Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-3 (2015) 1857–1864

An extension proposition for the Agent-Based Language Modeling Ontology for the representation of Human-Driven Collaboration in Supply Chain Systems

Arnaud Avédissian*, Raul Valverde,**
* University of Liverpool, Laureate Online Education, UK.
**John Molson School of Business, Concordia University, Montreal.,

(e-mail: linedux@gmail.com, rvalverde@jmsb.concordia.ca)

Abstract: Supply chain systems bring a new perspective to collaboration in organizations. Human driven collaboration through communication, knowledge sharing, and cooperation, can be extended outside the organization and its partners in the supply chain. However, while human driven collaboration is part of business processes in the supply chain, they are usually depicted in models either from a high-level abstract view or implicitly included in the use of the exception related mechanisms. This creates the need for an ontology capable of representing human-driven collaboration in Supply Chain Business Processes that can enable collaboration. The Agent Lab Language (TALL) ontology was chosen as a possible solution to the proposed research problem because it is constructed around the concept of an agent and business collaborations. A Bunge-Wand-Weber ontological representation analysis was further used to evaluate the ontological completeness of the Agent Language Lab (TALL). From this analysis a set of propositions was elaborated in accordance with human-driven collaboration requirements. Following these propositions and the results of the analysis, additional constructs were proposed to the TALL ontology as a solution to the research problem

© 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

.Keywords: Ontologies, Business Models, Business Process Modeling, Supply Chain Modeling, TALL

1. INTRODUCTION

In today's supply chain, B2B integration plays an integral part to the success in enabling collaboration. Information is exchanged between partners, and business processes are integrated to enable seamless and real-time dynamic B2B interactions and communication (Minsk et al 2007)., the research undertaken as part of this paper is related to Supply chain integrated business process modeling. The representation of collaboration and in particular social collaboration in supply chain integrated business processes from the perspective of the notation used was investigated. Although the selection of a specific notation to build a model depends on many variables, its ability to represent completely and clearly the domain should first be considered. Collaboration in business processes in the supply chain can take many forms, from individuals informal collaboration to highly structured, well-defined, protocol based and agreed upon collaboration between partners. Social business processes introduce new challenges by requiring more flexibility, more agility, and an extended participation of direct and indirect stakeholders. The business process modeling notation should thus be able to reflect the continuum of collaboration forms. The ability of modeling notation to depict less structured, less defined or emerging business processes in the supply chain incarnated in collaborative supply chain software such as e-procurement has not been evaluated. The modeling of collaboration is important when designing supply chain systems as the lack of integration of these systems with enterprise systems has been highlighted as a cause of disruptions of the supply chains by several studies (Valverde and Saade 2015)(Talla and Valverde 2012). The evaluation of the modeling notation representative ability thus appeared to constitute a first step in the selection and evaluation of a particular business process supply chain modeling notation to model this extended scope of collaboration. The Agent Lab Language (TALL) was chosen for this research mainly because it is constructed around the concept of an agent and interactions in business processes that are required to model supply chain collaboration. The approach taken in this research consists in evaluating the current ability of the agent-based language modeling to represent completely and clearly supply chain collaboration. To perform such an evaluation the Bunge-Wand-Weber Ontology was used to perform a representational analysis. Following this analysis, and the discovery of deficiencies, further propositions were elaborated in order to obtain a more complete set of constructs based on the human-driven collaboration domain. These propositions were further implemented in an adaptation and extension of an agent-based language modeling using the Unified Modeling Language class and state diagram constructs. Finally, an example process was partially model in order to evaluate the improvement brought by the extension proposed.

2405-8963 © 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.06.357

2. LITERATURE REVIEW

2.1 BWW Ontology

The BWW Ontology representational analysis aims at evaluating some properties of the expressive quality of a grammar, through its ability to represent and describe the world. The BWW framework is based on an adaptation of the ontology elaborated by Bunge (1977). In philosophy, ontology is the branch of metaphysics which studies what exists, including the relations that may exist between objects, their categorization, their structure, their properties, their similarities, their states or their changes. Wand and Weber (1995) used the ontology as a tool, the representation model to analyze modeling constructs. The BWW ontology is assumed to be a clear and complete representation of the constructs required to describe the world and its phenomenon as captured by an information system model. The BWW Ontology was chosen for the following reasons:

a) The BWW Model has been developed with Information System modeling in mind. It is well-formalized and represents domain independent but information system related concepts (Valverde et al, 2011)(Valverde 2008).

b) The methodology was chosen due to a rather long history of representational analysis applied to modeling grammars, such as Entity-Relationship (Evermann and Wand, 2001), and UML (Opdhal and Henderson-Sellers, 2002) notations for instance.

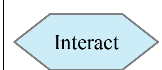
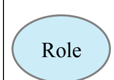
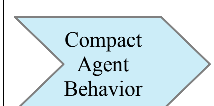
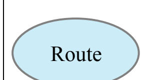
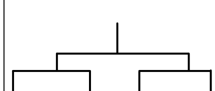
c) The well defined process defined for conducting a representational analysis (Fettke and Loos, 2003).

The BWW ontology allows evaluating the representational capability of a specific grammar through an evaluation of its ontological clarity and completeness. If a given modeling grammar is ontologically complete and clear, it should then offer a complete and faithful representation of the things, the phenomenon and their relations in the world. Such a modeling grammar would then be the best candidate to model a specific domain. The evaluation is based on a reference meta-model, which is a priori independent of domain specific constructs. A representational analysis of a modeling grammar consists in mappings of the modeling grammar constructs and the representation model in order to discover the eventual ontological deficiencies of the modeling grammar. The mapping between the representation model and the modeling grammar is executed in a bi-directional way: the grammar constructs are mapped to the representation model and vice versa. This allows for the evaluation of the modeling grammar ontological completeness (or incompleteness when there are deficiencies) and clarity (or overloading, excess and redundancy) (Valverde et al 2010).

2.1 The agent lab language (TALL)

An information systems diagram offers a highly abstracted view of process-wide behaviors through a tree structure representation of the interactions, their composition and the roles involved. Interactions are related to each other by dependency ('is part of' relation) or decomposition ('precedes' relation) (Stuit and Szirbik, 2009). Each interaction is defined at a specific level in the tree. Agents perform their behavior when an interaction is represented as a leaf. The completion of interactions follows the tree structure, a bottom-up approach from leaf to parent: a parent interaction is completed when all its children are completed. The TALL ontology is described in detail in table 1.

Table 1: Interaction Structure Diagram Elements

Symbol	Semantic
	An interaction
	A role
	An agent
	A compact agent behavior
	A route
	A role-interaction association
	An Initiator-role for a role-interaction connector.
	An agent-role association
	A interactions tree
	A human agent
	A software agent

	A synthetic agent
———	A human-software association

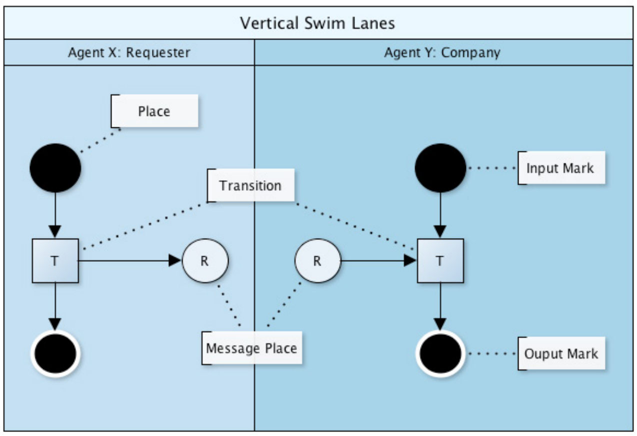
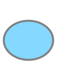
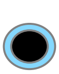
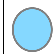


Figure 1 Agent Structure Example

An Agent Behavior diagram (figure 1) represents a local view of a behavior in an interaction from the owner's agent perspective. Note that in the diagram illustrated above, if the behavior is described from Agent X theRequester, then the Transition of Agent Y theCompany are only expected. Some references of the TALL language have thus use a cloud symbol to depict this expected behavior, an interaction belief as described by Stuit and Szirbik (2006). On the other hand, the intended behavior, which in the case is described in figure 6, is the behavior of the Agent X theRequester, might also represent a planned, a currently executing or the trace of a behavior. The agent behavior diagram elements are described in detail in table 2.

Table 2: Agent Behavior Diagram Elements

Symbol	**Meaning**
	A swimlane
	A transition: intended transition and expected transition
	A place
	An input place
	An output place
	A message place
	A token
	An Arc: incoming arc and outgoing arc.

3. RESEARCH METHODS

In order to obtain a more objective and complete approach to modeling notations, an agent-based business process modeling language was identified and selected in order to perform a BWW ontology ontological analysis. The TALL language was chosen because it is focusing on agent behaviors and interactions in business processes. For instance, employees interact with the company's partners or with other company's employee; employees also interact with each other in order to execute processes that are part of the supply chain management. These interactions can follow a protocol, either pre-agreed upon in the case of a partnership, or dependent on a social context in the case of human informal and formal interactions. A representational analysis of the TALL language was thus conducted, and began with the collection, enumeration and classification of the diagrams and constructs of the notation as exposed across the TALL research papers.

Table 3 TALL Representation Mapping

BWW constructs	**TALL constructs**	**Description**
Thing	Agent can represent physical active or passive thing. A swimlane represents an instance of an agent. A synthetic agent can represent a composite thing.	An agent can represent a concrete thing either passive (a software agent can represent a passive thing) (Stuit and Szirbik, 2009) or active.
Property	Agent type, name, swimlane instance name and role.	An agent name and type are properties in general, and its instance name, in the case of a swimlane in an Agent Behavior diagram, is a property in particular. It is mentioned that a role adds properties to an agent and thus can represent a binding property.
Class	Agent	An agent can represent a class of agents possessing a common property. For instance software or human agent.
Kind	Agent	An agent can represent a kind as it can be an agent group. An agent group contains agents with more than one common properties.
State	Input place, Output place, Place and	Input place represents the initial state of a behavior. A place represents a state between transitions and at a given instant

	token	is marked with a token to represent the behavior state. An output place represents the ending state of a behavior.
Conceivable State Space	N/A	
State Law	Place to transition to place	The place transitions to one or more places indicates that only the successive states for which an outgoing arc is available can be lawful.
Lawful State Space	N/A	The capacity of a place is not explicitly depicted in the notation description found.
Stable state	Output place	The output place mark the end state of the behavior and can not be further changed by internal events (transition or message).
Stability condition	The G function associated with an incoming arc.	Although without any graphical symbol, the function is defined as part of the formal definition of Behavior Net. It defines a boolean expression for incoming arcs (Meyer and Szirbik, 2007)
Unstable state	Place, Input Place and Token	As with Petri Net, the input place, place and token depicts states that could be changed upon internal (like transition) or external events (like message).
History	N/A	
Event	Transition, Message Place, Input and Output Place.	Transition represents a bridge between two states while a message place represents an external event that local agent receives. An output place also represent the behavior's termination event.
Conceivable Event Space	N/A	Although the message place has a defined data type, agents interactions through their behavior does not ensure that the agents will send the correct data type in a message. Further pre-interaction, on-the-fly or mediator alignment might be required.
Lawful Event Space	N/A	
External Event	Message Place	Within an agent behavior representation a message place depicts an external event.
Internal Event	Transition	A transition can be considered an event because it represents the transition from one state to another.
Well-defined Event	Transition, Output place, Input place	A transition leads to a new state which can be predicted according to the transition outgoing arcs associations. An output place being the end state, the next state is always the end of the behavior.
Poorly-defined Event	Message Place	Upon the reception of a message, the next state of the behavior is hard to define. The message content might not be the expected one and the agent behavior might need alignment.
Transformatio n	Transition, interaction	As in Petri Net, a transition change the state of the behavior. An interaction being the execution and the result of two or more agent behaviors, the process will change from one state to another.
Lawful transformatio n	Route [SEQ, PAR, XOR]	The route depicts the lawful organization of interactions, either sequential, parallel or exclusive choice (XOR). Additional decision rules can be specified on the route (SEQd, PARd, and XORd). Although a behavior is agent dependent, an interaction follows a parent-child and routing relation.
Acts on	Arc, Role	A role acts on an interaction by initiating it for instance. An arc depicts a relation between states.
Coupling	Agent association, role-interaction association and agent-role association.	An agent association, such as a software-human agents association depicts the coupling of both agents. A role-interaction association also depicts the influence of a role on an interaction. An agent-role association also depicts the particular influence of an agent instance in a given role.
System	Interaction	An interaction representing the action of behaviors on each other, coupling exist between any two agents engaged in the interaction.
System Composition	Interaction tree	The interaction tree depicts the composition of the business process, especially through the interactions parent-child relations.
System Environment	N/A	
System Structure	N/A	
Subsystem	Interaction tree	An interaction tree represent the parent-child relations of interactions. A child interaction can be considered a subsystem of its parent interaction.
System Decompositio n	Interaction tree and swimlane	The interaction tree represent the decomposition of parent interactions into child interactions. A swimlane represent an agent and contains the decomposition of its specific behavior.
Level Structure	Hierarchical tree of interactions and roles	The tree representation of interactions and roles can represent a hierarchical structure of both roles and interactions.
Process	Interaction	An interaction can represent an ordered sequence of behaviors if aligned with a protocol. The protocol defines how behavior should be aligned before an interaction.

Table 3 includes the BWW ontology mapping to TALL constructs. The mapping revealed few issues: The BWW constructs Conceivable State Space, Conceivable Event Space, Lawful Event Space, Lawful State Space, History, System environment and system structure have no corresponding constructs in the TALL grammar. The BWW constructs State, and in particular unstable state, Event, Well-defined Event, Coupling, Acts on, System Decomposition and Level Structure are represented by more than one element in the TALL grammar. The BWW Thing and transformation construct is also represented by both agent and swimlane.

Table 4: TALL Interpretation Mapping

TALL Constructs	BWW constructs
1) Interaction	An interaction is composed of at least two agents each exhibiting a behavior to fulfill a role in an exchange and thus depicts a transformation.
2) Role	A binding property of an agent with an interaction.

3) Agent	An Agent can represent thing in the world, even passive thing (Stuit and Szirbik, 2009). Note that an Agent Prototype diagram also appears to model a thing as a physical entity (Stuit, Szirbik and Meyer, 2009). A synthetic agent can also represent a composite thing because it can not only inherit properties of its parts but have its own properties and behaviors. Agent also represents a class, like human, software or synthetic agent with a single common property. An agent can also represent an Agent group, in which case agents posses more than one common properties and thus can represent a kind.
4) Agent Association	An association between two agents, a human and a software agents for instance, represents coupling between the agents.
5) Compact Agent Behavior	Compact view of a behavior which indicates a reference to an intended or already manifested local behavior. This construct appears to be in excess.
6) Route	A lawful transformation as it indicates which interactions are allowed as part of the parent interaction. Furthermore, route can have additional decision rules.
7)Role-Interaction association	A role affects an interaction, especially in the case of a mediator role played by an agent to allow for an alignment of the behavior (Stuit, Szirbik and de Snoo, 2007) . The role-interaction association can thus represent coupling.
8) Initiator Role-Interaction association	Represents a role which initiates an interaction. Initiator depicts an additional property of the role in the interaction.
9) Agent-Role Association	A role affects the behavior of an agent, possibly adding properties to the agent. The association can represent coupling.
10) Interaction Tree	A system with its composition and decomposition and level structure. The interaction tree describes a system through its interactions. The interactions are further decomposed and represented hierarchically with a dependence of the parent on the child interactions.
11) Role tree	Represents in a hierarchical way the relation between roles. It can be mapped to a level structure.
12) Swimlane	Represents a local behavior which is a subsystem of the interaction system.
13) Transition	Represents a transformation, the token can be modified by the transition and lead to a new state of the behavior.
14) Place	In a synthetic agent behavior representation, a place might represent concrete objects. For instance, a virtual community agent modeled as a synthetic agent could have a place representing customers, a place for company's employee.
15) Input Place	Represents an event, the first event that initiate the behavior. It also represents the initial state of the behavior which is unstable because it will change with transitions.
16) Output Place	Represents the ending event of the behavior, a well-defined event because we know what is the state of the behavior beyond this event.
17) Message Place	Is an event which from the receiving behavior point of view represents an external event. This event is also poorly-defined because the following state of the receiving behavior is only expected and represents a belief from the sender's perspective.
18) Token	A token represents the marking of the behavior state. It represents a state, as in the Petri Net notation. The state represented is also unstable as it can change after a transition. Token is apparently assumed to be always equal to one.
19) Arc	Arc comes from the Petri Net notation and thus represents the BWW acts on construct because it depicts a relation between things.

Table 4 includes the TALL ontology mapping to BWW constructs. An agent maps to several BWW entities, such as Thing, Class and Kind. A transition also maps to more than one BWW entity, including a transformation, and a well-defined internal event. An interaction tree can also be interpreted as a BWW System, System Decomposition, System Composition and Level Structure.

Only the Compact Agent Behavior was found to have no mapping construct in the BWW ontology. From our understanding of the notation, the association of an Agent with a Role in an Interaction Structure Diagram would have been sufficient to indicate the agent's behavior.

The UML grammar was selected as a grammar that could be used to complement the TALL ontology in order to compensate for ontological deficiencies. The UML grammar is strong ontologically speaking, although there are several BWW constructs that cannot find representation in any diagrams: system structure, subsystem, lawful event space, acts on, poorly defined event (Valverde & Toleman 2007).

4. RESULTS AND FINDINGS

The first proposition is to model artifact (passive thing) and agent (active thing) separately. In this way, an artifact representing the goal and tangible outcome of the collaboration can also be shared among agents. In addition, a shared artifact is constrained by rules. Agent behaviors on the other hand could also follow rules, but more in the form of policies. Moreover, a shared artifact can be composed of other shared artifacts, allowing the representation of a composite artifact. An artifact structure can thus be represented in order to model the relations existing between the whole and its parts. In order to express the required rules that could apply to an artifact, the UML state machine constructs can be used, thus depicting the conceivable state and event space with the additional representation of the rules applying before the transition to another state. For instance, a rule reference could be applied on a transition. In the upper part of such a diagram, the properties of the artifact, related to the applicable rules could be enumerated. An artifact can then be associated with an objective, or a child interaction.

The second proposition is to model the context of an interaction. This context should contain the history of the artifact associated, but also past interactions associated with this particular objective. Because agent can build knowledge from memory, the memory mechanism described in collaboration can be shared by agents through the context.

Moreover, a context should be shareable between interactions and interactions instances. A synchronization and marking of context in parallel interactions could also allow depicting the agent alignment.

The required auditing of processes and capture of interactions, actions on or toward the realization of the artifact should be available across the whole process. Although most workflow management system has such a log feature, the context proposed should explicitly represent the audit of the current artifact and interaction.

This lead to a distinction between the current workspace and the context, the workspace represents the shared goal artifact and shared objective artifacts as they are transformed or realized by agent's interactions. The shared artifacts organization in the workspace can be represented by a UML class diagram representing shared artifact relations.

In order to evaluate the pertinence and applicability of the propositions, a collaborative supply chain process was partially modeled. The collaborative process described here is an e-procurement content localization quotation process .

The Quotation process can be rather complex and is usually determining not only the cost of the project, but also the specific sequence of steps that will be used for this particular project and which could potentially benefit to all other (con-)current and future projects.

This process involve three categories of stakeholders, the requesting organization and its partners, commonly grouped into a client role, the organization in charge for the quotation, denominated the localization service provider (LSP), and the vendor role which group both linguistic, engineering, and domain experts.

The main goal of the process is to provide in a timely manner an accurate estimation of the cost and duration of the project, with all the technical and human resource unknowns possibly identified, evaluated and planned. At a high level, the process can be described in the following terms: the source content (provided by the client in the original locale) is received and analyzed with the help of localization tools to produce a quantitative analysis (word count). According to the price negotiated with the vendors, and an estimation of the duration according to work average, a priced quotation is delivered specifying the estimated duration of the localization work requested. However, in practice, during this process much expertise is usually required, technical issues usually appear and a specific knowledge of the content and its context is usually built. This might be due to the complexity of the content, which can include textual, visual or audio elements to be localized. The source content can also appear in different context, for instance in the course management application, or the course activity framework or the course content itself. Here the potential interactions are depicted with a dashed line. The roles are mostly generic but from the following interaction diagram, different type of agents possibly involved can be identified.

The further roles identified in the interactions and the agents fulfilling these roles can be depicted as illustrated in figure 5. The developer role can represent for instance the role of the content developer in the extraction of content, or the role of the content filter developer in the filtering interaction, or the role of the segmentation rules developer in the segmentation interaction.

In order to be able to only localize relevant content, the content must be filtered out of comments or variables. Then, a set of segmentation rules, inherited from the natural source language, and the addition made by the content designer must be specified to obtain coherent units of content with a high potential of repetition and thus reusability through the Translation Memory (TM). For instance a filtered unit of text, such as an instruction might contain more than one sentence and additional variable which must be properly segmented as a whole to be reusable across courses.

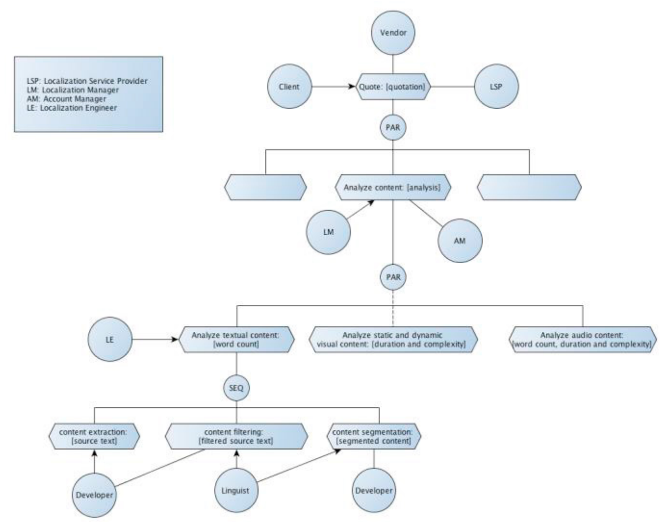


Figure 2 Overall Partial View

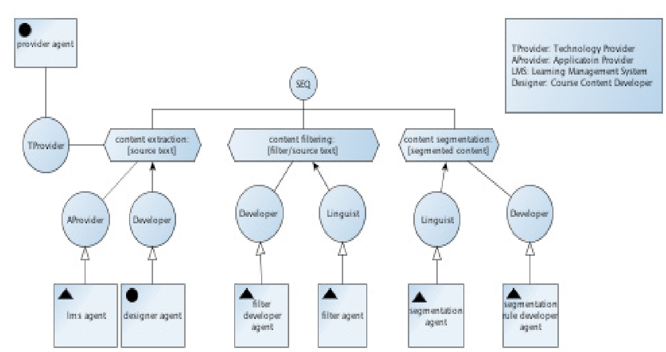


Figure 3 Agent in Interaction

Although the agents fulfilling the linguist and developer roles are usually depicted by individual, in practice, the linguist and developer are interacting with a service, such as the filtering or segmentation service. Furthermore, not only one linguist or developers work on a specific filter. A community of developers might for instance work on some specific application filter. Because content must also be indexed and available to search through keywords, the developer community of content indexer are also providing help in the filtering and segmenting sub processes.

The possible composition of the synthetic agent filter developer agent is represented in Figure 4. The Filter Integrator is a developer in charge of the filter engine. The filter Engine is representing the manifestation of the filter when applied to the content. The Community filter Developer represents a community of developer working on creating and improving digital content extraction filter. This community represents for instance an individual developer providing contributions on a converter application on a community question and answer website, or a developer contributing to a converter piece of code in a public repository.

An artifact diagram can help to represent the relation existing between the goal artifact and the specific objective artifacts. In the diagram depicted in figures 5 and 6, the artifact produced following the interaction is represented in between brackets. For instance, “content filtering:[filtered source text]” means that the interaction's goal is to filter the content and this goal tangible artifact is the filtered source text form. The figure 5 depicts with a simplified UML class diagram the relations between the artifacts.

A specific artifact can be modeled as in figure5, including its properties and states, as well as rules governing state transitions. This view allows representing the conceivable and lawful states for a given instance of the Scope type resulting from the interactions shown in figure 7.

The context construct is shown as a round rectangle with dashed line in the following intended behaviors between a community filter developer, a filter integrator and a linguist. As shown in Figure 7, a context can be shared and does not have to be unique. For instance the shared context between the filter integrator and the community filter developer is the need to filter text content (or audio, or video) from the content provided. This context is including past, present and future dimension because the filter might have been built in the past, the search might be current and the usage and customization might happen in the future. Note that the behaviors represented are only intended from the filter integrator point of view and further alignments might occur during execution. The shared artifacts filter and source content are also represented in the interactions workspace, symbolized by the frame including swimlanes.

Following the partial modeling of this particular quotation process, the following conclusions can be drawn. First, human-driven collaboration and the interactions it generates are govern and executed toward the achievement of particular objectives which participate to the achievement of a more general goal. The individual objectives might not be all known prior to the collaboration execution and might change or further objectives defined in response to internal and external event which affect the state and thus the progress of collaboration toward its goal. Secondly, the monitoring of the collaboration shared artifacts relations and status appeared of prime importance in the collective awareness of the progress of collaboration, as well as in the collective definition, evaluation and realization of intermediate objective. Thirdly, the collaborative and collective memory is playing an important role in the adaptation of the individual objectives and resulting interactions to a particular context in order to achieve the collaboration goal. These observations are however partially represented in human-driven collaboration models, and partially due to the ontological deficit observed following the representational analysis. Although processes are represented as a sequence developing in a temporal dimension toward a future, it appeared that a new dimension, transversal to the development of the activities can be added. This dimension not only represent past experience and practice accumulation, but also the current and concurrent context of a particular activity in the sequence.

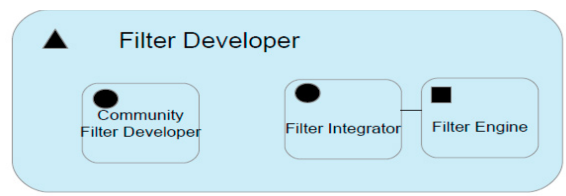


Figure 4 Filter Developer Agent

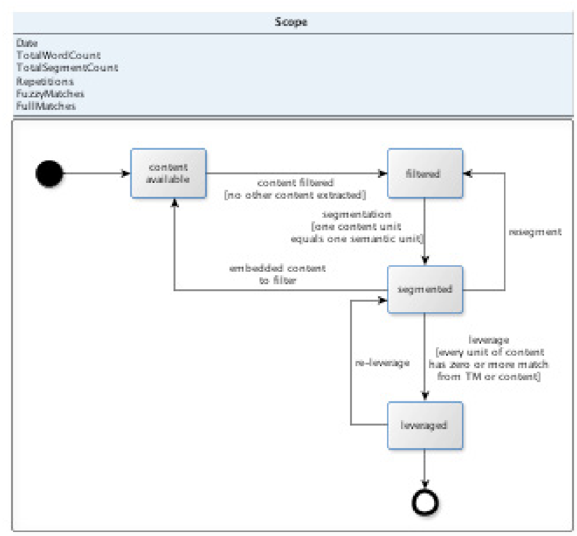


Figure 5 Artifact Type diagram

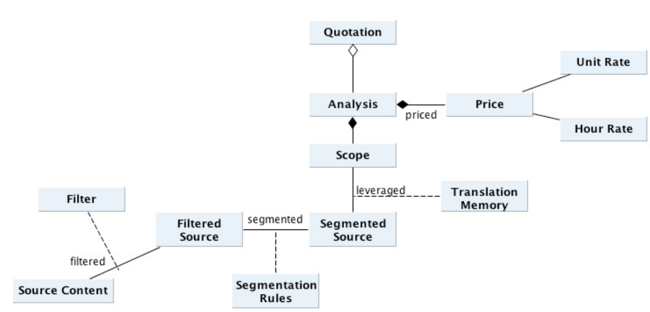


Figure 6 Artifact Relations in a shared workspace

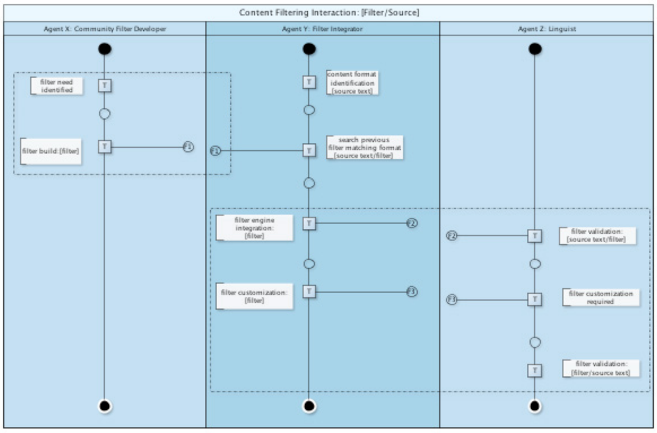


Figure 7 Content Filtering Interaction Intended Behaviors

5. CONCLUSIONS

The BWW ontology brought a piece of the answer to this research by providing a larger reference model, with the ontology, and also a methodology, in the form of the representational analysis. The model should reflect the evolution of the process from what it was to what it should be, while at the same time being able to represent all aspects of reality with fidelity. Consequently the expectations on the capability of the notation used are reflecting the model function. The notation must allow the complete and precise capture, simulation and description of the business process with various levels of details and from the multiple perspectives represented among the stakeholders. When representing collaboration, the same issue can be found. At a higher level, the view is focused on the essential traits of the business process, and defines the goal rather than the particular objectives. At a lower level, the details of the activities to be conducted, the resources to be used and the methods are described. At the lowest level, every atomic activity of a participant, including communication, modification of an artifact, or just notification of an activity should be considered in order to reconstitute or execute the various levels alignment.

The extension proposed in this research would need to be part of a new set of specification, a grammar describing a higher level of abstraction. This grammar and its construct could then be evaluated using the BWW Ontology representational analysis method. However, in order to obtain such specification, a more detailed and formalized reference model of collaboration should be elaborated. This could be part of future research that could add to the proposed research results in this article.

REFERENCES

Bunge,M. (1977), *Treatise on basic philosophy. III. Ontology the furniture of the world*, Dordrecht, Reidel

Evermann,J. and Wand,Y. (2001), 'Towards Ontologically Based Semantics for UML Constructs', in 20th International Conference on Conceptual Modeling (ER), Yokohama.

Fettke, p. and Loos, p. (2003), 'Ontological evaluation of reference models using the bunge-wand-weber model', proceedings of the 2003 americas conference on information systems, august, pp.4-6

Minsk, G. C., Poh, L. S., Wei, H., & Siew, T. P. (2007, September). Web 2.0 concepts and technologies for dynamic B2B integration. In *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on* (pp. 315-321). IEEE.

Opdahl, A.L. and Henderson-Sellers,B. (2002), 'Ontological evaluation of the UML using the Bunge-Wand-Weber Model', Software and System Modeling, Vol.1, Issue 1, pp.43-67.

Stuit, M., & Szirbik, N.B., (2006), 'Interaction Beliefs: a Way to Model and Execute Complex and Changing Business Processes with Agents' in Proceedings ESAW06, pp.169-184.

Stuit, M., & Szirbik, N. B. (2009). 'Towards agent-based modeling and verification of collaborative business processes: An approach centered on interactions and behaviors. International journal of cooperative information systems, 18(03n04), 423-479.

Talla M. & Valverde R. (2012) , Data oriented and Process oriented Strategies for Legacy Information Systems Reengineering, ACEEE International Journal on Information Technology, 2 (1), pp. 47-61.

Valverde, R. & Saadé, R. (2015) The Effect of E-Supply Chain Management Systems in the North American Electronic Manufacturing Services Industry, Journal of Theoretical and Applied Electronic Commerce Research, 10(1), pp 79-98.

Valverde, R., Toleman, M. & Cater-Steel, A.(2010) Design science: a case study in information systems re-engineering. In: Hunter, M. Gordon, (ed.) Strategic information systems: concepts, methodologies, tools, and applications. Information Science Reference (IGI Global), Hershey PA, USA, pp. 490-503.

Valverde, R., & Toleman, M. (2007). Ontological evaluation of business models: comparing traditional and component-based paradigms in information systems re-engineering. Ontologies, 49-77.

Valverde, R. (2008). The ontological evaluation of the requirements model when shifting from a traditional to a component-based paradigm in information systems re-engineering (Doctoral dissertation, University of Southern Queeensland).

Valverde, R., Toleman, M., & Cater-Steel, A. (2011). A method for comparing traditional and component-based models in information systems re-engineering. Information Systems and e-Business Management, 9(1), 89-107.

Wand,Y. and Weber, R. (1995), 'On the deep structure of information system', Vol.5, Issue 3, Wiley Online Library, pp.203-223