Classification of Ad-hoc Multi-lateral Collaborations based on Local Workflow Models *

Andreas Wombacher Bendick Mahleko Thomas Risse
Fraunhofer Gesellschaft, Integrated Publication and Information Systems Institute
64293 Darmstadt, Germany
firstname.lastname@ipsi.fhg.de

ABSTRACT

Establishing multi-lateral collaborations based on local workflows without having a global workflow is complicated, because the set of requirements used for the searching and matchmaking of trading partners is underspecified. The issue is to consider inter-dependencies of trading partners within a local workflow on the one hand side and between services on the other hand side. Within this paper, we suggest to develop an abstract model for establishing multi-lateral collaborations and propose a classification schema based on this model. The classification later on can be used to focus on particular aspects of the problem space and to develop solutions for different clusters within the problem domain.

1. INTRODUCTION

The Internet has become the standard communication platform for EDI. Traditional EDI is well-established in B-2-B scenarios, which are based on conventional contracts describing the regulatory framework. Based on new technologies, like web services, EDI relationships change to more dynamic (ad-hoc) approaches: e-services. Based on e-services ad-hoc multi-lateral collaborations can be established.

Within this paper we present a two-dimensional classification schema for the problem domain used as a basis for searching and finding trading partners in ad-hoc multi-lateral collaborations.

1.1 An Example: Procurement Scenario

As a motivation we describe an exemplary procurement scenario, which includes the following trading parties: customer, vendor, shipping company and bank. The aim of the customer is to buy some goods from a vendor using his credit card and potentially to do an online tracking of the shipped goods. The vendor intends to sell goods to a customer offering different payment methods and to use an external shipping service. The two remaining parties (bank and shipper) offer core services, that is different payment methods and shipping goods including parcel tracking.

Each of these parties has its own local workflow and provides respectively requests services from other trading parties, without knowing their local workflows. Thus, the main challenge is to set up an ad-hoc multi-lateral collaboration based on several local workflows resulting in a consistent and meaningful global workflow.

Figure 1 depicts the above mentioned scenario, where the white rectangles mean the local workflow of a particular party, the light gray boxes depict a service communication infrastructure used by the local workflow engine and the dark gray boxes represent offered and requested external services respectively. The dotted lines specify a mapping of a transition within the local workflow and receiving or sending a message by an external service.

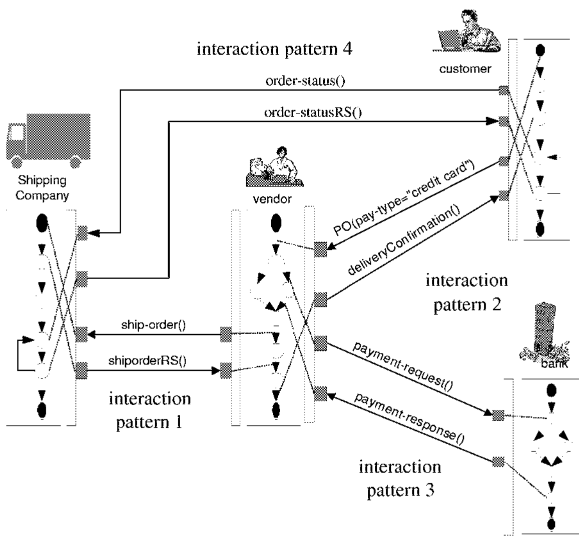

Figure 1: Multi-lateral Collaboration

1.2 Open Issues

From the above example, the global workflow can be understood and executed quite easily, but the global workflow does not exist a priori. Further, it is not possible to compose the global workflow on behalf of the local ones, because not all parties are willing to provide their local workflow to

*An extended version of this paper is available at www.ipsi.fhg.de/oasys/reports

SAC 2003, Melbourne, Florida, USA

trading partners, because mission critical information may be contained. Nevertheless, it is possible to calculate **interaction patterns** (as annotated in Figure 1) describing the potential message sequences exchanged between different trading partners without including mission critical information. So, these can be published and used for searching potential trading partners.

Based on the fact that no global workflow can be derived, the open issue is finding and matchmaking appropriate trading partners on behalf of interaction patterns (workflow aspects in addition to simple key word annotations) and how to set up an agreement at all.

In particular, it is impossible to statically derive an optimal plan for setting up a multi-lateral contract, because no global workflow is available and the plan calculation is based on partial information only, which is definitely not sufficient. The way of influencing the contracting is limited to the selection of trading partners, which affects the number of potential ways to set up a contract, which also affects the number of potential different kinds of comparisons of interaction patterns. We illustrate this aspect in Section 4.

A further challenge is considering dependencies between messages within different interaction patterns. To illustrate this, we consider interaction patterns 1 and 4 of the example in Section 1.1. There is no way to do parcel tracking before the parcel is shipped. This dependency of the two interaction patterns is semantically quite obvious, whereas the possibility to derive such dependencies based on interaction patterns is difficult. We are describing different cases of this problem in Section 3.

The aim of this paper is to do a classification for a better understanding of the problem domain service discovery (searching and matchmaking of services) to establish a multi-lateral collaboration. We are convinced, that such a classification is required to be able to position proposed approaches in the problem space to give a clear understanding of the addressed problem area.

2. MODEL OF MULTI-LATERAL COLLABORATION

In the following, we define a model used to formally express the classification schema by an abstract definition of multi-lateral collaboration. We start with an abstraction of aspects, which can be represented by *name-value* pairs, and the description of the underlying assumptions, which simplify the model and reduce formalization. Followed by several definitions, we pick up the example in Figure 1 for clarification and summarization of open issues.

2.1 Underlying Assumptions

The proposed model does not consider all details required for a real agreement assignment, but focuses on the parts related to structural aspects of the local workflow needed for setting up multi-lateral collaboration. Issues not addressed within the model are constraints influencing the process of setting up an agreement, which can be modeled as well understood *name-value* pairs.

Thus, we base our model on the following assumptions on workflows as explained in Section 1.2:

- No global workflow specification a priori exists.
- Each trading party has his own private local workflow and publishes partial information (interaction patterns) to the outside world.
- A multi-lateral agreement is set up by a combination of bilateral agreements.
- A global ontology of trading party names exists, which is modelled by introducing roles and basing all definitions on roles instead of concrete trading parties (instances).
- A domain specific ontology of messages exists, as it already does in EDI environments like X12, EDIFACT, or Rosetanet.

The last two assumptions are made to reduce the complexity of the model introduced in the next subsection. The model can be extended to overcome these assumptions, but then a formalism describing a mapping of different names used by different trading parties is required. In particular, this is still an open issue, which is addressed by the semantic web community [5, 9, 1].

These assumptions introduce simplification to the model as well as properties of the workflow structure of the observed system.

2.2 Definitions

While introducing the model, we use the following nomenclature: atomic elements are written as underlined lower case letters (e.g. $\underline{r}$), while domains are specified by upper case letters (e.g. R set of all potential $\underline{r}$). Sets of elements are represented as lower case letters (e.g. $r \subseteq R$). We use roles for representing trading parties (see Section 2.1).

DEFINITION 1. (Role)
A role is defined as a prototypic trading partner within a business process without naming a concrete service instance fulfilling it. The set of all disjoint roles is denoted as R.

The procurement example described in Section 1.1 contains the following set of roles: a bank, a shipper, a customer and a vendor, so $R = \{\underline{b}, \underline{s}, \underline{c}, \underline{v}\}$.
Using the definition of roles, we introduce now local workflows. These are often modeled using Petri nets [15, 17] or statecharts [10, 4]. Within this paper we use an abstraction of the concrete underlying workflow model and define a function to derive the set of external trading partners.

DEFINITION 2. (local workflow)
A local workflow is internal to the owning role and may contain interactions to external trading partner. A local workflow of role $\underline{r} \in R$ is defined as $WF(\underline{r})$. External trading parties (roles) of required services are represented as $ext(WF(\underline{r})) \subseteq R \setminus \{\underline{r}\}$.

In the example given in Figure 1 four local workflows are depicted as white boxes. Furthermore, the external trading parties of the shipper company are the customer and the vendor, so $ext(\mathrm{WF}(\underline{s})) = \{\underline{c}, \underline{v}\}$.
Based on roles and the abstraction of a local workflow, we now introduce bilateral interaction patterns representing sequences of exchanged messages. We give here an abstract definition of bilateral interaction patterns, because this is sufficient to our needs.

DEFINITION 3. (bilateral interaction pattern)
A bilateral interaction pattern is a tuple containing the role of the partner owning the local workflow and the role of an external trading partner (requested service), so both parties involved in a bilateral interaction. The set of all interaction patterns associated with a specific role $\underline{o}$ *is defined as* $I(\underline{o}) := \{(\underline{o},\underline{e}) \mid \underline{e} \in ext(WF(\underline{o}))\}$. *The set of all potential interaction patterns* $I : R \times R$ *is the union of all interaction patterns provided by the elements of R, that is* $I := \{(\underline{o},\underline{e}) \mid \underline{o} \in R \wedge \underline{e} \in ext(WF(\underline{o}))\}$.

In our example, the shipper company has the following set of interaction patterns: $I(\underline{s}) = \{(\underline{s},\underline{c}),(\underline{s},\underline{v})\}$.

The functional dependency between local workflow and bilateral interaction pattern is elaborated in [16], where the interaction pattern (modeled as deterministic finite state automaton (DFA)) is derived from a statechart workflow model based on a function view(). Within this model, we denote this dependency as $(\underline{o},\underline{u}) := view(WF(\underline{o}),\underline{u})$ or as a shorthand version $(\underline{o},\underline{u}) := view_{\underline{o}}(\underline{u})$.

Next, we define convenience functions to derive the set of external trading parties *ext*() respectively the *owner*() of an interaction pattern $\underline{i}$.

DEFINITION 4. (owner and ext)
The owner function returns the owner role of an interaction pattern, while the ext function returns the owner role of an externally requested service. Let $\underline{i} = (\underline{o},\underline{e})$ *then*

- $owner : I \longrightarrow R$ *with* $owner(\underline{i}) = owner((\underline{o},\underline{e})) := \underline{o}$
- $ext : I \longrightarrow R$ *with* $ext(\underline{i}) = ext((\underline{o},\underline{e})) := \underline{e}$

We extend this definition to sets of interaction patterns $i \subseteq I$

- $owner : 2^I \longrightarrow 2^R$ *with* $owner(i) := \{owner(\underline{i}) \mid \underline{i} \in i\}$
- $ext : 2^I \longrightarrow 2^R$ *with* $ext(i) := \{ext(\underline{i}) \mid \underline{i} \in i\}$

With regard to the procurement example, the shipper company provides the following set of interaction patterns: $I(\underline{s}) = \{(\underline{s},\underline{c}),(\underline{s},\underline{v})\}$. Applying the owner function to interaction patterns $I(\underline{s})$, the result is a set containing the role of the shipper company, so $owner(I(\underline{s})) = \{\underline{s}\}$. Using the ext function, the result is a set containing the vendor and the customer role, so $ext(I(\underline{s})) = \{\underline{v},\underline{c}\}$.

Based on the definition of roles and interaction patterns we introduce an abstract definition of service and atomic services, which are the basis for discussing agreements and contracts later on.

DEFINITION 5. (service, atomic service)
A service is defined as a tuple containing the set of already instantiated roles and the set of interaction patterns with still uninstantiated external roles. The set of all services S is defined as $S : 2^R \times 2^I$. *An atomic service* $\underline{s}_{\underline{r}}$ *provided by a role* $\underline{r} \in R$ *is defined as* $\underline{s}_{\underline{r}} = (\{\underline{r}\}, I(\underline{r}))$.

With regard to the introductory example, the atomic service offered by the shipping company can be characterized by $(\{\underline{s}\},\{(\underline{s},\underline{c}),(\underline{s},\underline{v})\})$, that is the own role name and the set of corresponding interaction patterns.

Based on this abstract definition of services, we define agreements as a function combining two services to a new compound service.

DEFINITION 6. (agreement, contract)

1. *An agreement is a function agree() combining two services to a new compound one.*
   $agree : S \times S \longrightarrow S$ *and* $(s,s') := agree((a,a'),(b,b'))$
   with $s := a \cup b$ *and*
   $s' := \Big(a' \setminus \{(\underline{x},\underline{x}') \mid \underline{x} \in a \wedge \underline{x}' \in b\}\Big) \cup$
   $\Big(b' \setminus \{(\underline{y},\underline{y}') \mid \underline{y} \in b \wedge \underline{y}' \in a\}\Big)$
2. *A contract is an expression based on a set of atomic services* $\{\underline{s}_{\underline{r}_1},\ldots,\underline{s}_{\underline{r}_n}\}$, *which are combined by a nested application of the agreement function resulting in a compound service* $\underline{s}_c = (\{\underline{s}_{\underline{r}_1},\ldots,\underline{s}_{\underline{r}_n}\},\emptyset)$, *where all roles* $\underline{r}_1,\ldots,\underline{r}_n$ *are covered within the set of instantiated roles and no un-instantiated interaction pattern exists.*

When applying this definition to the example given in Figure 1 and applying an agreement between shipping company and customer, the following formula holds:
$agree((\{\underline{s}\},\{(\underline{s},\underline{c}),(\underline{s},\underline{v})\}),(\{\underline{c}\},\{(\underline{c},\underline{s}),(\underline{c},\underline{v})\})) =$
$(\{\underline{s},\underline{c}\},\{(\underline{c},\underline{v}),(\underline{s},\underline{v})\})$

A potential contract of this example may be represented by the following expression $agree(\underline{s}_{\underline{s}}, agree(\underline{s}_{\underline{c}}, agree(\underline{s}_{\underline{b}},\underline{s}_{\underline{v}})))$ resulting in a compound service $(\{\underline{b},\underline{v},\underline{c},\underline{s},\},\emptyset)$

This model of agreement and contract allows a discussion of setting up multi-lateral agreements based on local workflows, mainly the service discovery aspects. The presented model deals with sets only, while concrete workflow modeling typically is based on graphs. Thus, we reduced the complexity of the model and continue to focus on setting up an agreement.

There exists necessary constraints for setting up an agreement, which are summarized in the following lemma. The sufficient parts of the conditions are used as one dimension of the classification and are explained in more detail in Sections 2.3 and 4.

LEMMA 1. (agreement constraints)
Let (a,a') *and* (b,b') *be two services. The necessary constraints for applying the agreement function are:*

- $a' \neq \emptyset \wedge b' \neq \emptyset$
- $a \cap ext(b') \neq \emptyset \wedge b \cap ext(a') \neq \emptyset$

In the following we describe the two dimensions of the proposed classification.

2.3 Classification Criteria

The definition of agreement given above is based on bilateral interaction patterns derived from a local workflow. Considering an exemplary scenario, where more than one external party is involved in a local workflow ($|ext(I(\underline{o}))| > 1$), it may be that two different interaction patterns are dependent on each other. Such a scenario is captured by the example in Section 1.1, where the customer needs first a shipment confirmation by the vendor before parcel tracking can be started, because the customer does not know the shipping company and session information beforehand.

Dependencies between external roles can not be represented in a set of bilateral interaction patterns, but have a strong impact on the matchmaking of potential trading partners. We use this as a classification criteria of the problem space and elaborate on this in Section 3.

After finding an appropriate trading partner, the agreement constraint (based on the set of minimal requirements as given in lemma 1) is applied and an agreement is assigned or not. The agreement constraint thus has a great influence on the global workflow structure, at least as it can be influenced by a single role. In particular, extensions of the set of minimal requirements are describing the **dependency between external services** combined by an agreement. We will use these additional requirements as a second classification criteria to structure the problem space. We elaborate on this in Section 4.

3. ROLE DEPENDENCY

We investigate the dependencies between external roles with regard to the matchmaking of trading parties.

3.1 Multi-lateral Definitions

We have already defined bilateral interaction patterns in definition 3 and now extend this to the multi-lateral case.

DEFINITION 7. (multi-lateral interaction pattern)
Let $\underline{o} \in R$ *be the owning role and* $e \neq \emptyset$ *a subset of external roles, then the set of multi-lateral interaction patterns* $I_m(\underline{o})$ *is defined as* $I_m(\underline{o}) := \{(\underline{o}, e) \mid e \subseteq ext(WF(\underline{o}))\}$. *The extension to a set of roles is given as* $I_m(o) := \bigcup_{\underline{o} \in o} I_m(\underline{o})$

To be able to do the classification, we introduce definitions for union, subsumption and equivalence of interaction patterns. As stated above the interaction patterns can be understood as DFAs, thus the union, subsumption and equivalence of interaction patterns is given in accordance with corresponding DFA definitions (see e.g. [7]).

DEFINITION 8. (union of interaction patterns)
Let $\underline{i}$ *and* $\underline{j}$ *be interaction patterns, then the union* $\cup_{IP}$ *of* $\underline{i}$ *and* $\underline{j}$ *is defined by the union of the two languages accepted by the interaction patterns, that is a message sequence is accepted, if it is accepted by both interaction patterns.*

The relationship between a multi-lateral interaction pattern and the union of bilateral interaction patterns is used as criteria to classify the role dependency dimension. To be able to express different options, we introduce now subsumption and equivalence of interaction patterns.

DEFINITION 9. (subsumption of interaction patterns)
Let $\underline{i} = (\underline{o}, e)$ *and* $\underline{j} = (\underline{o}', e')$ *be single multi-lateral interaction patterns.* $\underline{i}$ *is subsumed by* $\underline{j}$ *that is* $\underline{j} \subseteq_{IP} \underline{i}$ *if all message sequences accepted by interaction pattern* $\underline{i}$ *are also accepted by* $\underline{j}$.

Based on the definition of subsumption, equivalence can easily be defined as:

DEFINITION 10. (equivalence of interaction patterns)
The interaction patterns $\underline{i}$ *and* $\underline{j}$ *are equivalent, iff* $\underline{i}$ *subsumes* $\underline{j}$ *and vice versa, that is*
$\underline{i} \equiv_{IP} \underline{j} \Longleftrightarrow (\underline{i} \subseteq_{IP} \underline{j}) \wedge (\underline{j} \subseteq_{IP} \underline{i})$.

We use these definitions to distinguish the different types of role dependencies mentioned below.

3.2 Role Independence

The first and easiest case is that the external roles involved in a local workflow are completely independent of each other. Because of the independence, the multi-lateral interaction pattern is equivalent to the union of bilateral interaction patterns, that is $(\underline{o}, e) \equiv_{IP} \bigcup_{\underline{e} \in e}{}_{IP}(\underline{o}, \underline{e})$ with $e \subseteq ext(\mathrm{WF}(\underline{o}))$.
This has a great impact on matchmaking, because the maintenance of the multi-lateral interaction patterns derivable by the local workflow can be reduced to the maintenance of bilateral interaction patterns.

3.3 Role Dependence

The second and most complex case is that the external roles involved in a local workflow are dependent on each other. In particular, the union of the bilateral interaction patterns accept more message sequences as the local workflow does, that is $\bigcup_{\underline{e} \in e}{}_{IP}(\underline{o}, \underline{e}) \subseteq_{IP} (\underline{o}, e)$ with $e \subseteq ext(\mathrm{WF}(\underline{o}))$.
Here, for doing a correct matchmaking, the search engine must maintain all potential multi-lateral interaction patterns. There exists a high number of possibilities of comparing the potential interaction patterns within the matchmaking, thus the maintenance of interaction patterns is quite complex and does not guarantee good searching results.

3.4 Recoverable Role Dependence

The last and maybe most common case is partial role dependency, which is recoverable by using additionally provided mission un-critical information. Here the two interaction patterns can be merged by using this additional information to recover the ordering of the different messages used by the two interaction patterns. We informally introduce the function merge(). Then the following is valid
$\bigcup_{\underline{e} \in e}{}_{IP}(\underline{o}, \underline{e}) \subseteq_{IP} \mathrm{merge}(\underline{e}_1, \ldots, \underline{e}_n) \equiv_{IP} (\underline{o}, e)$
with $\{\underline{e}_1, \ldots, \underline{e}_n\} = e \subseteq ext(\mathrm{WF}(\underline{o}))$.
The advantage is that, similar to the independent case the bilateral interaction patterns are sufficient for doing a complete matchmaking. Nevertheless, the merging is more time consuming than a simple union. Additional information may be assignment of global variables or comparisons of global variables and message parameters.

4. SERVICE DEPENDENCY

Next, we investigate the effects of sufficient constraints for the agreement function on the global workflow structure as far as it can be influenced by a single role.

The figures used within this section are based on the following legend: a white box contains the specification of a service. A gray circle represents an interaction pattern, while a white circle contains the local workflow of the instance itself.

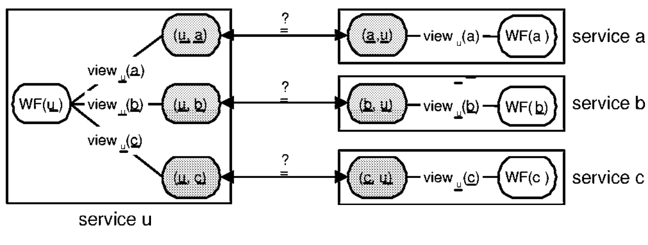

Figure 2: centralized assignment of agreements

4.1 Fully Centralized Model

The easiest case to deal with service dependency is the fully centralized model as depicted in Figure 2, where a service user is searching for multiple service providers, but accepts matches only, if the service provider does not require additional services than the user one. So, the different services used by the service user are independent of each other and have only interaction with the service user itself. An example is an online shop, which provides a customer a selling service by coordinating the involved shipping company, bank and vendor.

The additional requirements used for the agreement constraint are defined on the services $\underline{u} = (u, u')$ and $\underline{a} = (a, a')$ as:

$ext(a') \subseteq u \wedge a \subseteq ext(u') \wedge |ext(a')| = 1 \wedge |a| = 1$

The advantage of this case is that the coordinator (service $\underline{u}$) controls the establishment of the contract completely. The trading partners involved do not cooperate with additional services. The matchmaking is limited to bilateral interaction patterns only. With regard to the first dimension of the classification only the role independence (Section 3.2) can be applied.

4.2 Partially Centralized Model

The partially centralized model is quite similar to the centralized one, but takes into consideration that a requested service may be an already compound service providing the services of two dependent roles.

This effects the agreement constraint by omitting the constraint on the power of the set a. The extension of the agreement constraints for a service user $\underline{u} = (u, u')$ and a service provider $\underline{a} = (a, a')$ is

$ext(a') \subseteq u \quad \wedge \quad a \subseteq ext(u') \quad \wedge \quad |ext(a')| = 1.$

The advantage of this model is that the user is still controlling the establishment of a contract, but it is less restrictive by accepting also compound services. With regard to the first dimension of the classification, the role independence (Section 3.2) and the recoverable role dependence (Section 3.4) may be applied. But still the different services are independent of each other.

An exemplary use case is a B-2-B scenario, where a service provider offers message processing (service $\underline{b}$) and message transformation (service $\underline{a}$) within a specific domain.

4.3 Decentralized Model

The decentralized model as depicted in Figure 3 does not restrict the interrelations between the different service providers. An exemplary scenario is the motivation example described in Section 1.1 explaining a procurement scenario involving a parcel tracking. Here all different models of the role dependencies may be applied.

Within this generic scenario, it is not possible to define a unique way to limit the agreement constraint any further. Nevertheless, there exists different ways of extending the agreement constraint to achieve additional properties. We want to give some examples based on service $\underline{a} = (a, a')$ and $\underline{b} = (b, b')$:

- $a \cap b = \emptyset$
  the agreement is assigned between two services, which have no common roles instantiated already

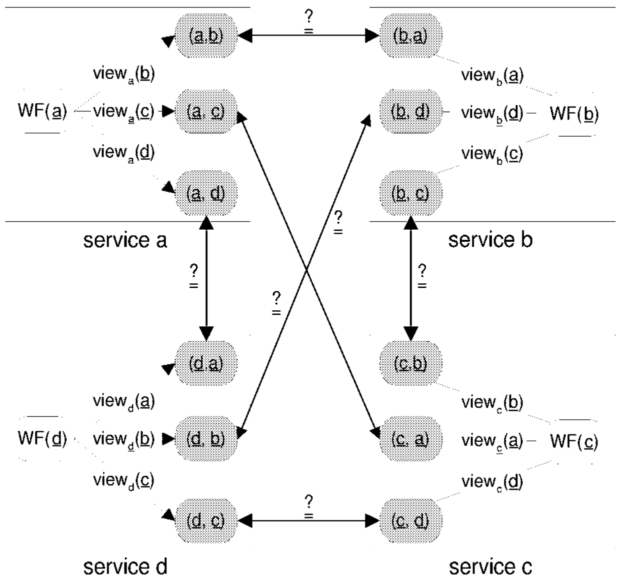

Figure 3: decentralized assignment of agreements

- $ext(b') \subseteq a$
  the new agreement does not introduce new uninstantiated roles into the new compound service
- $b \cup ext(b') \subseteq a \cup ext(a')$
  the agreement does not introduce new roles
- $b \subseteq a \cup ext(a')$
  all instantiated roles of B are also used in A, thus new roles are introduced by B

The above list of agreement constraint extensions is not exclusive at all. The possibilities to control the contract establishment are very limited. Dependent on the role dependency the efforts for doing matchmaking are quite high. The different services involved have side effects, that it may introduce new dependencies to other roles.

5. RELATED WORK

Several approaches for modeling workflows exist. One option is using statecharts for modeling workflows as proposed in [10, 4]. Another option is using petri nets as the underlying method for the modeling as done in [15, 17]. Nevertheless, modeling inter-organizational workflows with these approaches require knowledge of a global workflow model. We are aiming to overcome the need of global workflow knowledge, because in our opinion this can not be assumed in the domain of ad-hoc e-services. In [15] an approach is described for establishing interorganizational workflows. Based on a global workflow definition partitions of local workflows are calculated.

The approaches mentioned above are the methodological basis for different technological infrastructure as for web services, where exemplary WSFL [8] is based on petri nets and Selv-Serv [12] is based on statecharts.

The issue of setting up contracts has been addressed in the domain of digital rights management quite often. Here, typical approaches have been developed, which are securely encapsulating the traded content and the corresponding business model and passing both to the trading parties for execution. Exemplary systems are SeCo [11] and DigiBox [13], which are based on a global workflow defined by the instance encapsulating the content and the associated process description.

A potential technological infrastructure for implementing the example and facing the above mentioned open issues is the domain of web services. Related standards for service discovery (WSDL [6], UDDI [2], WSIL [19], WSCL [3]) as well as for describing local and global workflows (BPEL4WS [18], XLANG [14], WSFL [8]) are already available or under construction. None of them is dealing with establishing multi-lateral collaborations.

Because our main focus is on service discovery, we are also mentioning semantic web [5] as related work in the sense of an extended search capability of semantically annotated e-service descriptions as mentioned in [9, 1]. Within this domain, the focus is on semantic annotation, while we are focusing on doing structural matchmaking based on local workflows.

6. CONCLUSION AND OUTLOOK

Within this paper, we described a model of setting up multi-lateral collaborations based on **bilateral partial agreements**, and **knowledge of local workflows only**.

We extracted two dimensions specifying the problem space, investigated important cases within each dimension, and discussed affects on the searching and matchmaking of interaction patterns.

Our model is based on the assumption of a global ontology of roles, which seems to be not very likely. To overcome this assumption an "unification" of roles that is a semantical matchmaking of role descriptions is required. This task is addressed by the semantic web community by means of semantic meanings of role names, while we are observing the possibilities based on structural aspects in a different paper.

Future work will address the searching and matchmaking of interaction patterns in different areas of the problem space based on the introduced classification to come up with a search engine for e-services capable of considering structural searching and establishing multi-lateral collaborations.

7. REFERENCES

[1] D.-S. C. Anupriya. DAML-S: Web serice description for the semantic web, 2002.

[2] I. Ariba, I. Corporation, and M. Corporation. Universal description, discovery and integration, September 2000. http://www.uddi.org/.

[3] A. Banerji, C. Bartolini, D. Beringer, V. Chopella, K. Govindarajan, A. Karp, H. Kuno, M. Lemon, G. Pogossiants, S. Sharma, and S. Williams. Web services conversation language (WSCL) 1.0 W3C note, March 2002. http://www.w3.org/TR/wscl10/.

[4] M. Benyoucef and R. K. Keller. An evaluation of formalisms for negotiations in e-commerce. *Proceedings of the Workshop on Distributed Communities on the Web*, June 2000.

[5] T. Berbers-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific America*, 284(5):34–43, 2001.

[6] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web services description language (WSDL) 1.1 W3C note, March 2001. http://www.w3.org/TR/wsdl.

[7] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2001.

[8] F. Leymann. Web services flow language (WSFL 1.0), May 2001. http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf.

[9] S. McIlraith, T. Son, and H. Zeng. Semantic web services. *IEEE Intelligent Systems (Special Issue on the Semantic Web)*, April 2001.

[10] A. Peron. Statecharts, transition structures and transformations. In *Proceedings International Conference Colloquium on Trees in Algebra and Programming (CAAP- TAPSOFT'95, Springer, LNCS 915)*, pages 454–468, 1995.

[11] A. Runge, B. Schopp, and K. Stanoevska-Slabeva. The management of business transactions through electronic contracts. In A. Camelli, A. M. Tjoa, and R. Wagner, editors, *Proceedings for the 10th International Workshop on Database asn Expert Systems Applications*, pages 824–831, Florence, Italy, 1999. IEEE Computer Society Press.

[12] Q. Sheng, B. Benatallah, M. Dumas, and E.-Y. Mak. SELF-SERV: A platform for rapid composition of web services in a peer-to-peer environment. In *Proc. of 28th VLDB Conference*, Hong Kong, China, 2002.

[13] O. Sibert, D. Bernstein, and D. V. Wie. The DigiBox: A self-protecting container for information commerce. In *Proc. 1st USENIX workshop on Electronic Commerce*, pages 171–183, 1995. citeseer.nj.nec.com/sibert95digibox.html.

[14] S. Thatte. XLANG web services for business process design.

[15] W. van der Aalst and M. Weske. The P2P approach to interorganizational workflows. In *Proc. of 13. Int. Conf. on Advanced Information Systems Engeneering (CAISE'01)*, Interlaken, Switzerland, 2001.

[16] A. Wombacher and B. Mahleko. Finding trading partners to establish ad-hoc business processes. In *Proc. of COOPIS*. Springer, Oct. 2002.

[17] W.v.d.Aalst. Process-oriented architectures for electronic commerce and inter-organizational workflows. *Information Systems*, 24(8):639–671, 1999. Elsevier.

[18] Business processes and transactions for web services. http://www.ibm.com/software/solutions/webservices/bpel.html.

[19] Web services inspection language (ws-inspection) 1.0. http://www.ibm.com/developerworks/webservices/library/ws-wsilspec.html.