Research on Inheritance of Extend Case Transfer Interorganization Workflow Management

LU Fu-rong, DENG Hui-xue and LU Ting-jie

Beijing University of Posts and Telecommunications, Beijing 100876
frhappy@263.net

Beijing University of Posts and Telecommunications, Beijing 100876
xbdhx@163.com

Abstract: The paper mentions Extend Case Transfer Architecture (ECTA) interorganizational workflow management is a fit one to Collaboration Commerce. ECTA modeling has three steps. During the third step, modifying private workflow might cause some vital wrongs, such as dead lock, live lock and dead task. These problems could change the overall workflow’s soundness and efficiency. An approach, which based on Petri net, is present to protect the inheritance of public workflow in private domain, and an instance of C-commerce workflow modeling is shown.

Keywords: C-commerce, interorganization workflows, ECTA, Petri net, WF-net

1. INTRODUCTION

With the explosive growth of the Internet in the last couple of years, electronic commerce is now able to offer solutions for a much broader range of business processes than EDI (Electronic Data Interchange) previously addressed. Business-to-business E-commerce has seen tremendous growth due to the globalization of the worldwide economy. Many corporations are extending their markets by mergers and strategic alliances with other companies throughout the world. The C-commerce (Collaboration Commerce) is presented, which the business processes are key facture to implementing business collaboration[1]. Business processes of each business partner now become coupled in some way, creating interorganizational workflow processes[2].

The Workflow Management Coalition (WfMC) defines some workflow management system as follows[3][4]: Capacity Sharing Architecture (CSA), Chained Execution Architecture (CEA), Subcontracting Architecture (SCA), Case Transfer Architecture (CTA), Extend Case Transfer Architecture (ECTA), and Loosely Couplce Architecture (LCA), in which ECTA is considered more fit to C-commerce. For this form, each business partners has a copy of the workflow process description, and it is possible to allow local variations and extensions. ECTA provide the flexibility and promptness to business chance which based on market
demand. But, there are some potential mistakes to threaten the whole workflow soundness, after partners modify their sub workflows, such as deadlock, livelock, dead task etc\cite{5}.

In following sections, the conception of ECTA, workflow management of ECTA, and private workflow establishing method will be introduced. Then the mistakes will be shown, which produced by private workflow modifying. We will use the Petri net to express the workflow. At last, a theorem will be provided, which protects the inheritance of public workflow. And an instance shows using the theorem to modify the private workflows.

2. EXTEND CASE TRANSFER ARCHITECTURE

For this form, each business partners has a copy of the workflow process description. However, at any time, each case resides at the exactly one location. Case can be transferred from one location to another. However, it is possible to allow local variations, e.g., at a sub model the process maybe extended with additional tasks\cite{3}. It is important that the extensions allow for the proper transfer of cases. This means that the extensions are executed before transferring the case or that there is some notion of inheritance which allows for the mapping of the state of a case during the transfer. The case transferring can be to balance the workload, and tasks are not implemented at all locations, this mechanism make workflows efficient. Figure 1 shows the architecture of extend case transfer.

![Figure 1: Extend Case Transfer Architecture](image)

For all the forms of interorganization workflow, there are two partitioning dimension to distribute the workflow: the case and the process. Vertical partitioning uses the case dimension, i.e., the case are distributed over several business partners but the process is not cut into pieces. Horizontal partitioning is based on the process dimension, i.e., the process cut into pieces and cases are not allocated to specific business partners, just like several business partners may be working on the same case at the same moment in time, ECTA belong to vertical partitioning.

From the view of E-commerce, C-commerce based on the collaboration of business partners, and need more flexible than traditional E-commerce. Extended case transfer is more flexible and prompt\cite{4}, because the public workflow is created by the chance of market demand and it brings the market information to transfer to the partners. The private workflows extension could keep the partners business effective and efficient, farther, form the view of strategy collaboration; it keeps the answer speed to market. For extend case transfer workflow process, cases can be considered as the set of transferring information, a set concern to one case or more cases. A transfer policy is used to determine when to transfer to
which partner. During the collaboration implementing, both information sharing and case results transfer can be finished.

3. THE CHALLENGE OF ECTA

If extend case transfer modeling is used, each partner handle a sub workflow, however, all business partners agreed on the public workflow and division of tasks. The public workflow can be considered as a contract identifying the main steps of the overall process. After agreeing on the public workflow, each of the business partners involved starts to implement its part of the workflow. This local workflow is modified to accommodate local needs, because local autonomy is needed to enable each partner to handle their part of the workflow as effective and efficient as possible. These modifies include adding a causal relation between two tasks, reversing the order of the tasks, etc. Form a local point of view, these changes do not cause any problems, and however, these changes lead to global defections. At following example, we illustrate the problems with the Petri net workflow.

A Petri net is a network composed of squares and circles[6][7]. The squares are called transitions and correspond to tasks that need to be executed. The circles are used to represent the state of a workflow and are called places. The arrows between places and transitions are used to specify causal relations. A place p is called an input place of a transition t if there exists a directed arc from p to t. place p is called an output place of transition t if exists directed arc from t to p. At any time a place contains zero of more tokens, drawn as black dots. The state of the net, often referred to as marking, is the distribution of token over places. The number of tokens may change during the execution of the net. Transitions are the active components in a Petri net.

For the ECTA workflow, there are three steps implication the workflow process:

1) Design the public workflow process. The partners agree on the overall structure of the common business process. The key tasks are identified as well as the interface between them. Figure 2 gives an example for an ECTA Petri Net; it is a public workflow process. The square A to H specify key tasks, and circle p1 to p12 are input or output transitions, of which p6, p9, p10 and p11 are interface transitions between the left and right domain.

![Figure 2. An example of ECTA Petri net](image)

For the ECTA workflow, there are three steps implication the workflow process:

1) Design the public workflow process. The partners agree on the overall structure of the common business process. The key tasks are identified as well as the interface between them. Figure 2 gives an example for an ECTA Petri Net; it is a public workflow process. The square A to H specify key tasks, and circle p1 to p12 are input or output transitions, of which p6, p9, p10 and p11 are interface transitions between the left and right domain.

![Figure 3. The partitioning of public workflow](image)
Partition the public workflow process definition amongst business partners. In this step, the partners are assigned to be responsible for completing parts of the process. The public workflow is mapped on several domains. The partitioning creates a set of WF-net fragments for each partner. Each set contains fragments of the public WF-net that the partner is responsible for. In figure 2, the left and the right domain split by the broken line, and belongs to two partners respectively.

For each business partner create a private workflow. The partitioning stage creates sets that are not necessarily workflow nets, rather fragments of a workflow net. For example, in Figure 3, transition p12 is added in order to make sure task C implement before task B. The behavior is important for left partner.

Steps (1) to (3) are common process of ECTA modeling. However, in step (3), the modifications might cause some vitally wrongs. A public workflow of extend case transfer form is shown in Figure 2, and each partner agreed on this public workflow. This workflow is partitioned over two domains: task A, B, C and D are mapped onto left domain, and tasks E, F, G and H are mapped onto right domain. Each of the two domains is handled by one of the partners. For left domain, if a causal relation is added between task C and task B to make sure that C is executed before B, the condition is shown in figure 3. From a local point of view, this change does not cause any problems. For left domain it is perfectly acceptable to reduce the degree of parallelism. However, this change leads to a global deadlock. On the other hand, C should be executed before B and on the other hand, B should be executed before C (causal relation via p9, p6 and p10). The circular dependency involving tasks C, B, F and G causes the overall workflow process to deadlock in the state marking places p1, p2 and p5. As same, the partners always need modify the subprocess and extend private workflow in order to optimize the process. But it always causes deadlocks (a case gets stuck in some state where it is not possible to execute any tasks.), livelocks (a case trapped in an infinite loop where it is possible to execute tasks but no real progress is possible.), dead task (a task can never be executed for any case)\[5\], etc. to the overall workflow.

4. **HOW TO RESOLVE THE PROBLEMS**

There are many researches about the workflow management and Petri net structure. In fact, these are measures to keep Extend Case Transfer workflow running effective. In [Aalst, Desel, and Oberweis, 8], [Verbeek, Basten, and Aalst, 9] the sound Petri net workflow is defined, and there are some measures could avoid the mistakes which caused by modification in private workflow. The inheritance relations defined in [Aalst and Basten, 10, 11; Basten, 12], and [Basten and Aalst, 13]. Basically, inheritance relations relate many variants of a single process definition using subclass-superclass relationships. Modify private processes using inheritance preserving transformation rules. Here, each business partner will modify their private process created in the previous step to accurately reflect their own business process and incorporate details of how the tasks will be implemented at their organization.

According to researches above, here gives some definition and a theorem, in order to
inherit specialties of the public workflow mode. These rules are based on Petri net and workflow management.

**Definition 1:** There are three rules to add WF-net: iteration, sequential composition, parallel composition. Rule PJ refers sequential composition; rule PJ3 refers parallel composition; rule PP refers iteration.

**Definition 2:** A workflow net \( N = (P, T, F) \) is sound if the following three requirements are satisfied:

a. For any case, it is possible to terminate, i.e., it is possible to reach a state with at least one token in the output place \( o \) (the source place of the WF-net).

b. The moment the case terminates (i.e., a token appears in \( o \)), there are no tokens left behind in the workflow net, this means that there will be no dangling references.

c. There are no dead tasks, i.e., starting with a token in the input place \( I \), it should be possible to execute an arbitrary tasks by following the appropriate route through the WF-net.

**Definition 3:** \( (N_{\text{pub}}) = (P_{\text{pub}}, T_{\text{pub}}, F_{\text{pub}}) \) is a WF-net which represents the public (i.e., common) workflow process definition.

**Definition 4:** \( D \) is a set of domains, i.e., the business partners involved in the public workflow. The function map \( \in T_{\text{pub}} \rightarrow D \) maps each task in the public workflow onto one of the domains.

**Definition 5:** \( (P_{\text{exch}}) \): \( P_{\text{exch}} = \{p \in P_{\text{pub}} | t_1, t_2 \in T_{\text{pub}}: (t_1 \in \bullet p) \land (t_2 \in p \bullet) \land (\text{map}(t_1) \neq \text{map}(t_2)) \} \) is the set of exchange places, i.e., communication channels between domains.

**Definition 6:** \( (\text{inp}, \text{outp}) \). Function \( \text{inp} \in D \rightarrow P_{\text{exch}} \cup \{i\} \) maps every domain onto a local input place and function \( \text{outp} \in D \rightarrow P_{\text{pub}} \cup \{o\} \) maps every domain onto a local output place.

**Definition 7:** \( (N_{\text{epub}}) \), \( N_{\text{epub}} = (P_{\text{epub}}, T_{\text{epub}}, F_{\text{epub}}) \) is a WF-net which represents the extended public workflow process definition.

**Definition 8:** \( (N_{d_{\text{priv}}}) \). For each domain \( d \in D \), \( N_{d_{\text{priv}}}, \text{inp} \), and \( \text{outp} \) such that the following requirements are satisfied, the extended WF-net is a sound one inheriting of public WF-net.

1) \( N_{\text{pub}} \subseteq N_{\text{epub}} \).

2) For all \( p \in P_{\text{epub}} \), \( p \) is implicit in \( (P_{\text{pub}} \cup \{ p \}) \), \( T_{\text{pub}}, F_{\text{epub}} \cap ((P_{\text{pub}} \cup \{ p \}) \times T_{\text{pub}}) \cup ((T_{\text{pub}} \times (P_{\text{pub}} \cup \{ p \}) \times F_{\text{epub}}))) \) for the initial state with one token in input place \( i \).

3) \( N_{\text{epub}} \) is a sound WF-net.

4) For all \( p \in P_{\text{exch}}, | \bullet p | = | p \bullet | = 1 \).

5) For each \( d \in D \), \( N_{d_{\text{priv}}} \) is a sound WF-net.

Now, we give the theorem of inheritance preserving rules.

**Theorem 1:** Let \( N_{\text{pub}}, D, \) and \( \text{map} \) be as defined above. If there exist an \( N_{\text{epub}} \), \( \text{inp} \), and \( \text{outp} \) such that the following requirements are satisfied, the extended WF-net is a sound one inheriting of public WF-net.

1) \( N_{\text{pub}} \subseteq N_{\text{epub}} \).

2) For all \( p \in P_{\text{epub} \cup P_{\text{pub}}}, p \) is implicit in \( (P_{\text{pub}} \cup \{ p \}), T_{\text{pub}}, F_{\text{epub}} \cap ((P_{\text{pub}} \cup \{ p \}) \times T_{\text{pub}}) \cup ((T_{\text{pub}} \times (P_{\text{pub}} \cup \{ p \}) \times F_{\text{epub}}))) \) for the initial state with one token in input place \( i \).

3) \( N_{\text{epub}} \) is a sound WF-net.

4) For all \( p \in P_{\text{exch}}, | \bullet p | = | p \bullet | = 1 \).

5) For each \( d \in D \), \( N_{d_{\text{priv}}} \) is a sound WF-net.

The first requirement states that the extended public workflow should only add places. The second requirement states that these places should be implicit. The third requirement has
been added to make sure that after termination nothing is left in any of the implicit places. Communication channels should be unidirectional is the fourth requirement. i.e., there should be one input transition in one domain and one output transition in another domain. Note that this requirement indicates that there can be no choices at the interface. Any decision is made inside one domain. This does not limit the application since multiple logical communication channels (i.e., place in $p^{\text{exch}}$) can be mapped onto one tangible channel. The last requirement states that each of the mentioned in Definition 2 and Definition 3. Requirement 5 is not as restrictive it seems. Every partitioning of the public workflow will do as long as it is possible to identify a local input and output place and every task is executed in-between the states represented by both places.

An example is given at following statement, we will use the theorem to extend the workflow of partners, and it should shows inheritance preserving. Figure 4 is customizing process of M-business. It is a collaboration environment of telecom operator and Application Service Provider (ASP), the customer customize service via M-commerce platform. The requirement is specified in the Service Level Agreement (SLA) between telecom operator and ASP. Both operator and ASP have identified a set of primary tasks that are involved in providing this service and specified the relations between these tasks. Here, M-commerce environment use the ECTA (Extend Case Transfer Architecture), so, the partners share the common workflow process description. According the common workflow process description, each partner extends workflow in personal domain. We will use the former theorem to extend sub process in order to keep the heritance of common workflow. Figure 4 is the common workflow description of customizing process.

![Figure 4. WF-net of customizing workflow](image)
![Figure 5. partitioning of public workflow](image)

In Figure 4, the left part is process of telecom operator and the right part is of ASP. Step 1, the partitioning of the public process between operator and ASP results shown in Figure 4. The resulting fragments sets for operator and ASP do not form WF-net. For example, the partition for operator is not connected, while the partition for ASP contains multiple output places. The next step is to create two WF-nets from the two partition, one of operator and one for ASP. For ASP, place $p_2$ is assigned to be the input place of the private workflow and place
$p_8$ is assigned to be the output place. For operator, an implicit place $p_{11}$ is added. Also arcs form $p_{11}$ to task inform are introduced, the partitioning of public workflow is in Figure 5.

In step 3, both operator and ASP modify their private processes. The shaded tasks are the newly introduced tasks. Modifications made by operator include the addition of two tasks: task $N$ (Handle Cast) and task $P$ (Option). Task $N$ gives the service guide to user; task $P$ ensures that customer to select proper service, before transfer the requirement to ASP. $N$ is inserted using PP (iteration rule); $P$ is inserted using PJ (sequential composition rule). Also, in ASP domain, there are two tasks: $L$ (classify) and $M$ (judgment). Task $L$ is service search and user management, and it is inserted using PJ3 (parallel composition rule). $M$ is SMS/MMS regular judgment, and it is inserted using PP. Figure 6 shows the final private workflow of operator and ASP. Now, we examine the result using theorem 1.

**Demonstration:**

As we see, WF-net in Figure 1 is defined as $N^\text{pub} = (T^\text{pub}, P^\text{pub}, F^\text{pub})$, WF-net in Figure 2 is defined as $N^\text{epub} = (T^\text{epub}, P^\text{epub}, F^\text{epub})$. $D$ is the set of domain operator and domain ASP.

**step 1:**
$T^\text{pub} = \{A,B,C,D,E,F,G,H,I,J,K\}$, $P^\text{pub} = \{P_1,P_2,P_3,P_4,P_5,P_6,P_7,8,P_9\}$ ; and $T^\text{epub} = \{A,B,C,D,E,F,G,H,I,J,K,L,M,N,P\}$; $P^\text{epub} = \{P_1,P_2,P_3,P_4,P_5,P_6,P_7,8,P_9,P_{10},P_{11}\}$
then $T^\text{pub}$ $\subseteq$ $T^\text{epub}$; and $P^\text{pub}$ $\subseteq$ $P^\text{epub}$ so we know: $N^\text{pub}$ $\subseteq$ $N^\text{epub}$

**step 2:** For all $p \in P^\text{pub}\setminus P^\text{epub}$, every sequence enabled in the net without the $p$, and is also enabled in the net with the $p$. So $p$ is implicit in $(P^\text{pub} \cup \{p\}, T^\text{pub}, F^\text{pub} \cap ((P^\text{pub} \cup \{p\}) \times T^\text{pub}) \cup (T^\text{pub} \times (P^\text{pub} \cup \{p\}))).$

**step 3:** $N^\text{epub} = (T^\text{epub}, P^\text{epub}, F^\text{epub})$ is a sound WF-net, because, in Figure 2, for every case, it is possible to terminate; at the end of case, there will be no dangling reference; for each domain there are no dead tasks. So, $N^\text{epub}$ is a sound WF-net.

**step 4:** for domain operation and domain ASP, each domain has one input transition and one output transition, viz. all $p \in P^\text{exch}$, $|\cdot p| = |p \cdot | = 1$.

**step 5:** For each $d \in D$, viz. domain operator and domain ASP, as content in definition 2, $N^\text{operator}$ and $N^\text{ASP}$ are sound WF-net.

The results shows that projection inheritance is a good basis for designing ECTA...
interorganizational workflows where the business partners only agree on a very basic common workflow process.

5. CONCLUSION

With the trend for business-to-business E-commerce on the rise, support for interorganizational workflows has become increasingly important. The paper only give one condition of interorganizaion management, there are also many other approaches to designing and enacting the interorganization. Some problems will appear during modeling workflow, such as inheritance, private domain autonomy, communication quality. Some researches are difference to Petri net method, which use in this paper. For example, the work conducted in projects such as CrossFlow[14], WISE[15], OSM[16], and COSMOS[17] is highly relevant for the enactment of interorganizational workflows.

REFERENCES


