This is the published version (version of record) of:


Available from Deakin Research Online:

http://hdl.handle.net/10536/DRO/DU:30005759

©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Copyright : 2005, IEEE
Managing Collaborative Business Process in Knowledge-intensive Environment

Shang Gao, Zili Zhang
School of Information Technology
Deakin University
Waurn Ponds, Geelong 3217 Australia
+61 3 5227 1383
{shang, zzhang}@deakin.edu.au

Igor Hawryszkiewycz
Faculty of Information Technology
University of Technology, Sydney
PO Box 123 BROADWAY 2007 Australia
+61 2 9514 1809
igorh@it.uts.edu.au

ABSTRACT

Business processes especially those in knowledge intensive environments often emerge rather than following predefined steps. Supporting emergent processes is one of the key issues for collaborative knowledge sharing. This paper first introduces a component-based workspace metamodel used to support emergent processes. A loose-coupled collaborative process management model WorkPath is proposed based on the workspace structure to support flexible process evolution management and coordination among processes. Key elements that construct workspace and WorkPath, such as role, action, artifact, workspace and reference relation, are described in detail. An implementation prototype and future work is also discussed at the end of the paper.

1. Introduction

Global competition is urging business enterprises to optimize the way they do business and use new technology to support evolving business processes, especially those in knowledge intensive environments. Traditional workflow is usually suitable for well defined and repeated production process, but not the less well-understood and more fluid emergent process, such as the creation of new tasks, identification of goal changes or simply finding better ways to do something.

Emergent processes are processes that emerge rather than following predefined steps, which have been described in many ways. Dourish [1] described it as the opportunistic creation of new parallel task that must be coordinated to a common goal. Hatori and others [2] [3] described communities that work towards common goals with personal workspaces connected to community workspaces. Hoffman [4] for example distinguished between rules based processes and interactive processes. Among these methods, workspace concept proposed by Biuk-Aghai and Hawryszkiewycz [5] [6] captured the main elements of an emergent process: roles, actions, artifacts, and their relationships with each other.

In the workspace based emergent process model [6] [7] [8], dynamic structures, called workspaces, allow continuous modification and evolution and thereby support their users to turn them into places for collaboration. The aim of these workspaces is to support emergent processes which require a high degree of flexibility.

In a knowledge-intensive environment, a business process may be matched to several collaborative workspaces with simple role abilities, or few bigger workspaces, each of which has more complex role definitions. Usually we let the nature of process decide the boundaries of workspaces. For instance, a product development project involves requirement analysis, product design, product development, advertising, marketing, and product maintenance phase, each of which may correspond to one individual workspace. It is also possible to combine product design and development into one single workspace, and advertising and marketing into another single workspace. How to divide a business process into workspaces to achieve higher efficiency is beyond the
scope of this paper. Here we mainly focus on modeling and managing emergent collaborative process from technical point of view.

The workspace metamodel mainly focuses on the evolutionary event processing functionalities within a workspace, but not the collaborative relation and information sharing between workspaces. What we are interested in is to find a good way to improve the coordination between workspaces, while at the same time to benefit from the existing flexible event processing functionalities. This is also the initial impulse of proposing such a workspace-based collaborative process management model WorkPath.

The whole paper is organized as follows: Section 2 and section 3 introduce background knowledge of emergent collaborative process and workspace metamodel. Section 4 proposes a WorkPath model designed to support the evolutionary event trigger between workspaces. In Section 5, WorkPath coordination issues are discussed. And finally in Section 6, we draw a conclusion and list our future work.

2. Background

When we initiate a project with a goal, usually we divide its process into several steps and match each step to an individual workspace. These workspaces are interdependent, each of which aims at accomplishing a sub-goal. From start to finish, we need to deal with process evolution and coordination problem within and among these workspaces. These two issues have to be addressed while modeling and managing emergent collaborative process:

1) Evolution
   Evolution represents the emergent aspect of business process. At the very beginning, little may be known about emergent process except for the general role-action structure. Later, however, as business process is evolving, more requirements emerge and more artifacts need to be added to workspaces. The interdependency among these artifacts is becoming complex in the workspaces, so do the relationships of all the related workspaces.

2) Coordination
   As discussed above, a workspace might only involve a few roles, act on limited artifacts, and take a few steps in a series of procedures of a business process. With each of which fulfilling limited responsibilities, workspaces are usually interdependent to achieve one common goal. These workspaces need to coordinate with each other, communicating the accomplishment of tasks, requesting prerequisite artifacts, or activating interdependent actions.

   Given multiple tasks represented by workspaces attempt to coordinate to achieve one common goal, several issues arise:
   
   • How to ensure these workspaces to coordinate each other properly and the assigned tasks to be finished timely?
   • Furthermore, if an artifact required by multiple collaborative workspaces is modified in one workspace, how could the relevant workspaces be notified of this modification timely?

   Based on the research of workspace structure and the working mechanism of workflow, we propose a workspace-based business process management model – WorkPath, which is expected to bring more flexibility in managing the evolution and coordination of business process without breaking the promise of supporting emergent process.

   For a better understanding of WorkPath model, we explain the key concepts of workspace metamodel first, which will be further used in the WorkPath.

3. WorkSpace metamodel

Despite the diversity of functionality and appearance, an emergent process has four key concepts in common: people, artifact, action and workspace. Their relationship can be described as people with roles aiming at goals act on artifacts in workspaces.

1) Role
   A role refers to an organizational entity involved in a workspace. Role can be assigned to one or multiple participants in a workspace. The role assignment to a participant links the personnel to the organizational action performer. For instance, people taking “project manager” role need to fulfill the responsibilities of a project manager.

2) Action
   An action is a user-defined performance, which is executed by a role. Editing task, initiating a meeting, sending a notice message are all valid actions. Obviously role responsibility limits the manipulation scope of actions.

3) Artifact
Artifact is an object which is consumed or produced in an action, such as a MS Word document, an Excel form, a Gif or JPEG file, a URL link, and a meeting object. Different roles have different permissions to manipulate different properties of a given artifact.

4) Workspace

A workspace represents a working space for one or part of business process, where a collection of roles, actions and artifacts are organized and working logically.

A set of metamodel concepts [7] [8] has been developed to abstract and describe the emergent semantics as shown in Figure 1. Each of the rectangles in the diagram represents an abstract component.

Component User and Group reflect the people factor. A person, usually called a user, ‘is-in’ a group, which can evolve independently, and can contain subgroups. Groups support scalability as independent groups can exist in the same system but gradually merge or intersect if needed.

Component Role reflects the relationship between people and action. Each role is ‘occupied’ by a group whose users take on role abilities. Roles are given permissions to access artifacts and ‘can-take’ the actions. Actions ‘consume or produce’ artifacts. They can be solo-actions, which are taken by individuals, or interactions, such as discussion or meeting where more than one user are involved. Actions ‘use’ tools embedded in workspace.

Component Artifact reflects the operating object and resource factor. An artifact may consist of other follow-up artifacts. Typical artifact examples include “containers” like workspace (which is also considered as an artifact), task, meeting; calendar; folder, forum, and “singles” like URL, text, uploaded file, and chatroom.

Component Workspace represents a working unit in a business process, which can be made up of a number of sub-workspaces as indicated by the “contains” looping arrow. A workspace ‘has’ any number of roles and ‘contains’ any number of artifacts.

Associated with these concepts are commands and facilities that are used to create object instances of the above concepts and assign permissions to people in different workspaces or groups.

A role-based permission control mechanism is adopted in this metamodel. In an emergent process, a user’s roles can be updated to adapt to new situations. Every component except Role has different permission restrictions for different roles. For example, a workspace provides commands to create artifacts, to add new members, to create and edit roles, and to assign new actions to users. A manager role can create a new task and assign it to his team members. Team members usually can only view their own tasks and update limited properties of the tasks, such as task progress, but not the tasks’ main contents.

To facilitate future description, formalized definitions of the above concepts are given as follows:

Assume four sets $N$, $P$, $A$ and $E$ be the organizational entity names, participants, artifacts and operations, respectively.

**Definition 1** A role is a duple consisting of an organizational entity name and a subset of participants. The set of role is

$R = \{ r \mid r = (n, P_r) \in (N \times 2^P) \}$,

where $n \in N$, $P_r \subseteq 2^P$.

**Definition 2** An action is a quadruple consisting of a subset of operations $E_t$, a subset of roles $R_t$, a subset of artifacts $A_{input}$ as input and a subset of artifacts $A_{output}$ as output. The set of action $T$ is

$T = \{ t \mid t = (E_t, R_t, A_{input}^t, A_{output}^t) \in (2^E \times 2^R \times 2^A \times 2^A) \}$,

where $E_t \subseteq 2^E$, $R_t \subseteq 2^R$, $A_{input}^t, A_{output}^t \subseteq 2^A$.

**Definition 3** A workspace is a triple consisting of a subset of roles $R_w$, a subset of artifacts $A_w$ and a subset of actions $T_w$. The set of workspaces $S$ is
\[ S = \{ s \mid s = (R_s, T_s, A_s) \in (2^R \times 2^T \times 2^A) \} , \]
where \( R_s \subseteq 2^R \), \( T_s \subseteq 2^T \), \( A_s \subseteq 2^A \),
\[ A_s = A_{input}^s \cup A_{output}^s . \]

4. WorkPath model

It is common that an artifact is required by more than one workspace in a business process. Like a market analysis report from sales workspace can also act as one reference of requirement analysis report in a requirement analysis and new proposal development workspace. Such kind of artifacts is considered as “key artifacts” because they work as inputs of some particular actions in other related workspaces. They build the reference relations among interdependent workspaces.

Based on whether or not they are referenced by other workspaces, artifacts in one workspace can usually be divided into two categories: one category is for self-use and another is for shared-use or both. It is the key artifacts that combine the related workspaces together and push information to flow in a business process. They act like tokens in Petri Net that trigger enabled transitions and serialize business process steps to accomplish a common goal.

Assume \( A_{output}^s \) is the set of output artifacts produced in workspace \( s \). \( A_{input}^s \) is the set of input artifacts required in workspace \( s \).

**Definition 4** For any two workspace \( s, s' \in S \), if \( \exists a_k \in A_{output}^s \cap A_{input}^{s'} \), then artifact \( a_k \) is called a key artifact to workspace \( s' \) of workspace \( s \), denoted \( a_k^s \).

\( a_k^s \) represents an output of workspace \( s \), which is also an input of action(s) in workspace \( s' \). Key artifacts are glue adhering related workspaces together. Workspaces become interdependent with each other because of the existence of key artifacts. We use a binary relation \( \rightarrow \) to describe the artifact reference relation between workspaces.

**Definition 5** For any two workspace \( s, s' \in S \), \( s \) and \( s' \) have a reference relation \( f : s \rightarrow s' \), if and only if \( \exists a_k \in A_{output}^s \cap A_{input}^{s'} \), denoted \( f_{s \rightarrow s'}^s \), \( f_{s \rightarrow s'}^s \in F, F \subseteq S \times S \).

We say workspace \( s \) is referenced by workspace \( s' \) or workspace \( s' \) references workspace \( s \). Similar reference relation also exists inside a workspace. But this kind of internal reference relation is not identified with assumption that this internal relation can be handled well inside a workspace.

In a workspace, only the key artifacts communicate with other workspaces. They are necessary resources for these referencing workspaces to conduct their actions fluently. Acquiring them on time is one of the prerequisites for these workspaces to realize their sub-goals.

The key artifacts flowing along the reference relation \( \rightarrow \) between workspaces serialize the referencing actions in some degree. The key artifact flow enables these actions to be triggered accordingly. All the related workspaces form a queue based to their finished time. The queue can be considered as one execution path of a task/goal. Of course there may exist many possible execution paths among these workspaces for one goal. All these execution paths build a work path space, called WorkPath.

**Definition 6** A duple \( WP = (S; F) \) is called a WorkPath if it satisfies the following two conditions:

1. There exists at least one workspace \( s_0 \) such that \( s_0 \) does not reference any other workspaces via the reference relation \( f \rightarrow \), or the set of start workspaces \( S_{start} = \{ s \mid \exists s', s \rightarrow s, s \in S, s' \in S \} \neq \Phi \)

2. An end workspace set \( S_{end} = \{ s \mid \exists s', s \rightarrow s', s \in S, s' \in S \} \neq \Phi \)

These conditions mean that there exists at least one such workspace that does not be further referenced by any other workspaces.

**Definition 7** In WorkPath, a workspace sequence \( s_0, s_1, \ldots, s_i, \ldots, s_n \) is called a work path, if it satisfies \( s_0 \in S_{start} \), \( \forall s_i, \exists f_j : s_i \rightarrow s_j \in F \)

\( \land s_i \in S_{start} \lor s_i \in S_{end}, i = 1, \ldots n, j = 1, \ldots n \).

This definition reflects the diversity of workspace executive sequences. If \( s_i \in S_{start} \), \( s_i \) can appear at any position in the path because it never has any key artifact input constraints. If \( s_i \in S_{end} \), \( s_i \) can appear...
at any position except the first place, as long as its input artifacts have been satisfied. Or else there must exist at least one reference relation between any two workspace $s_i$ and $s_j$, which help $s_j$ get required key artifacts from $s_i$ via reference relation $f_{ij}$.

In WorkPath, all workspaces are trying to execute their actions parallely. Only when key artifacts are needed to further the execution, will they be waiting for the particular resources, interacting with related workspaces and keeping going after being satisfied.

Figure 2 demonstrates a four-workspace WorkPath. $s_1, s_2, s_3, s_4$ represent requirement analysis workspace, system design workspace, system development workspace and system installation and maintenance workspace, respectively. $s_1$ works out “requirement analysis report” and “product specification” for $s_2$. $s_2$ provides “design report” for $s_3$. $s_3$ gives “feedback and adjustment requests” to $s_2$. All these three workspaces all provide support to actions in $s_4$, for instance, providing product description, writing administration manual and programmer manual, etc.

Sometimes it is hard to identify the executive sequence clearly as most of them may occur or finish parallel. For instance, the work path may appear like the following sequence shown in Figure 3.

5. Coordination among Workspaces

Whether or not there has a valid work path among the workspaces depends on many issues. The four basic elements workspace, action, role and artifact, affect the sequences of a work path.
Case (i) obviously has much interaction between the two workspaces. Each workspace has to wait for each other intermittently until the required key artifacts arrive. Case (ii) has less interaction because the actions requiring frequent information exchange are mainly located in the same workspace, making it more easier to coordinate actions, hence, reducing communication cycle times. Case (iii), whereas, introduces a deadlock during task execution, where both $t_1$ and $t_4$ requires the output key artifacts interdependently. Human intervention undoubtedly is a good way to resolve deadlock.

Definition of roles and participant grouping may change the workspace structure, therefore, affects the executive sequence of a work path. Undefined responsibility or unqualified participants would delay work progress, which is usually an issue of resource configuration and will not be discussed here in detail.

A work path is adjusted accordingly in an evolutionary business process when:

1) Task update, e.g. adding new tasks, deleting old ones or dividing one into several sub-tasks.
2) Resource relocation, e.g. re-defining role, adding new participant, moving artifacts or participant from one workspace to another one.
3) Business process redesign, e.g. adding or deleting a workspace, merging workspaces, or subdividing workspaces.

For instance, a new workspace can be indeed added after all the existing workspaces know its expected output. A workspace can be removed only after there have no other workspaces referencing its output key artifacts. The coherence and continuity of work path can be kept well if and only if the key artifact reference relation is fluent, regardless of what kind of changes there would be.

To realize coordination, we need to maintain the reference relations between workspaces, or say, we need to know attributes such as a key artifact’s source workspace, destination workspace and related roles in each workspace. Once the relations are clear, different business management strategies can be applied to manipulate the relations. For instance, we maintain a global key artifact reference table to realize coordination. If we set up a new workspace, we first let existing workspaces know its expected output, then add its required key artifacts in the global key artifact reference table. Other workspaces requiring its output can add new entries to the reference table.

If we intend to improve the priority of “system development” workspace, all its referencing key artifacts have higher priority than other artifacts to be finished in the related workspaces. Involved roles are reminded by system to perform all the relevant actions preferentially.

A good work path usually has the following characteristics:

1) Fluent reference relation between workspaces.
2) Less interaction between workspaces. Existence of too many key artifacts usually means substantial waiting time and potential risk of deadlock.
3) Highly parallel execution of workspace actions.
4) Reasonable resource allocation, e.g. participant grouping, role definition, and task assignment.

Techniques and tools used in workflow [10] [11] [12] can be referred to detect the fluency of key artifact flow, for instance, deadlock detection in workflow management, which is also one of our future directions.

6. Model implementation prototype and future work

A successful component-based emergent process management system – LiveNet4 – has proved the flexibility of the workspace metamodel and been applied successfully in a number of domains [6] [7] [8] [9]. Java Enterprise Bean (EJB), Java Server Pages (JSP) and Sybase relational DBMS are used to construct the metamodel’s key concepts and to implement emergent business process management logics. Prototype system can be accessed at http://livenet4.it.uts.edu.au.

In LiveNet4, artifacts and roles are added and updated using embedded system tools as the business process evolving. The coordination for knowledge sharing has been implemented. LiveNet4 prototype supports two types of sharing: one is “artifact link copy and paste”; the other is “artifact copy and paste”. In the former situation, a key artifact only exists in its original workspace where it is created. It appears as a shortcut or copied link in all the other referencing workspaces. Users can freely update the key artifact anywhere if allowed. However, an update made at one place will directly affect all the appearances of that key artifact. In the latter situation, a key artifact is “really” copied. Any changes made only affect its local appearance but not others. Some screen shots (Figure 5, 6, 7 and 8) of LiveNet4 illustrating the artifact copying and pasting procedure are attached.
Tasks assigned to a role are highlighted once that role-taking user logs on the system. Notification is automatically triggered by time sensitive artifacts, which is quite helpful for activating time sensitive actions within or cross workspaces. Key artifact reference management is accomplished by maintaining a global reference relation table of key artifacts in database. Modification of “link copied” key artifacts is instantaneously passed on to all the affected workspaces.

Compared to traditional Workflow, WorkPath is a loose-coupled flexible process management model which not only supports the evolution of emergent business process, but also realizes coordination among interdependent workspaces in some degree. In an emergent process, nothing is initially well defined. We never know what would happen at the very beginning of a process. As the process emerging, things are getting clearer and WorkPath is constructed step by step. Adjustment of reference relation dynamically changes WorkPath, also brings the flexibility to process management.

Awareness is maintained well in WorkPath with the notification, global management mechanism of key artifact reference and other awareness support facilities provided by LiveNet4. Its basic work unit is workspace where some ‘issues’ like action execution, internal artifact management have been ‘absorbed’ and been solved well with built-in tools/functionalities. At the WorkPath level, only reference relations need to be maintained. The separation of detailed artifact manipulation and collaborative interdependency management simplifies the overall process management and coordination work.

In the WorkPath model, the most important issue is not the executive sequence of workspaces, but the reference relations among workspaces and the fluency of the relations. Unlike one step waiting for preceding
steps in a workflow system, maximum of parallel execution is achieved by encouraging workspaces to work parallelly, unless key artifacts are required.

Java component based implementation also enables the model to be easily expanded and be transplanted to other platforms.

Based on the workspace structure and WorkPath model, we are planning to further our research in adaptive key artifact re-allocation and reference relation checking by means of high-level Petri Net (http://www.win.tue.nl/~wsinwa, http://citeseer.nj.nec.com/), for instance, use timed Petri Nets to dynamically change the flow direction of key artifacts along reference relation between workspaces.

7. Acknowledgements

The author would like to thank a number of people contributed to the work described in this paper, especially Dongbei Xue, Luke Cole and Andrew Johnston, for their contributions in the development of LiveNet4.

The work of LiveNet4 was funded by the CRC for Distributed Systems Technology (DSTC), Australia.

8. References


