A Meta-model of Business Interaction for Assisting Intelligent Workflow Systems

Areti Manataki and Yun-Heh Chen-Burger

Centre for Intelligent Systems and their Applications, School of Informatics,
The University of Edinburgh, Informatics Forum, 10 Crichton Street, Edinburgh EH8 9AB, UK
A.Manataki@sms.ed.ac.uk, Jessicac@inf.ed.ac.uk

Abstract. Business interaction is a significant aspect of intelligent workflow systems, which are envisaged to deal with dynamic and interorganisational environments. However, automating business interaction in an intelligent way is not an easy task. To address this issue, this paper presents an ontological approach for modelling business interaction. The developed meta-model can support the process of automating business interaction, as well as assist its automated analysis. The ontology's development is described and its application is demonstrated through an example use case.

Keywords: intelligent workflow, business interaction, ontology, business action theory.

1 Introduction

Today's business environment is characterised as dynamic, uncertain, distributed, heterogeneous and collaborative. This business reality, often taking the form of virtual organisations and extended enterprises, enforces challenging constraints on workflow management systems. On this basis, a growing body of research has attempted to provide intelligent workflow systems addressing the following properties [1, 2, 3, 4]: Firstly, distributed, interorganisational and decentralised architectures are preferred, such as multiagent systems. Secondly, well-defined semantics are needed to support a common understanding of domain knowledge. Thirdly, advanced reasoning and learning capabilities are employed to allow for adaptive and flexible workflows. Lastly, a high degree of automation is required for the different phases of the workflows' lifecycle.

Business interaction is an important aspect of intelligent business workflows, especially given the highly distributed nature of the business environment. In a multiagent setting, for example, an ineffective communication and interaction between the agents can lead to a failure of the overall workflow. However, automating business interaction in an intelligent way is not an easy task. Such an intelligence requires the above-mentioned properties for the automated business interaction, so as to allow its analysis and automated improvement.

In this paper we argue that a formal model of business interaction can support the process of automating business interaction. Recognising Business Action Theory

J. O'Shea et al. (Eds.): KES-AMSTA 2011, LNAI 6682, pp. 692–701, 2011.

(BAT) as a useful framework of business interaction, we formalise it in the form of an ontology. This meta-model has clear semantics, suitable for automated reasoning, and hence can facilitate the analysis of business interaction.

The remainder of this paper is as follows: Section 2 provides background information on the adopted modelling framework, and describes Business Action Theory, upon which the ontology is based. Section 3 presents the development of the BAT ontology, and discusses outputs of different phases. A scenario of use is described in Section 4, while Section 5 concludes.

2 Background Information

Business Action Theory [5, 6, 7, 8] is a generic framework of business interaction. Based on communicative action theories and business relationship theories, it describes the interaction between a supplier and a customer with respect to communication and exchange of value. It consists of a phase model and supplementary conceptual models. Six phases of business interaction are identified: i) business prerequisites phase, ii) exposure and contact search phase, iii) contact establishment and proposal phase, iv) contractual phase, v) fulfilment phase and vi) completion phase. Each phase involves some exchange, respectively: i) knowledge about business prerequisites, ii) interests, iii) proposals, iv) commitments, v) value and vi) assessments, which can be acceptances or claims. In [6] three additional conceptual models are provided, describing i) the transition from business prerequisites to a contract, ii) the contract as a base for fulfilment and iii) the results of fulfilment. According to the first conceptual model, the supplier's ability is translated into an offer, which is equivalent to a sales proposal. From the customer's side, an operation can have some lack and need, giving rise to desire and demand. This desire and demand can be expressed as a purchase proposal. Since several proposals may be exchanged, it is typical that a received proposal may lead to a reformulation of the supplier's offers and the customer's desire and demand. When a proposal is accepted, it is transformed into a contract, while in the opposite case no deal is achieved.

Furthermore, BAT recognises five layers of business interaction [8]: i) business acts, ii) action pairs, consisting of two interrelated business acts, iii) exchanges, consisting of one or more action pairs of the same type, iv) business transactions, consisting of exchanges of interests (possibly), proposals (possibly), commitments, value and assessments, and v) transaction groups, consisting of one or more business transactions and frame contracting. Being a theoretical framework of business interaction, BAT is useful for understanding and designing business interactions; however, in its current form, it cannot be used as a direct input when automating and implementing business interactions within a workflow system.

The *three-layered Business Process Modelling framework* [9] supports the development of workflow management systems, as well as generic software systems, by separating the business logic from the implementation logic. Its three layers are: i) The business layer captures the business requirements in higher-level descriptions that may be formal or informal. ii) The logical layer formalises the captured business requirements in a machine-understandable language, while incorporating logical or operational requirements. iii) The implementation layer provides algorithms for the implementation of the logical model, while respecting overall system requirements.

3 Business Action Theory Ontology

With the aim of automating business interaction for intelligent workflows, and following the three-layered Business Process Modelling framework, we propose and develop a meta-model of business interaction. Business Action Theory is identified, in its current form, at the business layer of the framework. In order to transform it into a model at the logical layer, we develop a BAT ontology. This logical model can be particularly useful at web, knowledge-intensive environments.

The METHONTOLOGY [10, 11] approach is adopted for developing the BAT ontology, as it is a detailed, mature and widely accepted methodology for ontological engineering. METHONTOLOGY identifies five phases during the ontology life cycle (i.e. specification, conceptualisation, formalisation, implementation and maintenance), during which specific tasks take place. In the rest of this section, we will discuss outputs of the first four phases.

3.1 Specification and Knowledge Acquisition

The purpose of the BAT ontology is to support the development of intelligent workflows. The primary objective is to provide a formal, logical model of business interaction that can facilitate the automation of business interaction, while a secondary objective is the automated analysis of business interaction.

The source of knowledge for the BAT ontology is the literature on Business Action Theory, as presented in Section 2. It is worth mentioning that BAT has been refined and enriched since it was first introduced in [5]. However, this refinement has not been exhaustive, and several of the concepts and models that can be seen in the first versions of BAT are still applicable in the latest versions. Therefore, [5, 6, 7, 8] are the sources of knowledge for this ontology.

As far as the scope of the ontology is concerned, we distinguish the following four concept categories: i) phase model, as presented in [6], ii) additional concepts of the phase model, as presented in [7], iii) supplementary conceptual models, as presented in [6] and iv) business interaction layers, as presented in [8]. Finally, the granularity level of the ontology is the same as the one of BAT literature.

3.2 Conceptualisation

The acquired knowledge is conceptualised during this phase into a set of intermediate representations of the domain. METHONTOLOGY suggests a sequence of eleven tasks to support this activity, with outputs including among others a glossary of terms, a concept dictionary and a list of axioms. In this section the outputs of the first four tasks will be discussed.

Terms of the BAT domain were firstly gathered in a *Glossary of Terms* that contained 132 concepts, attributes and relations. An extract can be seen in Table 1, where some basic terms are described. For example, a description of the concepts "Business Role" and "Business Interaction" is provided, along with the relation between them "participatesInInteraction", i.e. a Business Role participates in some Business Interaction.

Table 1. Extract from the Glossary of Terms

Name	Synonym	Acronym	Description	Type
Business Interaction	--	--	An interaction between a supplier and a customer for a single business transaction	Concept
Business Interaction Phase	--	--	A phase of the business interaction process	Concept
Contractual Phase	--	--	A phase where a contract is signed among the business parties (4th phase of BAT's phase model)	Concept
Commitment Exchange	--	--	An exchange of commitments	Concept
Commitment	Promise, obligation	--	An engagement caused by contract	Concept
Business Role	--	--	The role that a business actor plays during a business transaction	Concept
participatesInInteraction *(BusinessRole, BusinessInteraction)*	--	--	An interaction in which a business role participates	Relation
hasPhase *(BusinessInteraction, BusinessInteractionPhase)*	--	--	A phase of a business interaction	Relation
isAccepted	--	--	Whether a proposal is accepted or not (boolean type)	Instance Attribute

A *Concept Taxonomy* was then defined, an extract of which can be seen in Figure 1. According to this classification, the concepts "Payment Promise" and "Delivery Promise" make up a partition of the concept "Commitment", which was described in Table 1. Furthermore, "Commitment" belongs to the partition of "Object of Exchange".

Relationships between concepts were also established and visualised in a set of *Binary Relations Diagrams.* Two such diagrams are presented in Figures 2 and 3. Figure 2 shows, among others, the relation "participatesInInteraction" that was presented in Table 1, while Figure 2 corresponds to a conceptual model from [6] that was described in Section 2.

Based on the outputs of the previous tasks, a *Concept Dictionary* was developed for BAT. Table 2 presents some concepts, along with the relations in which they participate.

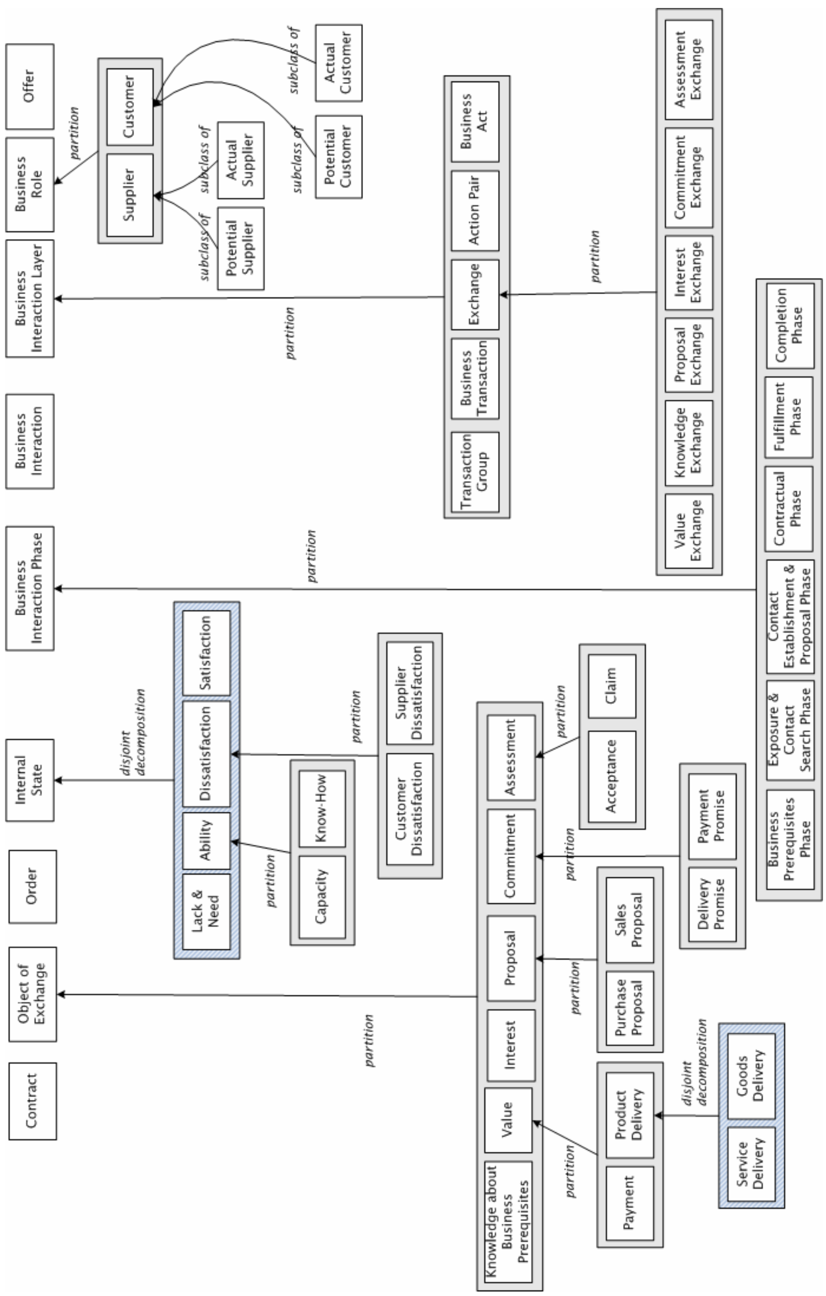

Fig. 1. Extract from the Concept Taxonomy

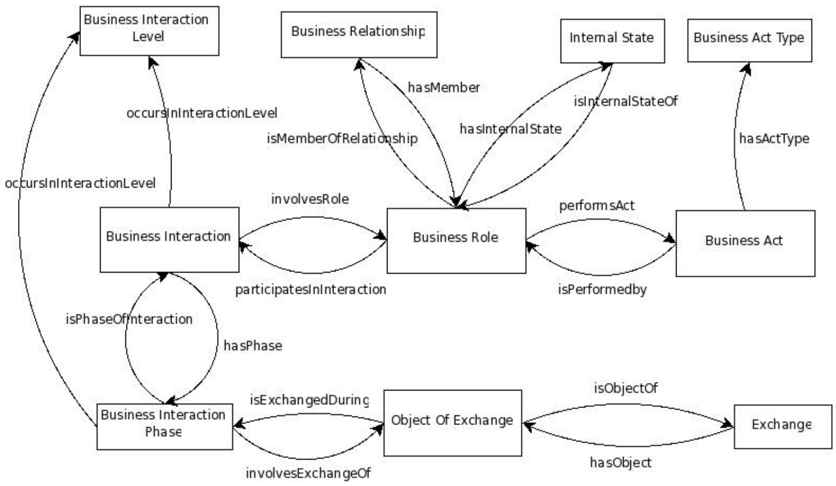

Fig. 2. Extract from the Binary Relations Diagram

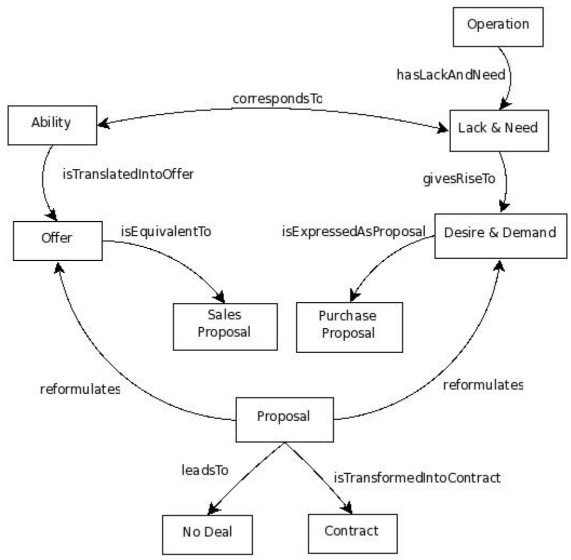

Fig. 3. Extract from the Binary Relations Diagram, describing BAT's model of transition from business prerequisites to a contract

Table 2. Extract from the Concept Dictionary

Concept Name	Relations
Business Role	participatesInInteraction, hasInternalState, isMemberOfRelationship, performsAct, hasInternalState, hasOperation
Contractual Phase	isPhaseOfInteraction, involvesExchangeOf, occursInInteractionLevel
Contract	expressesCommitment, isRealizedIn, putsNewRequirementsOnAbility
Ability	isInternalStateOf, isTranslatedIntoOffer, correspondsTo, isUsedFor

3.3 Formalisation

The formalisation phase involves the translation of the conceptual model into a formal or semi-computable model. This phase is important for two reasons: firstly, conceptual models are often ambiguous; secondly, an explicit domain model with well-understood formal semantics can be useful during the implementation phase. Description Logics (DL) [12] are generally regarded as an appropriate formalism for this task, covering both terminological knowledge (i.e. concept descriptions) and assertional knowledge (i.e. specific knowledge about individuals) about a domain. They also allow for efficient reasoning with respect to concept satisfiability, subsumption, equivalence and disjointness, as well as consistency and instance checking of the assertional knowledge. This reasoning support can be of great value, as modelling errors can be detected and resolved.

For the above reasons, the developed BAT conceptual model was transformed into a formal model expressed in the DL language *ALCN*. An extract of this formal model is provided in the following code, where terminological axioms can be seen. For example, according to the first statement, the concept “PaymentPromise” is subsumed by (i.e. subclasss of) the concept “Commitment”, and according to the second, the concepts “PaymentPromise” and “DeliveryPromise” are disjoint. The fourth statement is a necessary and sufficient definition of the concept “BusinessTransaction”, according to which a BusinessTransaction is defined as a “BusinessInteractionLayer” that consists of at most 5 “Exchanges” and that consists of some “CommitmentExchange”, some “ValueExchange” and some “AssessmentExchange”.

```
PaymentPromise ⊑ Commitment
Paymentpromise ⊓ DeliveryPromise ⊑ ⊥
CommitmentExchange ≡ Exchange ⊓ ∃hasObject.Commitment
BusinessTransaction ≡ BusinessInteractionLayer ⊓
  ≤5consistsOfExchange ⊓
  ∃consistsOfExchange.CommitmentExchange ⊓
  ∃consistsOfExchange.ValueExchange ⊓
  ∃consistsOfExchange.AssessmentExchange
```

3.4 Implementation

Utilising the conceptual and formal models discussed in Sections 3.2 and 3.3, the BAT ontology was implemented during this phase. The language chosen was OWL [13], as it has become a standard of the Semantic Web community and also because of its correspondence with DL. The ontology editor Protégé [14] was used for this task, a screenshot of which can be seen in Figure 4. Some sample code is provided below, specifying the class "PaymentPromise", and capturing the first two DL statements of Section 3.3.

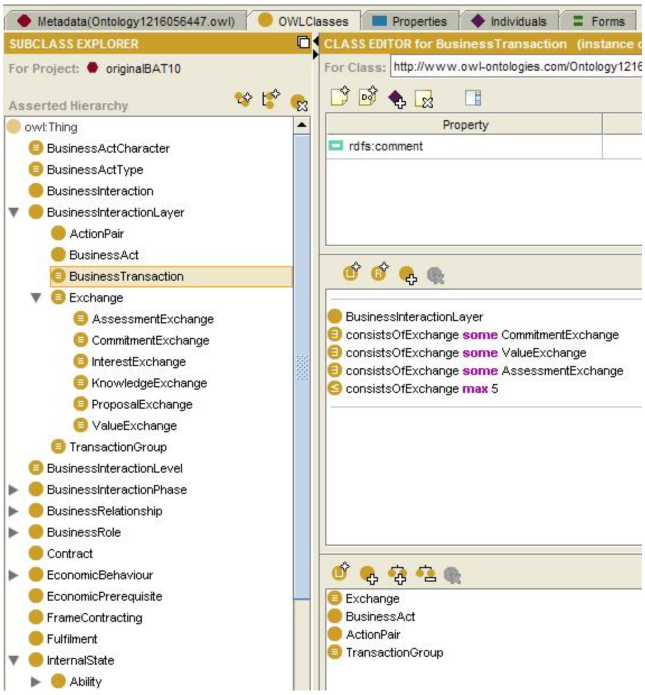

Fig. 4. Screenshot from Protégé

```
<owl:Class rdf:about="#PaymentPromise">
  <rdfs:subClassOf rdf:resource="#Commitment"/>
  <owl:disjointWith rdf:resource="#DeliveryPromise"/>
</owl:Class>
```

4 Example Use Case

As specified in Section 3.1, the BAT ontology is intended to be used firstly for facilitating the automation of business interaction, and secondly for supporting the automated analysis and improvement of business interaction. A scenario of use of the second type follows.

Imagine an agent-based virtual organisation (VO) within the travel agency sector. The VO comprises of several independent member companies, each one represented by an intelligent agent. During the VO's life cycle there is a high level of interaction between the agents; some interactions involve long-lasting relationships between VO members, while others involve the choice of new partners for collaboration. As part of their intelligent behaviour, VO members/agents analyse their business interactions with respect to the BAT ontology in order to assess and improve their performance.

Let's suppose now that agent1 analyses his business interactions at different layers (as defined at the BAT ontology), and identifies the following issue at the business transaction layer with agent2: Even though at the completion phase there is a high level of satisfaction from agent2's delivered service, the overall transaction has long duration. By following the definitions of business interaction layers from the BAT ontology, agent1 shifts the analysis to the next layer (i.e. the Exchange layer), and finds that the long duration is due to the long exchange of proposals, which corresponds to the contact establishment and proposal phase. Having identified the source of the problem, agent1 can decide on a future change of his negotiation strategy during this phase, and he can also let agent2 know about this problem. A message to agent2 suggesting to shorten the proposal phase of their future transactions could, thus, lead to a collaborative solution to the arisen problem.

But how can we assure that agent2 will understand the content of that message? If the message references the BAT ontology for the relevant used terms, agent2 will be able to look up their definition and understand their meaning, thus leading to an effective communication between the two agents. To sum up, two different uses of the BAT ontology were identified in this scenario. First, as a reference point for analysing an agent's business interactions, and second as a common vocabulary between agents to support their communication.

5 Conclusions

Recognising the significance of business interaction for intelligent business workflows, we have suggested in this paper a meta-model of business interaction. We have developed this meta-model in the form of an ontology based on appropriate business theories, and we have presented in this paper outputs throughout its life cycle. Its value is dual: Firstly, it can support the implementation of automated business interaction, and secondly it can assist the automated analysis of business interaction, thus injecting intelligence into workflows. Its well-defined semantics make it suitable for web-based, distributed and knowledge-intensive environments, such as agent-based workflows. A scenario of use in such an environment has also been presented in this paper. In the future we would like to explore the use of the BAT ontology within a supply chain simulation setting, and study issues of supply chain members' power and dominance, as determined by their business interactions. Evaluating the meta-model's completeness and coverage within such a setting, and comparing it to those of alternative approaches, such as ebXML, is another possible future avenue of research.

References

1. van Der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business process management: A survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
2. Chung, P.W.H., Cheung, L., Stader, J., Jarvis, P., Moore, J., Macintosh, A.: Knowledge-based process management–an approach to handling adaptive workflow. Knowledge-Based Systems 16(3), 149–160 (2003)
3. Wang, M., Wang, H., Xu, D.: The design of intelligent workflow monitoring with agent technology. Knowledge-Based Systems 18(6), 257–266 (2005)
4. Cardoso, J., Sheth, A.: Semantic E-Workflow Composition. Journal of Intelligent Information Systems 21(3), 191–225 (2003)
5. Goldkuhl, G.: Generic business frameworks and action modelling. In: Dignum, F., Dietz, J., Verharen, E., Weigand, H. (eds.) Communication Modeling – The Language/Action Perspective, Proceedings of the First International Workshop on Communication Modeling, Electronic Workshops in Computing, Springer, Berlin (1996)
6. Goldkuhl, G.: The six phases of business processes – business communication and the exchange of value. In: 12th Biennial ITS Conference "Beyond Convergence" (ITS 1998), Stockholm (1998)
7. Goldkuhl, G., Lind, M.: Developing e-interactions – a framework for business capabilities and exchanges. In: Proceedings of the 12th European Conference on Information Systems (ECIS 2004), Turku (2004)
8. Lind, M., Goldkuhl, G.: The constituents of business interaction–generic layered patterns. Data & Knowledge Engineering 47(3), 327–348 (2003)
9. Chen-Burger, Y.-H., Stader, J.: Formal Support for Adaptive Workflow Systems in a Distributed Environment. In: Fischer, L. (ed.) Workflow Handbook 2003, pp. 93–118. Future Strategies Inc., Florida (2003)
10. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: Spring Symposium on Ontological Engineering of AAAI, pp. 33–40. Stanford University, California (1997)
11. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer, Heidelberg (2004)
12. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook. Cambridge University Press, New York (2003)
13. OWL Web Ontology Language Overview, `http://www.w3.org/TR/owl-features/`
14. Protégé, `http://protege.stanford.edu/`