artifacts and discussions with managers led to a consensus on REA types, grouping and associations. We reviewed use-cases, design documents, correspondence and participated in design decisions with project architects and managers. The analysis identified resources, events, agents and commitments that were then categorized into types, groupings and association classes according the extended REA model. The resulting sets were coded into the REA extended OWL and SWRL ontology. We assessed the utility of the extended REA ontology to logically organize change efforts, maintain consistency and completeness to provide an audit trail, and access OWL and SWRL as tools to support partner negotiations.

Field Study Background

DEAL is a pseudonym for a cooperating group of enterprise partners whose mission is to provide innovative and web-based customer services to sell nearly new vehicles obtained from rental car agencies. Rental car agencies remove vehicles from their rental fleets when those vehicles are slightly over six months old or have mileage of over ten thousand miles and are therefore considered not suitable for rental. The DEAL partnership acquires de-fleeted rental vehicles, refurbishes and warranties those vehicles and then offers them for sale. The vehicles are offered for sale through an e-commerce web site that assists customers in researching vehicles, selecting vehicles, configuring options, acquiring deposits, processing trade-ins, obtaining warranties and securing financing.

The business model requires multiple and geographically dispersed partners to cooperate across an international supply chain. Financial partners include numerous banks and credit processors that offer customer financing. Customer service partners process customer vehicle requests and attempt to match those requests to current and planned inventory. Marketing partners offer customer referrals, discounts and promotions. Rental car partners supply vehicles for sale as they are removed from their rental fleet. Other rental car agencies coordinate test drives for vehicles types that match customer requests. Refurbishing and warranty partners prepare and certify vehicles and logistics suppliers transport vehicles to and from suppliers and ultimately deliver purchased vehicles to customers. Vehicle data partners support the web site's back office databases by supplying timely data concerning vehicle models and option descriptions, trade-in valuations and pricing information.

The managing organization is located in Belgium which has responsibility to refine the business model and coordinate partner negotiations. Belgium-based client participants include senior management representatives, business leads and customer service managers, software architects, technical leads and project managers.

A core set of resource decisions were documented to capture partner responsibilities. We reviewed over one-hundred use cases that specified system and partner processes ranging from initial vehicle research to processes for a customer to return unsatisfactory vehicles.

In early June, DEAL began web site development but by July the evolving business model and changing partnerships forced the management and development teams to continually re-align priorities. As development continued, the business requirements team identified nearly two-hundred outstanding requirement changes. New banking partnership agreements and new business policies for English and German markets forced the requirements team to continually reconsider past decisions. The relationships and strategic enterprise agreements among partners were turbulent and marked by complex negotiations.

Field Study Results

We reviewed project documentation including use-cases, UML based design documents, partner correspondence and business planning documents. Follow up discussions with the project architect supported ontology development as we identified and captured policy concerns within REA extended types, grouping and associations. The ontology was then coded in OWL and SWRL using the Protégé ontology editor (O'Connor, Knublauch et al. 2005). The following summarizes key parts of the ontology and the subsequent discussion assesses its epistemological adequacy.

The model is organized by membership groupings and typifications as previously discussed. Figure 5 presents resulting domain specific membership grouping and Figure 6 presents the typification hierarchy applied to classify domain instances.

Membership groupings represent a hierarchy that can be used to assign membership categories for instances. For example, REA agent groups such as vehicle specialists may provide a range services such as vehicle configuration and trade-in assistance, while another may provide trade-in assistance and vehicle location assistance.

Typifications represent a hierarchy to categorize instances with either similar properties, or conceptual meaning. For example, partners negotiate with customers by offering a virtual vehicle that represent a make and model that does not physically exist but instead is initially offered to customers to begin the sales process. An actual vehicle's make and model, while containing essentially the same properties as a virtual vehicle represents a different business concept. The typification hierarchy identifies independent REA resource types of virtual and actual vehicles.

Figure 7 presents *Derived Associations* that relate typifications and groupings. Derived associations represent association classes that help identify and align policy interactions among collaborating partners. For example, arranging a customer test drive involves associations among multiple stakeholders including agents (sale agent, test drive agent), events such as payment arrangements (deposits, mileage restrictions), resources (vehicle type, availability and location) and commitments (reservations, purchaser interest in a specific vehicle).

Figure 5. Class and subclass categories to classify REA elements into DEAL membership groupings

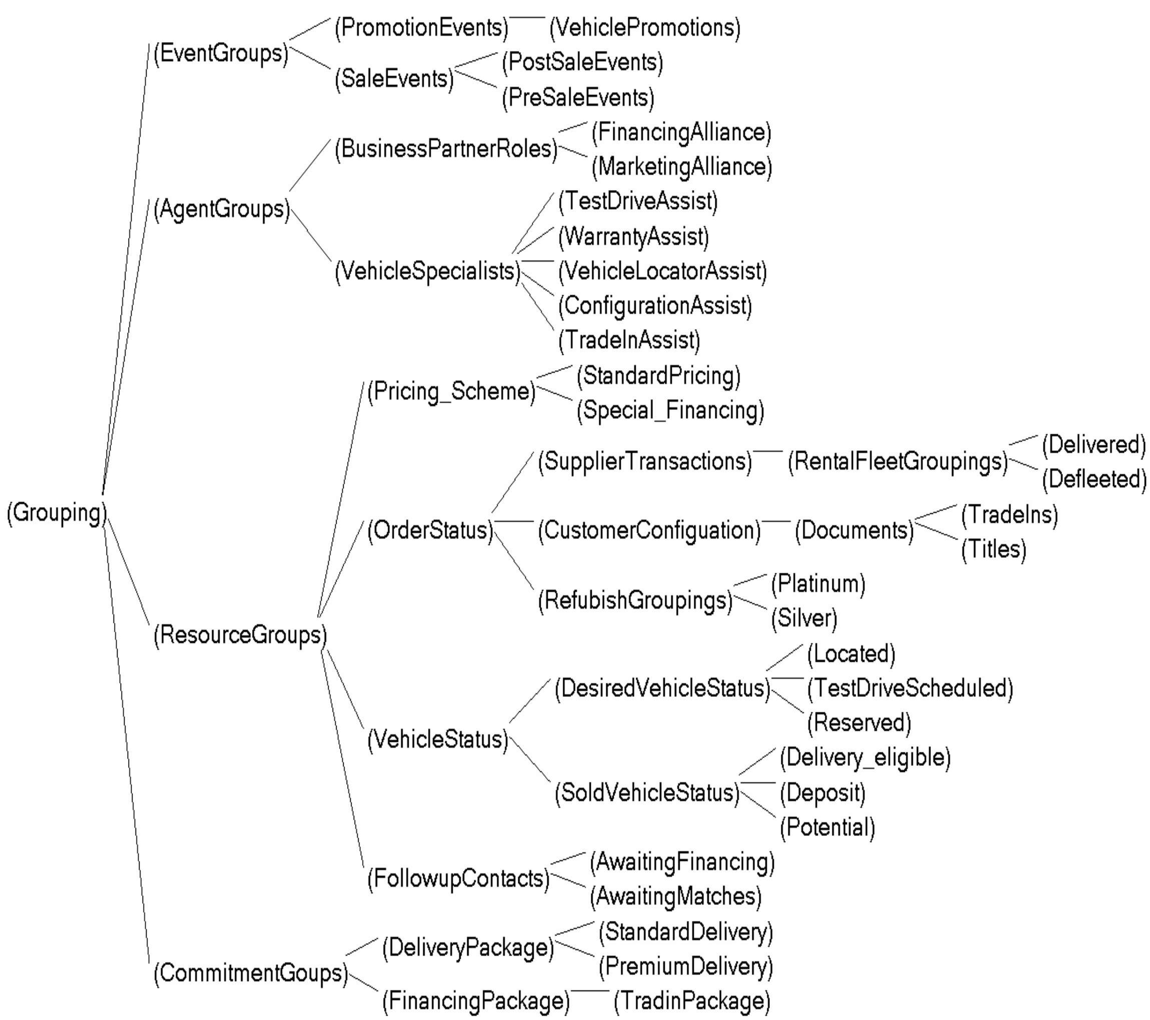


Figure 6. Type collections hierarchies define characteristics of REA elements for DEAL partners.

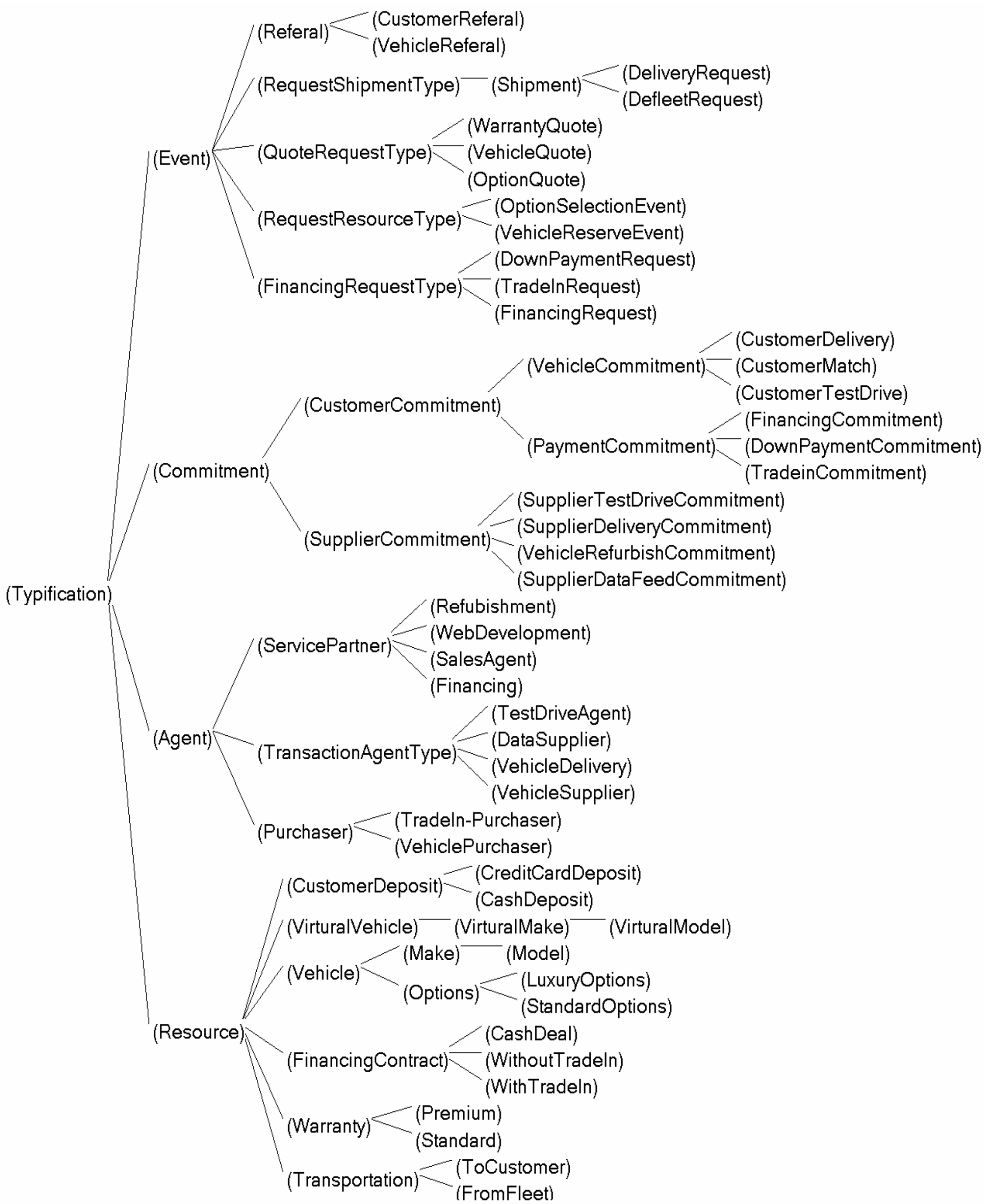
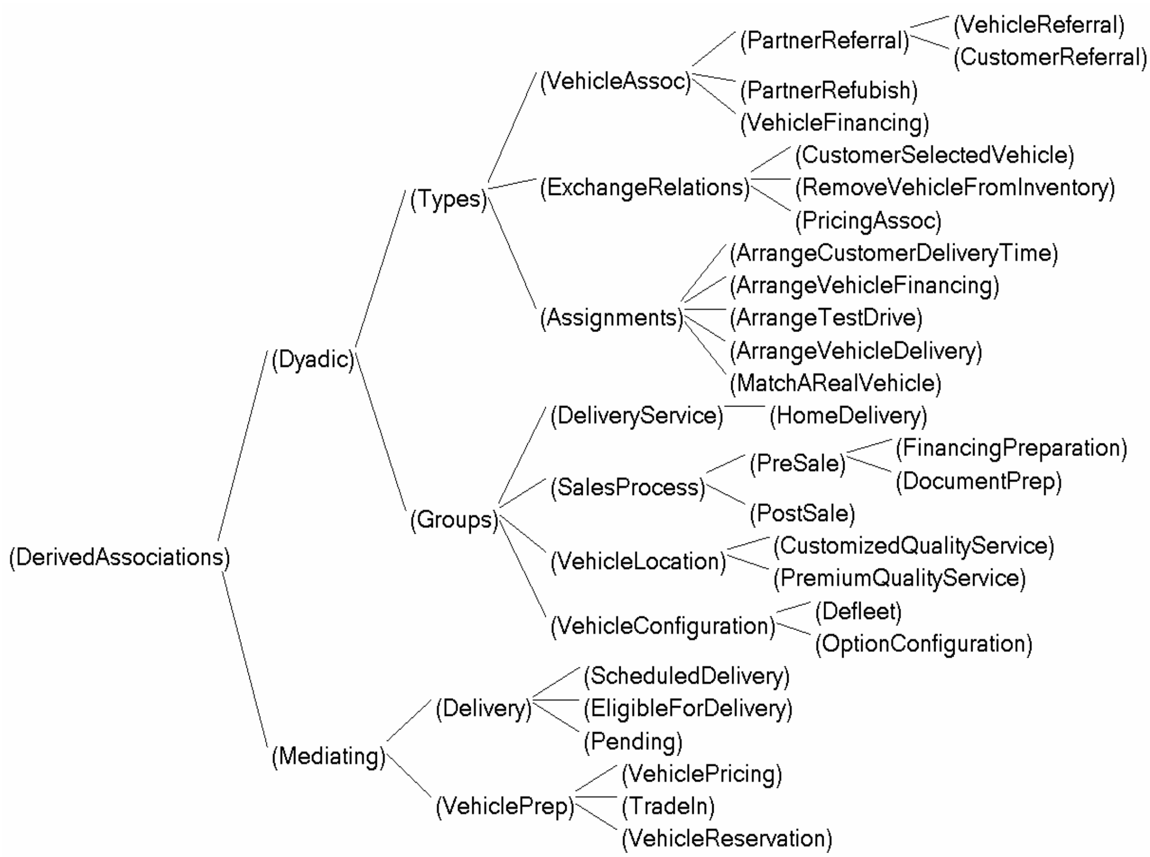


Figure 7. Derived association classes for Deals' partner associations



Business rules operate on the extended REA grouping, types and association classes to derive each association class's properties. We began modeling association classes by directly assigning instance to association class properties. It soon became clear that, due to frequent change, that manual assignment of types and membership groupings to associations was neither maintainable nor scalable. Model changes required frequent additions including changes to typification hierarchies, new grouping categories, and identification of new resource, event, agent and commitment instances. For example adding a new marketing partner that provided customer referrals for Peugeot cars required manually updating multiple association classes including financing, marketing referrals, matching and locating vehicles.

SWRL rules automated the process of populating and maintaining association classes. This eased maintenance since updating association class properties required only updating rules or creating new rules. The analysis identified over sixty rules that explicitly capture business policy relationships in derived association classes. Tables 2 and 3 illustrates a sample that covers a representational spectrum of SWRL rules dealing policy associations among combinations of resource, event, agents and commitments.

Table 4 illustrates mediating associations that combine dyadic associations with each other and with other typifications and groupings. For example, as expressed in the first rule in Table 4, a customer request to reserve a specific vehicle requires removal of that vehicle from inventory. The second rule captures policies for vehicle delivery and explicitly defines that a scheduled delivery involves a customer arranged delivery time and a supplier arranged vehicle delivery. The third rule specifies that in addition to customer policy, referred to in the second rule, the customer's nearly new vehicle is not eligible for delivery until that customer has arranged financing.

Table 2's Agent-to-Agent rules models relationships between agent types that collaborate to coordinate transactions across enterprises. For example, a sales agent must arrange a test drive to allow a customer to test the specific model of their selected vehicle. The Agent-to-Agent rule helps the sale agent find the right rental agent by dynamically combining and reasoning about distributed sets of partner ontologies.

Application of this Agent-to-Agent rule relies on each partner rental agency maintaining a compliant REA ontology at their site that includes vehicle resources, current rental commitments, scheduled events and rental agents. A central repository maintains a set of web-based hyperlinks directed to the ontologies housed at each rental agency. In response to a sales agent test drive query, the SWRL rule dynamically integrates sets of distributed ontologies to help schedule test drives.

The field study found enterprise designers did not think alike, business requirements continually changed, UML modeling and design approaches were not consistent across distributed design groups. The central problem was difficulty in coordinating impacts of business policy changes among partners.

The field study showed that REA types, grouping and associations form a comprehensive and consistent organizing framework for distributed enterprise knowledge. The extended REA ontology provided a shared vocabulary that allowed designers to better model and coordinate ongoing changes to policies. The upper-level REA ontology when combined with OWL and SWRL web technologies provides a concise declarative representation to support web-based reasoning, sharing and reuse across distributed partnerships.

Table 2. Example SWRL rules expressing characterizing associations among REA Agent types and membership groupings.

Example Agent Policy Descriptions		SWRL Rule
Agent /Agent Type-Type	Agent Type **Test Drive Agent** and Agent Type **Sales Agent** participate in **Arranging a Test Drive**	TestDriveAgent(?x) ∧ SalesAgent(?y) ∧ ArrangeTestDrive(?z) → hasLeftHandParticipants(?z, ?x) ∧ hasRightHandParticipants(?z, ?y)
Agent/Resource Type-Type	Agent Type **Sales Agent** and Resource Type **Virtual Vehicle** participate in a **Matching a Real Vehicle**	SalesAgent(?x) ∧ VirturalVehicle(?y) ∧ MatchARealVehicle(?z) → hasLeftHandParticipants(?z, ?x) ∧ hasRightHandParticipants(?z, ?y)
Agent/Resources Group -Group	Group Members providing **Configuration Assistance** and customer group members in the process of **Customer Vehicle Configuration** require an **Option Configuration**	ConfigurationAssist(?x) ∧ CustomerVehicleConfiguation(?y) ∧ OptionConfiguration(?z) → hasLeftHandParticipants(?z, ?x) ∧ hasRightHandParticipants(?z, ?y)
Agent/Event Group-Group	Members of the **Marketing Alliance** support **Vehicle Promotions** to assist the **Sales Process**	MarketingAlliance(?x) ∧ VehiclePromotions(?y) ∧ SalesProcess(?z) → hasLeftHandParticipants(?z, ?x) ∧ hasRightHandParticipants(?z, ?y)
Agent/Commitment Type-Group	The agent for providing **Financing** service must work with customer members committed to the **Finance Package** in the **Financing Preparation**	Financing(?x) ∧ FinancingPackage(?y) ∧ FinancingPreparation(?z) → hasLeftHandParticipants(?z, ?x) ∧ hasRightHandParticipants(?z, ?y)

Table 3. Example SWRL rules expressing characterizing associations among REA Commitments, Events and Resources.

Example Commitment – Event – Resource Policy Descriptions		SWRL Rule
Commitment/Commitment Group-Group	Members committing to a **Delivery Package** must commit to members of a **Financing Package** to **Arrange Vehicle Financing**	DeliveryPackage(?x) ∧ FinancingPackage(?y) ∧ ArrangeVehicleFinancing(?z) → hasLeftHandParticipants(?z, ?x) ∧ hasRightHandParticipants(?z, ?y)
Commitment /Resource Group-Type	Members committing to a **Delivery Package** must specify a **Vehicle** resource type before they can **Arrange Vehicle Delivery**	DeliveryPackage(?x) ∧ Vehicle(?y) ∧ ArrangeVehicleDelivery(?z) → hasLeftHandParticipants(?z, ?x) ∧ hasRightHandParticipants(?z, ?y)
Commitment/Event Group -Type	Members committed to **Standard Delivery** are require execution of the Event **Delivery Request** to **Arrange Delivery Time**	StandardDelivery(?x) ∧ DeliveryRequest(?y) ∧ ArrangeDeliveryTime(?z) → hasLeftHandParticipants(?z, ?x) ∧ hasRightHandParticipants(?z, ?y)
Event/Event Type-Type	A **Vehicle Reserve Event** requires a **Down Payment Request** to **Remove a Vehicle From Inventory**	VehicleReserveEvent(?x) ∧ DownPaymentRequest(?y) ∧ RemoveVehicleFromInventory(?z) → hasLeftHandParticipants(?z, ?x) ∧ hasRightHandParticipants(?z, ?y)
Event/Resource Type-Group	A **Vehicle Reserve Event** And a **Vehicle** type are involved in **Select Vehicle**	VehicleReserveEvent(?x) ∧ Vehicle(?y) ∧ CustomerSelectedVehicle(?z) → hasLeftHandParticipants(?z, ?x) ∧ hasRightHandParticipants(?z, ?y)
Resource/Resource Type-Group	A type of **Financing Contract** and a **Vehicle** type are involved in a **Vehicle Financing**	FinancingContract(?x) ∧ Vehicle(?y) ∧ VehicleFinancing(?z) → hasLeftHandParticipants(?z, ?x) ∧ hasRightHandParticipants(?z, ?y)

Table 4. Mediating associations combine dyadic associations with each other and with other typifications and groupings.

Example Mediating Policy Descriptions		SWRL Rule
Event-Event + **Event-Resource** Mediating Relation	If **Customer Selected a Vehicle** and we **Remove Vehicle from Inventory** then we create a **Vehicle Reservation**	CustomerSelectedVehicle(?x) ∧ RemoveVehicleFromInventory(?y) ∧ VehicleReservation(?z) → hasLeftHandRelation(?z, ?x) ∧ hasRightHandRelation(?z, ?y)
Commitment- Event + **Commitment-Resource** Mediating Relation	If there is an **Arranged Delivery Time** and an **Arranged Vehicle Delivery** then we create a **Scheduled Delivery**	ArrangeDeliveryTime(?x) ∧ ArrangeVehicleDelivery(?y) ∧ ScheduledDelivery(?z) → hasLeftHandRelation(?z, ?x) hasRightHandRelation(?z, ?y)
Mediating Relation + **Commitment – Commitment** Mediating Relation	If we have created a **Scheduled Delivery** and we **Arranged Vehicle Financing** then according to policy the vehicle is **Eligible for Delivery**	ScheduledDelivery(?x) ∧ ArrangeVehicleFinancing(?y) ∧ EligibleForDelivery(?z) → hasLeftHandRelation(?z, ?x) ∧ hasRightHandRelation(?z, ?y)

Discussion

The DEAL project was characterized by constant changes in requirements, partnership agreements and business models. DEAL is typical of inter-enterprise collaborations that must continually align supply chains and adapt to new business environments.

OWL and SWRL models of the extended REA ontology support machine executable representations to help distributed set of mangers assess and coordinate business policy change. SWRL provides an explicit machine executable representation of associations that allows inter-enterprise designers to formally express their policies, experiment with updates and quickly incorporate changes in the extended REA ontology. The field study analysis suggests semantic tools that model the extended REA model effectively capture and apply policy-level knowledge.

OWL and SWRL representations provide a shareable and standard way to organize a policy-based ontology. OWL relies on extensible mark-up language (XML) data syntax and standard communication protocols to transport the ontology to geographically dispersed partners. Internet applications can then directly manipulate OWL to allow enterprise partners to cooperate in geographically distributed policy design.

We are investigating a prototype information sharing framework to support web-based inter-enterprise collaboration. The prototype applies OWL and SWRL to model the extended REA ontology, where the semantic model is maintained in a central repository. Partners submit incremental updates to the model and propose new business rules. SWRL rules then derive new policy associations, populate association class attributes and apply consistency checking tools identify potential semantic conflicts.

The prototype then applies the extended REA ontology's updated representations to allow distributed designers to selectively extract policy information through built-in and customized queries. In the case of DEAL standard queries might include:

Query 1: Show all policies related in some way to customer quotes,
Query 2: Show associations between rental agencies and vehicles

Query 3: Show any relationships involving vehicle X
Query 4: Show financial constraints imposed on vehicle delivery
Query 5: Show all payment connections involving customer Y
Query 6: Show all deposits collected on vehicles

The prototype's design attempts to overcome limitations inherent in interpreting graphical notations by explicitly capturing intentions of managers, automating consistency checks and promoting distributed information sharing.

Conclusion

We presented an executable ontological representation of the extended REA enterprise model by applying semantic web technologies. We illustrated the use of OWL to capture typification, membership groupings and association semantics. We also identified SWRL rules that automatically derived attributes of association classes. The field study suggests OWL and SWRL offer a maintainable and scalable way to capture and quickly update evolving partnership policies.

The field study demonstrates the effectiveness of applying semantic web technologies to model REA extensions. Our investigation lends support to future efforts that apply semantic web technologies with the extended REA ontology to coordinate partner negotiations.

Limitations of the field study include a post-hoc analysis that addressed the design of the ontology only after issues were already identified through design and business documents. The resulting OWL models therefore reflect subjective policy interpretations by investigators rather than on-going negotiations among collaborating partners. Future studies should investigate settings that study partner co-development of the extended

REA ontology and that allows partner to apply OWL and SWRL to define and mitigate changes to inter-enterprise collaborations.

Literature Cited

Adamson, I. L. and D. M. Dilts "Development of an Accounting Object Model from Accounting Transactions." Journal of Information Systems 9(1): 43.

Amer, T. S. (1993). "Entity-Relationship and Relational Database Modeling Representations for the Audit Review of Accounting Applications: An Experimental Examination of Effectiveness." Journal of Information Systems 7(1): 1.

Baclawski, K., M. K. Kokar, et al. (2002). "Extending the Unified Modeling Language for ontology development " Software and Systems Modeling 1(2): 142-156.

Bergholtz, M., P. Jayaweera, et al. (2003). Process Models and Business Models –A Unified Framework. 22nd International Conference on Conceptual Modeling (ER 2003), Chicago,IL, Springer.

Brodie, M. and D. Ridjanovic (1984). On the Design and specification of Database Transactions". In On O. C. Modeling. New York, NY, Springer-Verlag.

Chen, J.-L., D. McLeod, et al. (1995). "Domain-knowledge-guided Schema Evolution for Accounting Database Systems " Expert Systems With Applications 9(4): 491-501.

Dunn, C. L. and S. V. Grabski (2000). "Perceived semantic expressiveness of accounting systems and task accuracy effects." International Journal of Accounting Information Systems 1: 79-87.

Gailly, F. and G. Poels (2006). Towards an OWL-formalization of the Resource Event Agent Business Domain Ontology. Formalization REA Business Domain Ontology. Ghent University Department of Economics and Business Administration.

Gasevic, D., D. Djuric, et al. (2006). Software Engineering Approaches to Ontology Development. Model Driven Architecture and Ontology Development. Heidelberg, Springer-Verlag: 145-180.

Geerts, G. L. and W. E. McCarthy (1999). "An Accounting Object Infrastructure for Knowledge-based Enterprise Models." IEEE Intelligent Systems & Their Applications 14(4): 89-94.

Geerts, G. L. and W. E. McCarthy (2000). "Augmented Intensional Reasoning in Knowledge-Based Accounting Systems." Journal of Information Systems 14(2): 127.

Geerts, G. L. and W. E. McCarthy (2002). "An ontological analysis of the economic primitives of the extended-REA enterprise information architecture." International Journal of Accounting Information Systems (2002)(3): 1-16.

Geerts, G. L. and W. E. McCarthy (2006). "Policy-Level Specifications in REA Enterprise Information Systems " Journal of Information Systems(Fall 2006).

Gruber, T. (1993). "A Translation Approach to Portable Ontologies." Knowledge Acquisition 5(2): 199-220.

Hessellund, A. (2006). Supply Chain Modeling with REA, IT University of Copenhagen, Denmark. TR-2006-80: 1-15.

Hevner, A. R., S. T. March, et al. (2004). "Design Science in Informatin Systems Research." MIS Quarterly 28(1): 75-105.

Horrocks, I., P. F. Patel-Schneider, et al. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML, World Wide Web Consortium (W3C).

Hruby, P. (2006). Model-Driven Designs Using Business Patterns. Berlin, Germany, Springer.

ISO/IEC (2002). Information technology -- Business agreement semantic descriptive techniques -- Part 1: Operational aspects of Open-edi for implementation. I. t. Joint Technical Committee ISO/IEC JTC 1, Subcommittee SC 32, Data management and interchange.

Lampe, J. C. (2002). "Discussion of an Ontological Analysis of the Economic Primitives of the Extended-REA Enterprise Information Architecture." International Journal of Accounting Information Systems (2002)(3): 17-34.

McCarthy, W. E. (1982). "The REA accounting model: A generalized framework for accounting systems in a shared data environment." The Accounting Review July: 554-578.

McCarthy, W. E. (2006). ISO/IEC 15944-2:2006. The 2nd International REA Technology Workshop. Santorini Island, Greece.

Mentzas, G., D. Apostolou, et al. (2006). "Inter-organizational Networks for Knowledge Sharing and Trading." Information Technolgy Management 7: 259-276.

Nonaka, I. and R. Toyama (2003). "The knowledge-creating theory revisited: Knowledge Creation as a Synthesizing Process." Knowledge Management Research & Practice (2003)(1): 2-10.

O'Connor, M., H. Knublauch, et al. (2005). Supporting Rule System Interoperability on the Semantic Web with SWRL. 4th International Semantic Web Conference (ISWC 2005), Galway, Ireland, Springer-Verlag.

Orriëns, B., J. Yang, et al. (2003). A Framework for Business Rule Driven Web Service Composition. Conceptual Modeling for Novel Application Domains, ER 2003 Workshops ECOMO, IWCMQ, AOIS, and XSDM, Chicago, IL, USA, Springer.

Oshri, I., S. L. Pan, et al. (2005). "Trade-offs Between Knowledge Exploitation and Exploration Activities." Knowledge Management Research & Practice (2005)(3): 10-23.

Reuber, A. R. (1990). "CO-STAR: A Semantic Representational Schema for Cost Management." Journal of Information Systems 4(2): 15.

Sowa, J. F. (2000). Knowledge Representation: Logical, Philosophical, and Computational Foundations: Logical, Philosophical, and Computational Foundations Boston, Brooks/Cole Thomson Learning.

Sugumaran, V. and V. C. Storey (2006). "The Role of Domain Ontologies in Database Design: An Ontology Management and Conceptual Modeling Environment." ACM Transactions on Database Systems, 31(3): 1064-1094.

Verdaasdonk, P. (2003). "An Object-Oriented Model for Ex Ante Accounting Information." Journal of Information Systems 17(1): 43.

Visser, P. and T. J. M. Bench-Capon (1998). "A Comparison of Four Ontologies for the Design of Legal Knowledge Systems." Artificial Intelligence and Law 6: 27-57.

Zhang, W., L. Lin, et al. (2005). An Ontology-Based Functional Modeling Approach for Multi-agent Distributed Design on the Semantic Web. Computer Supported Cooperative Work in Design II, 9th International Conference CSCWD 2005, Coventry, UK.