IPR: Automated Interaction Process Reconciliation

Zongxia Du[1], Jinpeng Huai[1], Yunhao Liu[2], Chunming Hu[1], Lei Lei[1]

[1]*School of Computer Science*
Beihang University
Beijing, 100083, China
duzx@act.buaa.edu.cn

[2]*Department of Computer Science*
Hong Kong University of Science & Technology
Hong Kong
liu@cs.ust.hk

Abstract

Inter-organizational business processes usually require complex and time-consuming interactions between partners than simple interactions supported by WSDL. Automated reconciliation is essential to enable dynamic inter-organizational business collaboration. To the best of our knowledge, however, there is not a practical automated reconciliation algorithm available. In this paper, we propose a practical automated reconciliation algorithm, called IPR (Interaction Process Reconciliation) based on Petri Net, which is able to effectively facilitate dynamic interactions among trading partners in a peer-to-peer fashion. We implement a prototype IPR server in our lab, and evaluate our design by comprehensive experiments. Results show that IPR significantly outperforms existing approaches in terms of matching success rate, response time, and matching efficiency.

1. Introduction

To date, E-business is evolving from long-lasting well-defined business relationships to a more dynamic situation, in which parties with no prior trading relationships collaborate to implement their business transactions. How to realize the automation of business-to-business (B2B) interactions is reckoned as a hot topic for corporate IT. The compatibility of partners is the basis of realizing the automation of a B2B interaction. Previous works [17] focus upon the compatibility of individual stateless messages, which is insufficient as B2B interactions usually require more complex and time-consuming interaction processes.

Consider the following example as shown in Fig.1, where two interaction processes represent a pair of partners: a customer and a vendor. The customer process in Fig.1 (a) initiates the process with an order request. It then waits for a confirmation before payment. On the other hand, the vendor in Fig. 1 (b) is waiting for payment before confirming the order. Although the interaction processes of the customer and the vendor match at the individual message level, they require different sequences for payment and delivery. Consequently, this interaction brings about a dead-lock, and it is unable to be accomplished.

Several studies have focused on the compatibility of complex inter-organizational interaction processes. In [12], the compatibility of interaction processes was verified based on FSP (Finite State Process). In [2], the authors proposed to match the business processes based on aDFA (annotated Deterministic Finite State Automata). In existing approaches, however, the interaction processes cannot be reconciled to produce an agreement, although reconciliation is effective in finding appropriate partners with a high success rate of matchmaking. For example, if the required message (*v*, *c*, *orderConfirm*) in the customer process as Fig 2 (a) can be released, the partners can interact successfully. The major issue here is how to automatically relax exact matchmakings to fuzzy matchmakings, so as to improve the success rate of E-business interactions without losing the autonomy of peers.

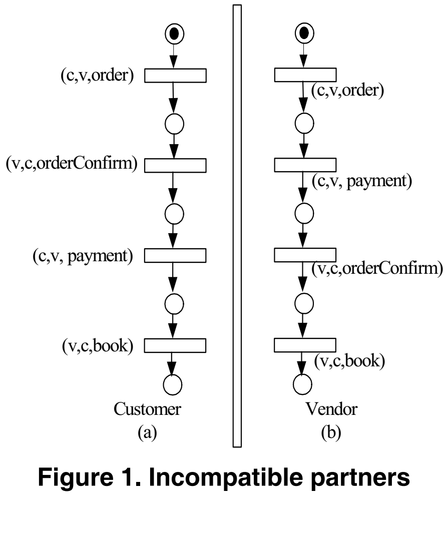

Figure 1. Incompatible partners

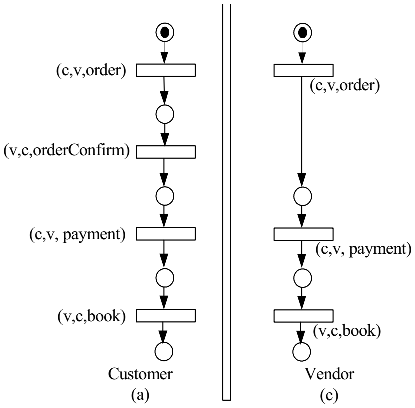

Figure 2. Compatible partners with reconciliation

Automated reconciliation is essential to enable dynamic inter-organizational business processes [1]. To the best of our knowledge, however, there is no feasible automated reconciliation algorithm available. In this paper, we propose a practical automated reconciliation algorithm, IPR (Interaction Process Reconciliation) based on Petri Net. The consistency of individual messages is guaranteed through generic ontology, such as RosettaNet [18].

We implement a prototype IPR server in our lab, and evaluate our design by comprehensive experiments. Results show that IPR significantly outperforms existing approaches in terms of matching success rate, response time, and matching efficiency. IPR is able to effectively facilitate dynamic interactions among trading partners in a peer-to-peer fashion [10, 19].

The rest of this paper is organized as follows. We discuss related works in Section 2, and introduce IPR algorithm in Section 3. Section 4 presents the experiment methodology. Section 5 evaluates IPR. Finally, we conclude this work in Section 6.

2. Related Work

Traditionally, inter-organizational B2B integrations usually adopt a top-down fashion [3, 11, 13] and starts with a definition of the predefined global process in languages like WS-CDL [16], etc. Such a process is then decomposed by individual partners into coarse-grained sub-processes. The partners further decompose the respective tasks into locally controllable fine-grained internal processes using a public to private method based on Petri Net in [3] or the 3-level hierarchical structure in [11]. Although easy to use and control, these approaches lack autonomy and flexibility due to their predefined nature. They are not fit for established applications or flexible collaborations.

Peer-to-peer fashion [1, 2, 12] is better for flexible collaborations between business partners. The interaction business process of a partner is called a "peer". Potential participants define their requirements and contributions in the forms of web service interaction processes, i.e. Abstract Processes [9], to specify mutually visible message exchange of each partner involved, without revealing their internal behaviors. Generally, peer-to-peer fashion is superior to hierarchical ones in terms of flexibility and practicability. However, it is always challenging to construct such a dynamic structure, the consistency of individual messages and the matching of sequences has to be guaranteed.

In UDDI [8], string matching is applied to service discovery. To improve accuracy, semantic matching based on the ontology in a specific domain is also employed [17]. However, these approaches can only discover and match strings for simple stateless services, and are insufficient for complex inter-organization interactions, where the processes need to be matched as well.

BPEL4WS Abstract Processes (Business Process Execution Language for Web Services, Abstract Processes) [9] and WSCL [5] enable a certain organization to describe its interaction process. A.Wombacher et al. [2] propose an automata-based approach, in which process branches are divided into optional and indispensable ones. Matchmaking criterion lies in whether the intersection of two aDFAs (annotated Deterministic Finite State Automata) is NULL. The limitation is that only optional process branches can be selected, but the optional tasks in a sequence process cannot be skipped, such as (*v,c,orderconfirm*) in Fig.2 cannot be skipped for an agreement. In addition, finite state automata cannot describe parallel branches in processes. The authors in [12] attempt to address the issue of interaction compatibility by verifying the interface compatibility, safety compatibility and liveness compatibility of web service compositions. However, the incompatible business process cannot be modified automatically because of a lack of reconciliation methods.

3. IPR Design

3.1. IPR Framework

The framework of IPR is illustrated in Fig. 3. The IPR server is a key part of this framework executing reconciliations between service requesters and providers. In this design, potential partners are required to register their service information and interaction processes in an extended service directory. All service requesters find an appropriate partner through the IPR server. The extended service directory is responsible for managing service categories and process information of service providers.

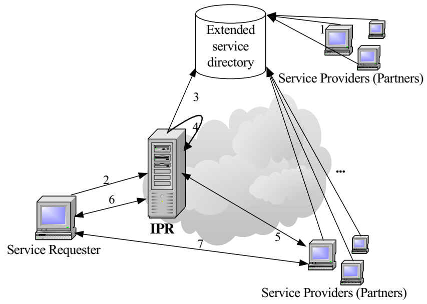

Figure 3. Framework of IPR

The processing flow of this framework, as illustrated in Fig. 3, is as follows.

1) Service providers register their service information and interaction processes to the extended service directory;

2) A service requester sends its requirements to the IPR server to query a compatible service provider;

3) IPR queries the extended service directory to find the result set of service providers according to the required service type;

4) IPR initializes a reconciliation thread for each provider to map an appropriate service provider, which we called a Right Provider (RP);

5) IPR sends the reconciliation result and the related modification to RP to evaluate the validity of reconciliation result;

6) IPR sends the RP information, the reconciliation result, and the related modification to the service requester;

7) The requester handshakes with RP and does business with it if it agrees with the reconciliation result.

3.2. Overview of Process Inheritance

The proposed algorithm of IPR is based on the concepts in [4], such as WF-net and the inheritance of workflow, and so on. The interaction processes are translated to WF-net and reconciled according to their intersection part.

Definition 1 (WF-net)*:* Let $PN = (P, T, F, l)$ be a WF-net if and only if the following conditions are satisfied:

1. $P \subseteq U$ is a finite set of places, where U is some universe of identifiers;
2. $T \subseteq U$ is a finite set of transitions such that $P \cap T = \varnothing$;
3. $F \subseteq (P\times T)\cup(T\times P)$ is a set of directed arcs, called the flow relation;
4. l: $T \rightarrow L$ is a labeling function, where L is a set of action labels;
5. P contains an input place $i \in U$ such that $\bullet i = \varnothing$;
6. P contains an output place $o \in U$ such that $o \bullet = \varnothing$;
7. $\overline{PN} = (P, T \cup \{\bar{t}\}, F \cup \{(o, \bar{t}), (\bar{t}, i)\}, l \cup \{(\bar{t}, \tau)\})$ is strongly connected.

Definition 2(Alphabet): The alphabet of WF-net $PN = (P, T, F, l)$ is defined as a set of visible labels of all transitions of the net: $\alpha(PN) = \{l(t) \mid t \in T \wedge l(t) \neq \tau\}$, where τ is unobservable transition labels.

The inheritance of the workflow proposed by W. van der Alast includes two transformation methods [1, 4]. The former, named blocking a task, simply disallows the execution of a task, while the latter, named hiding a task, allows the execution of a task without considering its effect on the process. By blocking a task, the branches connected to the task are separated from the main tree and become unreachable. By hiding a task, the transitions are still available and the process traverses the task, independently of its execution. Take the three processes in Fig. 4 as an example. Hiding Task *d* in Process (2) or blocking Task *d* in Process (3) results in a transformation to Process (1), while blocking Task *d* in Process (2) makes Task *c* inaccessible and hiding Task *d* in Process (3) leads to a direct connection between Tasks *a* and *c*, which is impossible in Process (1).

In addition, taking branching bisimilarity as the standard equivalence relationship on WF-nets, W. van der Aalst proposes inheritance rules to implement blocking and hiding methods. An inverse -protocol-inheritance-preserving transformation rule, $rPTS^{-1}$, is defined to process the alternative branch tasks. An inverse-projection-inheritance-preserving transformation rule, $rPJS^{-1}$, is defined to process the sequential tasks. An inverse-protocol/projection-inheritance-preserving transformation rule, $rPPS^{-1}$, is defined to process the loop tasks. Another inverse-projection-inheritance-preserving transformation rule, $rPJ3S^{-1}$, is defined to process the parallel tasks. IPR is designed based on the above rules.

3.3. Extension of Process Description

To guarantee the validity of business functions, IPR extends the transition attributes in WF-net on its *transition label*, *significance* and *weight*.

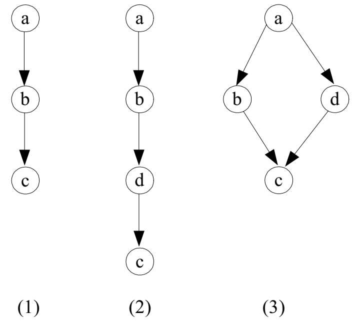

Figure 4. Schematic diagram of the transformation methods

The value of *transition label* is a triple <*sender, receiver, message*>, where *sender* and *receiver* represent the role names of the message sender and the receiver respectively, and *message* represents a message exchanged between partners.

The attribute *significance* is a Boolean: TRUE means this transition is vital and irreconcilable, and FALSE implies the transition is optional and reconcilable, depending on the business model of partners. For example, the significant value of (v, c, *orderConfirm*) in Fig. 2 can be FALSE if neglecting this activity has little influence on the business function. In contrast, the significant value of (c, v, *order*) must be TRUE because neglecting this activity results in failure of the business function.

The attribute *weight* is a non-negative integer to indicate the importance degree of transitions. In other word, a transition with a heavier *weight* means deeper impact on the process.

3.4. Matching Degree and Threshold

IPR uses *MatchDegree* and *threshold* to evaluate how "appropriate" a partner is.

Definition 3: Matching Degree is given by

$$MatchDegree = \frac{\sum_{\{t \mid l(t) \in L_{recResult}\}} t.weight}{\sum_{\{t \mid l(t) \in L_{requester}\}} t.weight},$$

Where $L_{recResult}$ denotes the alphabet of the reconciliation result, $L_{requester}$ denotes the alphabet of the requester interaction process, t denotes the transition of corresponding process and *t.weight* is the value of transition t's weight. Thus, *MatchDegree* is actually the ratio of the total weight of all translations in the reconciliation result to that in the requester interaction process.

The *threshold* is used to guarantee the matching degree and improve the reconciliation efficiency. It ranges between [0, 1], depending on specific strategies of companies. When the *matchDegree* is larger than the threshold, IPR supposes an appropriate partner has been found for the requester.

3.5. The Algorithm of Reconciliation Thread

To implement the reconciliation, IPR uses WF-net as the formal description of the interaction processes. After a service requester initiates the approach with input parameters, including the description of the desired service, the requester interaction process, and the threshold, IPR queries the providers on their service types. Upon obtaining interaction process descriptions, IPR creates a reconciliation thread for every provider with input parameters including the interaction process descriptions $PN_{requester}$, $PN_{provider}$ and *threshold*. Thus, as shown in Table 1, a reconciliation thread is executed between the requester and each provider.

The output of a thread includes reconciliation result, $P_{interaction}$, the modification of the requester, $L_{\triangle requester}$, the modification of the provider, $L_{\triangle provider}$, and the matching degree, *MatchDegree*, of the provider. In this algorithm, the *threshold* is introduced to guarantee the matching degree and improve the reconciliation efficiency, where the potential matching degree is calculated before any modification to the interaction processes. Only the processes with potential matching degrees lager than the *threshold* are allowed to enter into the next steps.

Table 1. Algorithm of reconciliation thread

Input: $PN_{requester}$, $PN_{provider}$, *threshold*

Output: $P_{interaction}$, $L_{\triangle requester}$, $L_{\triangle provider}$, *matchDegree*

Procedures:

Step 1, compute the intersection of requester alphabet and provider alphabet.

$L_{requester} = \alpha(PN_{requester})$; $L_{provider} = \alpha(PN_{provider})$;

$L_{intersection} = L_{requester} \cap L_{provider}$;

Step 2, evaluate the acceptability of the changes according by checking whether all vital transitions have been included in the intersection.

Step 3, compute the potential matching degree

$$prematch = \frac{\sum_{\{t \mid l(t) \in L_{intersecti\ on}\}} t.weight}{\sum_{\{t \mid l(t) \in L_{requester}\}} t.weight};$$

If (*preMatch-threshold*<0){return (null,0,null,null); }

else{continue to step 4;}

Step 4, project the respective interaction process of requester and provider according to the intersection by means of $rPTS^{-1}$, $rPPS^{-1}$, $rPJS^{-1}$ and $rPJ3S^{-1}$ mentioned above.

$P_{interactionR}$ = projection($PN_{requester}$, $L_{intersection}$);

$P_{interactionP}$ = projection($PN_{provider}$, $L_{intersection}$);

Step 5, check the matchmaking of corresponding transitions in $P_{interactionR}$ and $P_{interactionP}$ starting with their initial ones. If there is any unmatched transition, the algorithm fails and terminates with *coefficient* =0. If all transitions are matched, then the algorithm will succeeds and terminates with *coefficient* =1.

Step 6, *matchDegree = coefficient * preMatch*;

If (*matchDegree <= 0*){return (null,0,null,null); }

else{return ($P_{interactionR}$, , $L_{requester}$-$L_{intersection}$, $L_{provider}$- $L_{intersection}$, *matchDegree*);}

■

3.6. Selection Strategy

One key issue that deserves some words in the IPR design is the provider selection strategy. As discussed above, IPR executes a reconciliation thread for each potential provider, and there could be multiple providers with matching degrees larger than the *threshold*. Intuitively, we have at least two choices as follows. (1) IPR selects the first provider with a matching degree larger than *threshold*, which we called FSS (First Service Strategy); (2) IPR always selects the provider with the largest matching degree, named OSS (Optimal Service Strategy). Obviously, there lies a tradeoff between process matching time and quality. We have more discussion on the strategy selection in Section 5.

4. Experiment Methodology

4.1 Experimental Environment

We are currently working on our key project, CROWN (**C**hina **R**&D Environment **O**ver **W**ide-area **N**etwork) Grid [14, 15]. Aiming at integration of resources and cooperation of researchers, the CROWN project was started in late 2003. As illustrated in Fig. 5, a number of universities and institutes, such as Tsinghua University, Peking University, Chinese Academy of Sciences, and Beihang University, have joined CROWN, with each contributing 50-100 computing nodes.

We conduct experiments in the CROWN environment. A prototype IPR server over a PC with Pentium IV 2.4Ghz CPU and a 512M memory is implemented in our lab. The service information and the interaction processes of service providers are stored in an extended service directory, and can be reached via a database in another Pentium IV PC. The interaction processes of the service requester and the providers are established in PNML (Petri Net Markup Language) [6] through PIPE [7] and stored in the extended service directory as well.

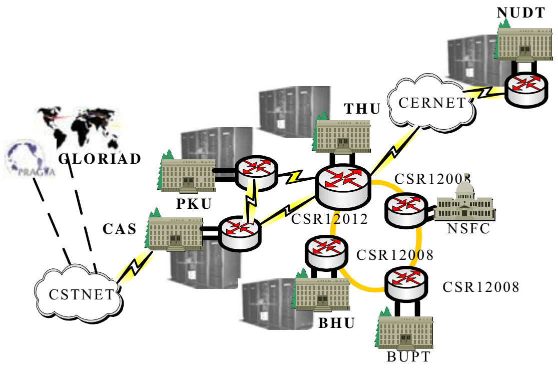

Figure 5. CROWN grid

4.2 Process Generation

We exploit RosettaNet to ensure realistic processes and compatible individual messages. The number of transitions of an interaction process is denoted as *Plen*. Mostly, same type services are always organized in line with the same business standard. However, there are also service providers willing to support special functions beyond the standards to attract requesters. On the other hand, service requesters want to implement individualized functions with appropriate providers. In our experiments, we assume 60% transitions of interaction process following common standards and 40% transitions designed to provide special messages. Furthermore, we assume there is no duplicate label of transitions in an interaction process and the weight of all transitions is set as 1 for the simpleness.

4.3 Metrics

Indeed, existing approaches are the special cases of IPR when the *threshold* of IPR is defined as 1.0, called as Exact Matchmaking (EM). In previous studies, EM is widely employed. In IPR, the *threshold* falls between 0.0 and 1.0, meaning that the reconciler ensures the matchmaking of all vital and optional transitions to be reconciled, which we called fuzzy matchmaking (FM). IPR aims at high matching success rate, short average response time and high matching efficiency.

To better evaluate IPR, we use three major performance metrics as follows.

The *Success rate* is defined as the number of the requests successfully matched with partners over the number of all generated requests.

The *Average Response Time* is defined as the average time of all queries, and the response time of a query is defined as the time period from a requester issuing a query, to the discovery of an appropriate partner with a compatible interaction process. If there is no compatible with the requester, the query ends after searching all potential partners.

The *Matching Efficiency* is given by:

$$\text{Matching Efficiency} = \frac{Na(t)}{Nr},$$

Where $Na(t)$ denotes the number of requests matched with appropriate partners during given time period t, and Nr denotes the total number of all generated requests.

5. Performance Evaluation

IPR is evaluated by comprehensive experiments in terms of *matching success rate*, *average response time*, and *matching efficiency*. We present the experimental results in this section.

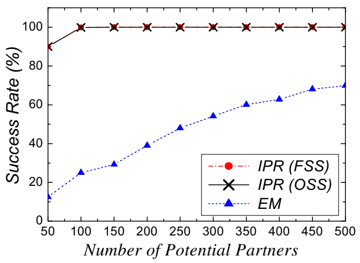

Figure 6. Success rate v.s. number of partners (*Plen*:10)

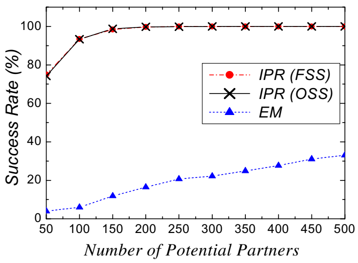

Figure 7. Success rate v.s. number of partners (*Plen*:15)

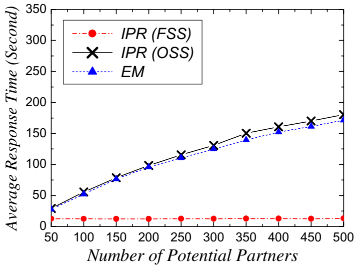

Figure 8. Response time v.s. number of partners (*Plen*:10)

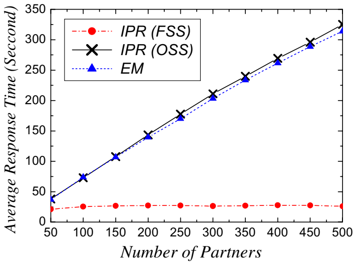

Figure 9. Response time v.s. number of partners (*Plen*:15)

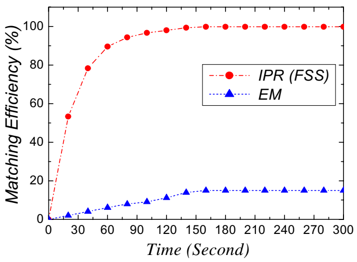

Figure10. Matching efficiency of IPR FSS and EM

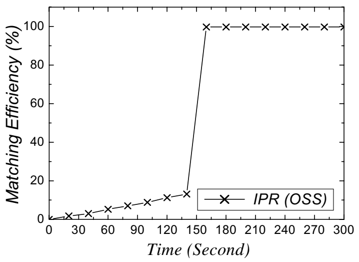

Figure 11. Matching efficiency of IPR OSS

5.1. IPR v.s. Exact Matchmaking

As most of the existing approaches require exact matching (EM), we compare IPR with EM on their success rates and response times in our first set of experiments.

We plot the curves of the success rate when the *threshold* of IPR is assigned as 0.8 and *Plen*=10, 15 in Figures 6 and 7 respectively. Clearly, the success rate of the IPR is more than 80% higher than that of the EM. For example, the success rate of IPR is close to 100% when the system has 100 potential service providers with the same service type, while in EM, the success rate is less than 40% when 500 potential partners are available. Intuitively, due to the diversification of the individualized transitions, it is rather complex to discover a service provider that matches the interaction process exactly. With the aid of reconciliation, EM of the interaction process is substituted with FM, and results in a sharp increment to the success rate.

Figures 8 and 9 plot the average response time of IPR and EM, where the *threshold* of IPR is assigned as 0.8 and *Plen*=10, 15 respectively. We can see the response time of EM increases with lager number of potential partners. In IPR, if we use FSS strategy, the response time is nearly unchanged with different number of partners, but when OSS is employed, the response time significantly grows and even little longer than that with EM because of the reconciliation time.

We also notice that the interaction processes with higher complexities lead to lower success rates and longer response times, because too many interaction messages seriously affect the metrics. Therefore, the partners in e-business should simplify their interaction processes and hide business transactions inside the border of the enterprise as much as possible.

5.2 OSS v.s. FSS

As previously discussed, the selection of FSS and OSS is one of the key issues in IPR. Figures 6 and 7 show that the success rates by using FSS and OSS are similar, while Figures 8 and 9 show that the average response time of OSS is much longer than that of FSS.

The matching efficiencies of IPR (FSS), IPR (OSS) and EM are shown in Figures 10 and 11, where *Plen* is 15 with 200 potential partners. The curves with different number of potential partners and *Plen* are consistent and we show the representative. We find that the matching efficiencies of IPR with FSS and OSS are all high compared with EM. It is noteworthy that the curve of OSS jumps to a high level at a certain time *t* when all candidates have been searched through. The reason is that without finding any exact compatible partner, OSS selects the best one of the candidates whose matching degree is larger than the *threshold*.

Based on the above observations, we have the following suggestions for selection of FSS and OSS: (1) when large amounts of service providers are available for the same service type, such as popular services, the searching space is large, so that OSS may spend too much time. In such a situation, FSS is a reasonable choice. The search quality of IPR with FSS is controllable by tuning the *threshold*. Such choice is also applicable when the interaction processes are too complex. Otherwise, OSS should be employed; (2) if the IPR server is bearing heavy burdens, FSS is always a good choice; (3) if the requester wants to realize as many special requirements as possible, and the response time of query is not important, OSS will be better.

6. Conclusion

The compatibility of partners is of great importance in B2B interactions. In this paper, we propose a practical approach of automated reconciliation, IPR, for inter-organizational interaction processes. By using the inheritance of the WF-net, IPR helps parties to find appropriate partners in a short time and at a high success rate. The attributes, such as *significance* and *weight*, are introduced to validate the reconciliation results.

We implement a prototype IPR server in our lab, and evaluate this design by comprehensive experiments in a real grid system, CROWN. The results show that IPR outperforms existing approaches without losing peer autonomy. We believe that wide deployment of IPR will facilitate inter-organizational interactions worldwide.

7. Acknowledgements

This work was supported in part by China NSFC 90412011, by Hong Kong RGC Grants DAG 04/05 EG01, and by Microsoft Research Asia. The participation of the conference is partially supported by Nokia.

8. References

[1] A. Byde, G. Piccinelli, and W. Lamersdorf, "Automating Negotiation over B2B Processes," Proceedings of the 13th International Workshop on Database and Expert Systems Applications (DEXA'02), 2002.

[2] A. Wombacher, P. Fankhauser, B. Mahleko, and E. Neuhold, "Matchmaking for Business Processes based on Choreographies," Proceedings of International Conference on e-Technology, e-Commerce and e-Service (EEE'04), 2004.

[3] W. v. d. Aalst and M. Weske, "The P2P Approach to Inter-organizational Workflows," Proceedings of the 13th International Conference on Advance Information System Engineering (CAiSE'01), 2001.

[4] W. v. d. Aalst and T. Basten., "Inheritance of Workflows: An Approach to Tackling Problems Related to Change," *Theoretical Computer Science*, vol. 270, 2002.

[5] A. Banerji, C. Bartolini, D. B. A. Chopella, K. Govindarajan, H. K. A. Karp, G.. P. M. Lemon, S. Sharma, and S. Williams. *Web Services Conversation Language (WSCL) 1.0*. http://www.w3.org /TR/wscl10. 2002.

[6] J. Billington, S. Christensen, K. v. Hee, E .Kindler, O. Kummer, L. Petrucci, R. Post, C. Stehno, and M. Weber, "The Petri Net Markup Language: Concepts, Technology, and Tools," Proceedings of the 24th International Conference on Application and Theory of Petri Nets (ICATPN'03), 2003.

[7] J.D. Bloom. *Platform Independent Petri-Net Editor*. https://sourceforge.net/projects/petri-net. 2004.

[8] L. Clement, A. Hately, C. Riegen, and T. Rogers. *UDDI Version 3.0.2*. http://www.uddi.org/uddi-v3.0.2-20041019.htm. 2004.

[9] F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte, and S. Weerawarana. *Business Process Execution Language for Web Services, Version 1.1*. http://www-900.ibm.com/developerWorks/cn/webservices/ws-bpel_spec/index_eng.shtml. 2003.

[10] F. Leymann, D. Roller, and M. T. Schmidt, "Web Services and Business Process Management," *IBM System Journal*, vol. 41, 2002.

[11] Y. Fan and J. Lai, "An Architecture for Cross-organization Business Process Integration," Proceedings of the 5th International Conference on Managing Innovations in Manufacturing (MIM'02), 2002.

[12] H. Foster, S. Uchitel, J. Magee, and J. Kramer., "Compatibility Verification for Web Service Choreography," Proceedings of the 2nd International Conference on Web Services (ICWS'04), 2004.

[13] P. Grefen, K. Aberer, Y. Hoffner, and H. Ludwig, "Crossflow: Cross-organizational Workflow Management in Dynamic Virtual Enterprises," *International Journal of Computer Systems Science & Engineering*, vol. 15, 2000.

[14] C. Hu, Y. Zhu, J. Huai, Y. Liu, and L. M. Ni, "Efficient Information Service Management Using Service Club in CROWN Grid," Proceedings of the 2005 IEEE Conference on Service Computing (SCC 2005), 2005.

[15] J. Huai, Y. Zhang, X. Li, and Y. Liu, "Distributed Access Control in CROWN Groups," Proceedings of the 34th International Conference on Parallel Processing (ICPP'05), 2005.

[16] N. Kavantzas, D. Burdett, and G. Ritzinger. *Web Services Choreography Description Language Version 1.0*. http://www.w3.org/TR/2004/. 2004.

[17] L. Li and I. Horrocks, "A Software Framework for Matchmaking based on Semantic Web Technology," Proceedings of the 12th International Conference on the World Wide Web (WWW'03), 2003.

[18] *RosettaNet PIPs*. http://www.rosettanet.org/pips. 2002.

[19] S. Ren, L. Guo, S. Jiang, X. Zhang: "SAT-Match: A Self-Adaptive Topology Matching Method to Achieve Low Lookup Latency in Structured P2P Overlay Networks," Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS'04), 2004.