Dynamic Configuration and Management of e-Supply Chains Based on Internet Public Registries Visited by Clusters of Software Agents

Konrad Fuks, Arkadiusz Kawa, and Waldemar Wieczerzycki

Poznan University of Economics,
al. Niepodleglosci 10, 60-967 Poznan, Poland
{konrad.fuks, arkadiusz.kawa, w.wieczerzycki}@ae.poznan.pl

Abstract. There are three main contributions of the work presented in this paper. First, a particular approach to develop and manage multi-agent environments is presented. It is based on so called agent clusters which group agents that mutually cooperate to perform a particular mission, e.g. to build and manage supply chains, to negotiate details of order. Second contribution is the presentation of public registry model. Authors have widened RosettaNet approach to public registries. Third, the integration of agent cluster model and public registry model is presented. New approach shows how agent technology can automate information exchange process within public registry.

Keywords: Agent clusters, e-Supply Chains, RosettaNet Document Exchange Standards, e-Business Registries.

1 Introduction

Globalization entails necessity of easily configured and reconfigured supply chains. Enterprises which are subjects of supply chains must have well-defined access to the information generated by their suppliers and customers. Additionally, one has to emphasize that information flow must assemble all links of the supply chain, not only the closest partners. Information scope must start from suppliers' suppliers and end on customers' customers. This kind of information flow allows for fast and precise reaction of whole supply chain in changing environment conditions. Very important role in this scenario plays standardization of exchanged information within supply chain. Applied standard must be elastic and interoperable to meet needs of all trading partners. Large companies with highly developed back-end systems as well as small and medium enterprises should have possibility to utilize and generate information within their supply chains. This very important need for information can be satisfied by combining agent technology with e-business registries [10].

Agent technology is one of the most prominent and attractive technologies in computer science at the beginning of the new millennium [2]. Using software agents, especially to support e-commerce automation is a very promising direction. Additionally, nowadays business partnership is very often created dynamically and

V. Mařík, V. Vyatkin, A.W. Colombo (Eds.): HoloMAS 2007, LNAI 4659, pp. 281–292, 2007.

maintained only for the required period of time such as a single transaction. For this reason agent technology can be useful for e-supply chain configuration, reconfiguration and management based on e-markets [3, 8].

To conduct B2B transactions properly agents must have access to well defined and structured business information. Therefore e-business registries can solve this problem. E-business registries – electronic platforms build for information exchange between supply chain's or supply network's enterprises. This kind of registries should be based on well-defined uniform information exchange standard. Therefore e-business registries secure coherent communication between trading partners. Nowadays one of the most popular meta-language for building e-business registries is XML (eXtensible Markup Language) and its descent standards. Usage of XML ensures system interoperability of the e-business registry and its openness for new trading partners (single enterprises or whole supply chains).

The paper is a continuation of work presented at the last HoloMas Conference [12]. Its structure is as follows.

In Section 2 multi-agent environment supporting collaboration of agents is briefly reminded. First, basic concepts of the approach are given. Next management problems concerning agents collaboration are described. The approach discussed in this section is inspired by selected mechanisms of the database technology which plays the role of a core technology in the development of business information systems [12].

Section 3 presents a new model of *RosettaNet* based public registries. In the beginning premises and environment of the model are described. In the second part of this section general interaction model and registry information model are presented.

In Section 4 integration of models presented in the above sections is given. Authors start from presentation of integration premises and agent types in the public registries. Section ends with exemplification of agent usage within public registries.

Section 5 concludes the paper.

2 Clusters of Collaborating Agents

We start with the presentation of mechanisms of multi-agent environment which support and facilitate cooperation among agents visiting the same environment. It is worth to emphasize that agent technology background is fully compatible with FIPA specification [4, 5, 6].

The main idea of the proposed multi-agent environment design is the use of so called agent clusters [12]. Before we introduce agent clusters, we have to define some basic concepts. Web servers built in agent technology (e.g. e-markets, e-auctions) can be accessed in practice by an arbitrary number of agents (e.g. representing many businessmen) which work independently, or collaborate with other agents, e.g. to dynamically re-configure electronic supply chains. Depending on whether agents collaborate or not, and how tight is their collaboration, we distinguish two levels of agents grouping: *constellations* and *clusters*. A constellation *Ci* groups agents which aim to achieve the common goal, typically to buy or sell products or services. Agents belonging to the same constellation can communicate with each other and be informed about progress of business efforts by the use of typical communication channels. Constellations are logically independent, i.e. an agent belonging to a single constellation is not influenced by the evolution of other constellations. It is possible for a single agent to contribute simultaneously in many constellations, thus the intersection between two constellations need not be empty. In this case, however, actions performed in the scope of one constellation are logically independent from actions performed in other constellations.

Agents belonging to the same constellation can collaborate tightly or loosely, depending on whether they represent the same business party, e.g. enterprise (possibly extended by business partners), corporation, or they represent different business parties, aiming to find new business opportunities. Tightly collaborating agents are grouped into the same agent cluster AC_i. Thus, a cluster is an agents subset of a corresponding constellation, with the restriction that a single agent A_i belongs in the scope of a single constellation exactly to one cluster, in particular, to a single-agent cluster. Of course, if the agent is included in many constellations, say n, then it belongs to n clusters.

A *semi-transaction* is a flat, ordered set of operations on the same business data repository (in particular - database operations) performed by agents of the same cluster, which is atomic, consistent and durable. In other words, a semi-transaction is the only unit of communication between a virtual agent representing component agents of a single cluster, and the business data management system (or DBMS).

Formally, a semi- transaction is defined as a triple:

$$ST = (Tid, Cid, ACid) \quad (1)$$

where *Tid* is a transaction identifier, *Cid* is an identifier of the encompassing constellation, and *ACid* is an identifier of the cluster to which *ST* is assigned.

Now we focus on operations which can be performed on semi-transactions being contexts of agent clusters activity.

Every semi-transaction is started implicitly by *initialize*(T_i) operation, which is performed automatically by the agent environment on the very beginning of respective cluster activity, i.e. after first data operation is requested by one of cluster members. This cluster member is called a *transaction leader*. *initialize*(T_i) is also triggered automatically, directly after one of cluster members performs explicit *commit*(T_i) operation, or implicit *auto-commit*(T_i) operation. All consecutive transactions of the same cluster are executed in a serial order.

After a semi-transaction is initialized by the transaction leader, other cluster members can enter it at any moment of the transaction execution, by the use of explicit *connect*(T_i) operation, which is performed in asynchronous manner. Once connected to the transaction, any member of a cluster can perform *disconnect*(T_i) operation, provided there is still at least one agent assigned to this transaction. *disconnect*(T_i) operation brakes the link between transaction T_i and the agent, which can next:

1. close its session,
2. suspend its operations for a particular time moment and re-connect to the same cluster later,
3. wait until transaction commits and connect to a next semi-transaction of the same cluster,
4. continue to work in different cluster in the scope of another semi-transaction, provided the agent belongs to more than one cluster.

In cases: 1, 2 and 4, *disconnect(T_i)* operation plays the role of *sub-commit* operation, which means that the respective agent intends to commit its own operations, and leaves the final decision whether to commit or not the semi-transaction to its recent collaborators (agents).

Operations introduced up till now concern a single semi-transaction. Next two operations: *merge(T_i)* and *split()* are special, since they concern two semi-transactions. They are extremely important for supporting typical business activities (e.g. on e-markets). *Semi-transaction* T_j can merge into transaction T_i by the use of *merge(T_i)* operation, provided the members of a cluster assigned to T_i allow for it. After this operation, transaction T_j is logically removed from the agents environment, i.e. operation *abort(T_j)* is automatically triggered by the environment, and all T_j operations are logically re-done by transaction T_i. These actions are only logical, since in fact operations of T_j are just added to the list of T_i operations, and T_i continues its execution, however, the number of agents assigned to it is now increased. It means that until the end of T_i execution, the agents cluster assigned previously to T_j is merged into the cluster assigned to T_i. Of course, *merge(T_i)* operation is only allowed in the scope of the same constellation. Obviously *merge(T_i)* can be useful when an access conflict between two clusters of the same constellation arises. But in terms of business activities this operation allows to couple representatives of business parties which decide to cooperate (e.g. negotiate cooperation conditions, browse proposals of business contracts and other documents, or even electronically sign them finally). Its use is more detailed further in this section.

Similarly to *merge* operation, *split()* operation can be used in order to avoid access conflicts. *split()* operation causes that a single semi-transaction T_i is split into two transactions: T_i and T_j. After *split()* operation, a subset of cluster members, originally assigned to T_i, is re-assigned to newly created transaction T_j. Also all operations performed by re-assigned agents are logically removed from transaction T_i and redone in transaction T_j directly after its creation. Notice, that in practice this operation can be useful if cooperation of representatives of two business parties for some reasons fails, and they decide for example to autonomously analyze the e-market to look for new parties offering them better cooperation conditions.

Contrarily to *merge* operation which is always feasible, provided members of the other cluster allow it, *split* operation can be done only in particular contexts. Speaking very briefly, a semi-transaction can be split if two sub-clusters, which intend to separate their further actions, have operated on disjoint subsets of data, before *split* operation is requested. If the intersection between the data accessed is not empty, *split* operation is still possible, provided the data have been accessed by the two sub-clusters in a compatible mode (in a classical meaning).

Finally, there are two typical operations on transactions: *commit* and *abort* which are performed in the classical manner (according to the database technology).

3 Public Registry Model (*PRM*)

Premises of the Model

In this section we propose a new approach to public registries. In order to better illustrate the model and to relate it to practical aspects (existing solutions) authors utilized as backbone of the public registry model *RosettaNet* standards.

Themes of this paper don't bring up detailed description of *RosettaNet* standards[2]. We have focused our work on public registry model presentation and interactions between trading partner within public registry.

Main value of the presented model is addressed to SMEs because they mostly left behind well-defined information exchange in the supply networks. SMEs comprise over 95% of enterprises and two thirds of private sector employment in OECD countries [9]. This statistic shows how important is active participation of those enterprises in building supply chains and/or supply networks.

Environment of the Model

1. *RosettaNet* standards do not define registry building specification. Public registries can be built with UDDI or ebXML standards.
2. All trading partners have the same level of public registry control and possibilities of its components usage.
3. Public registry is maintained by Public Registry Host (PRH) with support of *RosettaNet* as a standard body.
4. Public Registry data is well-defined for specific market, thanks to PRH knowledge about the market.
5. PRH should be *RosettaNet* member due to good knowledge about standards.
6. Back-end system in trading partner's IT infrastructure is not obligatory. Information can be processed in machine or human readable format.
7. Trading partners can join PRM alone to search market for new suppliers and clients or adapt PRM to their own supply chains.

General Interaction Model

Interaction between entities starts after successful public registry modeling and implementation by Public Registry Host (PRH). PRH decides which registry standard to use (ebXML, UDDI, other). PRH deploys all needed data (potential business scenarios and connected with them documents, messages and business interaction models (*BS/D/M*[3])) into Public Registry (PR). All deployed data should result from *RosettaNet* standards. Also before deployment all data must be validated with *RosettaNet* centrally maintained Validation Tool. When PR is ready to use Trading Partners (TPs) can create their own profiles within registry. All needed data structure for TPs profiles is specified by PRH on registry modeling stage. Well defined and structured profiles allow TPs to search for possible business contacts [7]. TPs profiles serve also as a base for authentication process in all further business information exchanges. Trading Partners communicate with each other using *RosettaNet* protocols and documents. For wide adoption of PRM authors assumed that access to Public Registry is granted for all TPs (new and well familiar with *RosettaNet* standards).

---

[2] In order to better understanding of *RosettaNet* standards authors encourage readers to acquaint with information on *RosettaNet* web portal – www.rosettanet.org.

[3] *BS/D/M* – all possible business scenarios and related with them documents, messages, business interaction models published by PRH. All these entities are based on *RosettaNet* standards. Documents and messages are stored in XML files and related XML Schemas. Business models are UML diagrams.

Additionally within Public Registry can be stored statistic data about business scenarios utilization among TPs. Such data can be very helpful in PR improvements and developments. RosettaNet may also use this statistics as base for fundamental standards development and upgrades.

Registry Information Model

Presented information model relies on modular approach for BS/D/M creation. We propose two level classification of BS/D/M stored in the PR:

- General Level – PRH creates BS/D/M according to possessed knowledge about the market and *RosettaNet* standards. In the PR should be stored all possible variants of BS/D/M divided into well-described modules. Nature of each module relies on differences between possible business scenarios within specific market domain.
- Trading Partner Level – level of individual TPs registries. Smaller, constrained bases of BS/D/M General Level modules or self – created by *RosettaNet* familiar TPs BS/D/M according to *RosettaNet* standards. Additionally TPs can add their own constrains to General Level BS/D/M but those new modules must be validated by PRH. Validation is conducted with centrally maintained by *RosettaNet* Validation Tool [TPIR – PIP].

We have distinguished three possible scenarios of good PRM utilization. First two scenarios refer mostly to SME sector. Third scenario focuses on *RosettaNet* familiar enterprises, involvement of those entities in PRM adoption is very important because they create foundations of *RosettaNet* standard utilization.

1. In the first scenario TP is new to *RosettaNet* standards and does not have proper IT infrastructure (back – end systems and/or human staff) for final BS/D/M preparation. TP operates mainly on General Level BS/D/M and information is manually processed within its company by using RosettaNet complaisant forms. PRH has here very important role of advisor and supervisor in TP's B2B transactions. On the one hand PRH advises in choosing proper General Level module / modules of BS/D/M, on the other hand it supervises first steps of TP's PR transactions.
2. Second scenario is addressed to those SMEs which are new to *RosettaNet* standards and have well trained IT stuff. Back – end system is not required but its possession can significantly improve business information processing. TPs without back – end system manually process information using RosettaNet complaisant forms. TPs can work on General Level BS/D/M, constrain or expand them accordingly to their business needs. All changes in BS/D/M must be validated before usage. Lower level of PRH support is required in this scenario.
3. Third scenario entities are *RosettaNet* familiar enterprises. They can either utilize all needed General Level modules of BS/D/M from PR or create their own BS/D/M due to their *RosettaNet* standard knowledge. Of course all acquired form General Level BS/D/Ms can be additionally constrained or expanded, but before final utilization they must be validated. PRH supports TPs with comprehensive information about General Level modules.

4 Application of Agent Technology to Public Registry Model

In this section we propose the adaptation of agent technology, especially agent clusters into Public Registry Model (PRM).

The approach stipulates that information about Trading Partners included in PRM can be also successfully applied in creating dynamic and flexible supply chains. Hence, this new proposal expands the present opportunities offered by *RosettaNet* standards.

Thanks to embedding agent technology in PRM, automation of comparing, reorganizing and verifying BS/D/M scenarios is greatly facilitated.

Premises of the Approach

- There is a need to automate some of the operations in PRM, such as: checking, comparing and reorganizing BS/D/M scenarios.
- Enterprises have limited opportunities to automatically search, select and choose future TPs.
- Supply chains are characterized by the integration of activities, operations and functions carried out at different supply stages scattered in various corners of the world. At present supply chains compete with one another.
- There is a need to create temporal and dynamic supply chains aimed at executing a single transaction.
- The aim of supply chain configuration problem is to find a feasible configuration with which the supply chain can achieve a high level of performance [1].

Environment of the New Approach

- Public Registry serves as a base for searching new TPs and creating stable, temporal and dynamic supply chains.
- PRM is expanded with multi-agent environment and each TP is represented by specific group of software agents.
- Agents are divided into two constellations: PRH constellation and TPs constellation.
- Access to Public Registry is granted for all TPs' software agents, but level of privileges of each agent can differ.
- TPs' agents collaborate with each other and consequently add value to established supply chain.

Types of Agents Used in PRM

Two types of software agents representing both PRH and TPs can be distinguished.

PRH is represented by the following software agents:

a. Leader Agent – it creates the agent cluster, it is an axis of the whole process, and it groups other agents which can cooperate to perform a particular task. It's also responsible for building temporary Trading Partners Registries and extending TPs registries with reorganized BS/D/Ms.
b. Validating Agent – it cooperates with *RosettaNet Validation Tool*, checks and validates BS/D/Ms (whether they are compatible with *RosettaNet* standards).

c. Notifying Agent – notifies TPs' agents when a new scenario appears and informs about the progress of the validation of created by TPs BS/D/M.
d. Error Handling Agent – is responsible for informing TPs about validation failure and inserting errors to error registry.

TPs are represented by the following software agents:

a. Searching Agent – it's responsible for picking the right TPs for specific scenario that would be compatible with the requirements given by delegating it TP.
b. Coordinating Agent – this agent is the most complex one, since it can manage multi-task processes, it cooperates with and manages other agents, it represents the entity which coordinates the whole supply chain and all the processes connected with it, it creates the sub-cluster.
c. Offering Agent – it represents particular entities (e.g. supplier, logistic service provider, insurance company) which take part in creating a specific supply chain.
d. Validating Agent of TP – is responsible for checking and comparing BS/D/Ms and for contact with Validating Agent of PRH (if changes are introduced to BS/D/M).

Exemplification of Using Agent Clusters in PRM

A sample process of building new supply chain within PRM will be introduced in this section.

Let's imagine that a TP-A (e.g. a trading network) wants to find a potential business partners for delivery and distribution of 100 pallets of fresh juice, a weekly cycles, from a distributing center in the California to 30 supermarkets scattered in the US, for no more than $15 for a pallet and in deferred payment of 30 days. The scenario discussed above is quite complex and hence several entities such as producer, supplier, logistic service provider, insurance company will have to be employed in order to succeed with the realization of the scenario.

The Searching Agent (SA) of TP-A starts searching potential TPs appropriate for this specific business scenario. If the searching process is successful SA informs Leader Agent (LA) of PRH about need of a new agent cluster to be created (*initialize* operation, cf. Section 2). LA designates for leader of new created cluster Coordinating Agent (CA) of TP-A. CA invites to the cluster Offering Agents (OAs) of potential TPs (*connect* or *disconnect* operation, cf. Section 2). OA, which do not compete with one another, can start cooperation to place better offer to cluster leader (CA).

Agents merge (cf. Section 2) with one another in order to create temporal and dynamic supply chains (sub-clusters) that would be able to compete with other OAs of TPs or supply chains (sub-clusters). Each sub-cluster has its own CA which supervises the procedures and it selects and integrates potential OAs within specific cluster. Each OA can initiate sub-cluster creation and become CA. The offers of particular agents (OAs) can merge but they can also split depending on the offer details changeability of other clusters, e.g. price, delivery terms, etc. If after negotiation a different SA can propose the same price but better delivery terms, the CAs of this cluster and its sub-clusters have to react in new circumstances and find an alternative OAs that would fill the split gap.

Particular OAs or sub-clusters of OAs, who are subject to CA, work independently on their tasks. For example, OA of a logistic service provider searches for the proper means of transport and empty spaces in warehouses and distribution stores. Before starting the negotiation process between CA of the cluster, CAs of the sub-clusters and independent OAs the CA has to gather all the offers from OAs and pick the best ones.

Next parts of the process are seven stages of TPs agents operation on the Public Registry. To better illustrate those operations experimental environment with the following conditions were created:

- There are two business sides of the process. One is TP-A (cluster leader). Other is TP-B – group of TPs which have united into one sub-cluster to place the best offer to TP-A.
- TP-B is represented by group of its Leaders Agents. Agents of other TPs in sub-cluster only communicate with agents of sub-cluster Leader.
- TP-B entities have to unify their BS/D/Ms within sub-cluster (see stage 2, 3, below). This operation is crucial for business cooperation between TP-A and TP-B in connection with need of one coherent version of BS/D/Ms in TP-A and TP-B registries.

Agent operations on PR are divided into seven stages:

1. Validating Agent (VA) of TP-A communicates with VA of TP-B and informs which BS/D/Ms are required for cooperation (e.g. Notify Of Shipment Receipt, Distribute Inventory Report, Request Shipping Order, Notify of Shipment Documentation). VA of TP-B checks if these BS/D/Ms are in sub-cluster's TP's registries and expands this list of required BS/D/Ms if needed with proper BS/D/Ms from sub-cluster's TP's registries. If there not all BS/D/Ms are present in sub-cluster's TPs registries then proper BS/D/Ms are acquired from other side registry/registries. On this stage only presence of required BS/D/Ms is checked.
2. VA of sub-cluster Leader communicates with PRH's LA to commission temporary Trading Partner Level Registry creation for unified BS/D/Ms. PRH's LA creates temporary registry and grants administration privileges to sub-cluster Leaders agents. Other TPs agents can only communicate with sub-cluster Leader agents and delegate them for administrative work on registry.
3. VA of sub-cluster Leader groups all BS/D/Ms into topical sub-registries (e.g. Notify Of Shipment Receipt sub-registry) in temporary Trading Partner Lever Registry.
4. Each sub-registry's BS/D/Ms are compared with each other to elaborate one unified version of BS/D/M. Authors bring up fragment of standard RosettaNet "Notify Of Shipment Receipt" XML document as an example of unification process. This stage consists of three steps[4]:
   - Firstly leading VA compares all proper BS/D/M's XML documents with each other to determine which fields are missing (see below code example – comparing of two documents is shown, bolded field is missing). Lists of missing fields for each TP with each document are prepared.

[4] (...) symbol placed in code samples shows that there is additional XML code. This procedure was applied due to transparency of the process is increased.

Example of a “Notify Of Shipment Receipt” XML document code. Partner Interface Processes, RosettaNet Standard.

```
Document 1.

(...)
<ShipmentReceipt schemaVersion="">
           (...)
           <ReportDateTime>2007-02-
15T08:30:00+08:00</ReportDateTime>
           <ShipmentIdentifier>String</ShipmentIdentifier>
           <ShipmentReceiptLineItem schemaVersion="">
                  (...)
                  <uuom:UnitOfMeasure
(...)>1BF</uuom:UnitOfMeasure>
           </ShipmentReceiptLineItem>
           <dl:TrackingReference schemaVersion="">
           (...)
           </dl:TrackingReference>
           <TransportedBy schemaVersion="">
           (...)
           </TransportedBy>
</ShipmentReceipt>
(...)

Document 2.

(...)
<ShipmentReceipt schemaVersion="">
           (...)
           <ReportDateTime>2007-02-
15T08:30:00+08:00</ReportDateTime>
           <ShipmentIdentifier>String</ShipmentIdentifier>
           <ShipmentReceiptLineItem schemaVersion="">
                  (...)
           </ShipmentReceiptLineItem>
           <dl:TrackingReference schemaVersion="">
           (...)
           </dl:TrackingReference>
           <TransportedBy schemaVersion="">
           (...)
           </TransportedBy>
</ShipmentReceipt>
(...)
```

- Secondly TPs’ VAs informs leading VA about fields which are obligatory in their BS/D/Ms (c.f. Section 3). Above lists are expanded with information about it (additional ids are added). This model of reorganizing and unifying XML documents is based on simple rule concerning obligatory fields in XML documents: *Obligatory field can’t be removed from document*.
- Thirdly leading VA is reorganizing each XML document if needed. Reorganization can take the form of two opposing processes – simple adding and removing fields (XML makers). When VA removes fields it must check if this field is obligatory (c.f. above rule).

If BS/D/M was changed, it has to be validated by PRH's VA. If the validation is successful, then PRH's Notifying Agent informs TP-B's VA about it. If it isn't, then PRH's Error Handling Agent transfers this information to TP-B's VA and inserts the errors to Error Registry. In this case TP-B's VA starts reorganizing process again but must take all errors into consideration.

5. On this stage VA of TP-A communicates with PRH's LA to commission temporary Trading Partner Level Registry creation for this cluster needs. PRH's LA creates temporary registry and grants administration privileges to TP-A's agents. TP-B's agents can only communicate with TP-A's agents and delegate them for administrative work on registry.
6. VA of TP-A groups own and unified TP-B's BS/D/Ms into topical sub-registries (e.g. Notify Of Shipment Receipt sub-registry) in temporary Trading Partner Lever Registry.
7. Each sub-registry's BS/D/Ms are compared and unified analogical to stage 4.

After unification of BS/D/Ms TPs' VAs inform TPs' CAs about process results. If reorganization were successful each CA inform its delegating entity (TP, TP's back-end system, other agent) about it and cooperation along with RosettaNet standards between TPs starts. When reorganization ends with failure cluster Leader may search for other potential partners and invite their OAs into cluster or simply may inform PRH's LA to cancel the cluster, its related sub-clusters and temporary registries.

When cooperation between TPs ends all temporary registries may be included in Trading Partner Level Registry as sub-registries with proper identifier of cooperation (involved TPs identifiers, type of cooperation, etc.). These sub-registries may be used as base for future cooperation between the same TPs or other TPs.

Of course, all the procedures take place in accordance with *RosettaNet* standards. Trading Partners communicate with one another using *RosettaNet* rules (cf. Section 3).

5 Conclusions

The following main advantages of the proposed approach can be distinguished:

- Agent clusters allow unrestricted collaboration among agents of the same business party, as well as safe and controlled cooperation among different business parties.
- Public Registries and agent technology give the opportunity to integrate smaller supply chains into global supply network.
- Enterprises can automatically search, select and choose future TPs.
- Thanks to embedding agent technology in PRM, automation of comparing, reorganizing and verifying BS/D/M scenarios is greatly facilitated.
- Agents can be able to support efficient supply chain configuration, including partners who offer one another the best cooperation possibilities and conditions at a given time.
- There is a possibility to create temporal and dynamic supply chains aimed at executing a single transaction.

Of course, the approach presented in this paper is general and it can take place not only within supply chains but also in other areas of cooperation within e-business.

Nowadays authors work also on advanced negotiation mechanism. It will allow VAs to negotiate which fields of BS/D/Ms XML documents are really required and which can be omitted.

Next interesting direction of research is cooperation between TPs and/or TPs' agents from different public registries based on the same standard (e.g. *RosettaNet*).

References

[1] Blomqvist, E., Levashova, T., Öhrgen, A., Sandkuhl, K., Smirnov, A., Tarassov, V.: Configuration of Dynamic SME Supply Chains Based on Ontologies. In: 2nd Int. Conference on Industrial Applications of Holonic and Multi-Agent Systems, Holomas (2005)

[2] Call for Participation: In: 4th International Joint Conference AAMAS on Autonomous Agents and Multiagent Systems, www.agtivity.com

[3] Denkena, B., Zwick, M., Woelk, P.O.: Multiagent-Based process Planning and Scheduling in Context of Supply Chains. In: 1st Int. Conference on Industrial Applications of Holonic and Multi-Agent Systems, Holomas (2003)

[4] Foundation for Intelligent Physical Agents, FIPA Abstract Architecture Specification, www.fipa.org/specs/fipaSC00001L/

[5] Foundation for Intelligent Physical Agents, FIPA Communicative Act Library Specification, www.fipa.org/specs/fipa00037/

[6] Foundation for Intelligent Physical Agents, FIPA ACL Message Structure Specification, www.fipa.org/specs/fipa00061

[7] ebXML Registry Information Model v2.0, http://www.oasis-open.org/committees/regrep/documents/2.0/specs/ebrim.pdf

[8] Labarthe, O., Tranvouez, E., Ferrarini, A., Espinasse, B., Montreuil, B.: A Heterogeneus Multi-agent Modelling for Distributed Simulation of Supply Chain. In: 1st Int. Conference on Industrial Applications of Holonic and Multi-Agent Systems, Holomas (2003)

[9] OECD SME and Entrepreneurship Outlook – 2005 edn. www.oecd.org/document/15/0,2340,en_2649_33956792_35096847_1_1_1_1,00.html

[10] Ulieru, M.: The Holonic Enterprise: Modelling Holarchies as Mass to Enable Global Collaboration. In: Mařík, V., McFarlane, D.C., Valckenaers, P. (eds.) HoloMAS 2003. LNCS (LNAI), vol. 2744, Springer, Heidelberg (2003)

[11] Trading Partner Implementation Requirements Standard, RosettaNet Standards, portal.rosettanet.org/cms/sites/RosettaNet/Standards/RStandards/tpir/index.html

[12] Wieczerzycki, W.: Polymorphic Agent Clusters – The Concept to Design Multi-agent Environments Supporting Business Activities. In: Mařík, V., Brennan, R.W., Pěchouček, M. (eds.) HoloMAS 2005. LNCS (LNAI), vol. 3593, Springer, Heidelberg (2005)