5-26-2012

Change Support in Cross-Organizational Dynamic Process-Aware Software Architecture – A Pattern-Based Analysis

Sien Chen  
School of Management, Xiamen University, Xiamen, Fujian, CHINA

Zhenyu Liu  
School of Management, Xiamen University, Xiamen, Fujian, CHINA, zhenyliu@xmu.edu.cn

Follow this and additional works at: http://aisel.aisnet.org/whiceb2011

Recommended Citation  
http://aisel.aisnet.org/whiceb2011/48

This material is brought to you by the Wuhan International Conference on e-Business at AIS Electronic Library (AISeL). It has been accepted for inclusion in Eleventh Wuhan International Conference on e-Business by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
Change Support in Cross-Organizational Dynamic Process-Aware Software Architecture – A Pattern-Based Analysis

Sien Chen1, Zhenyu Liu2*

1School of Management, Xiamen University, Xiamen, Fujian, CHINA
2School of Management, Xiamen University, Xiamen, Fujian, CHINA

Abstract: Process-aware information systems (PAIS) offer promising perspectives in this respect and are increasingly employed for operationally supporting business processes. In this paper, we describe the emergence of different process support paradigms and the lack of methods for comparing existing change approaches have made it difficult for process-aware software architecture (PASA) engineers to choose the adequate technology. A pattern-based analysis combines self-adapting and self-evolution theory in PAIS, we adopt a set of changes patterns and change support features to put forwards four kinds of model of PASA according to the situation of the needs business processer facing and changeable environment. Based on these change patterns and features, we provide a detailed mechanism analysis and case study evaluation in the healthcare industry of the relationship between cross-organizational dynamic process-aware software architecture (CD-PASA) and change patterns of business processes. In summary, we identified change patterns and change support features facilitate the comparison of change support frameworks, and consequently will support PASA engineers in selecting the right technology for realizing flexible PASA. In addition, this work can be used as a reference for implementing more flexible PASA.

Keywords: process-aware, business processes, relationship, self-adaptive, software architecture

1. INTRODUCTION

In today’s dynamic business world the economic success of e-business enterprises increasingly depends on its ability to react to changes in its environment in a quick and flexible way [1]. Causes for these changes can be manifold and include the introduction of new laws, market dynamics, or changes in customers’ attitudes. For these reasons, e-business enterprises have recognized business agility as a competitive advantage, which is fundamental for being able to successfully cope with business trends like increasing product and service variability, faster time-to-market and business-on-demand. A Process-aware information system (PAIS) is a work system that supports the delivery of products and/or services to customers by processing information on the basis of explicit process models [2]. Meanwhile to provide effective business process support, flexible PAIS are needed which do not freeze existing business processes, but allow for loosely specified processes, which can be detailed during run-time. In addition, PAIS should enable authorized users to flexibly deviate from the predefined processes if required (e.g., by allowing them to dynamically add, delete, or move process activities) and to evolve business processes over time.

1.1 Problem Statement

Process-aware software architecture (PASA) is to support managed e-business processes. E-business enterprises increasingly adopt PASA, which offers promising perspectives for more flexible enterprise computing. However, there is still a significantly lack of methods for systematically comparing the change frameworks provided by existing process support technologies. The emergence of different process support paradigms and the lack of methods for comparing existing change approaches have made it difficult for PASA
engineers to choose the adequate technology\textsuperscript{3}. This makes it difficult for PASA engineers to assess the maturity and the change capabilities of those technologies, often resulting in wrong decisions and expensive investments, especially for cross-organizational process-aware software architecture. Whether it is commercial or academic research is the lack of cross-organizational process-aware software architecture. Therefore, we need to provide a detailed mechanism analysis of the relationship between PASA and change of business processes. It consequently will support PASA engineers in selecting the right technology for realizing flexible PASA. In addition, this work can be used as a reference for implementing more flexible PASA.

1.2 Related Work

Process-aware information systems are the core of an ongoing trend, and it has drawn the attention of e-business systems engineers and managers shift from data and objects to the processes that the information system and the intra-organizational or cross-organizational environment in which it operates—is intended to enact, enable or support\textsuperscript{4}. This trend has resulted in a myriad of approaches to support the analysis, design, implementation, execution and maintenance of e-business information systems, ranging from those supported by groupware\textsuperscript{5} to those supported by workflow management systems\textsuperscript{6} and more recently business process management systems\textsuperscript{7}. We subsume such different information systems with a process focus on the basis of PASA.

Process awareness has emerged as a guiding principle not only in the design and analysis of information systems, but also as a management discipline in its own right. Indeed, the improvement of business process is a perennial top priority of chief information executives\textsuperscript{8}. When they are seeking to increase compliance or foster business innovation, re-design business process to improve operational efficiency, managers increasingly turn to Business Process Management (BPM)\textsuperscript{9} – a body of principles and methods that support the modeling, analysis, design and implementation of PASA. The appeal of BPM lies in its promise to create sustainable high-performing processes that demonstrate strategic strengths such as the ability to respond better to rapid change or to standardize operational best practices across business units or locations.

The e-business systems engineering community has concentrated on PASA analysis, design and implementation, using for example case handling systems\textsuperscript{10}, workflow technology\textsuperscript{11}, business process management systems\textsuperscript{7} or service-oriented architectures\textsuperscript{12}. The e-business systems management community has focused on the impact of information systems technology to support process-oriented organizations\textsuperscript{13}, the usage of process-aware technology\textsuperscript{14}, process modeling methods\textsuperscript{15,16} and process redesign practices\textsuperscript{17}, or the management of cultural and organizational change to enable process improvement\textsuperscript{18}.

Above the principles and methods that support the self-adaption and self-evolution theory of PAIS. Through principles and method solve the emergence of different process support paradigms and the lack of methods for comparing existing change approaches.

1.3 Approach

1.3.1 Self-Adaption Theory

PASA together with service-oriented computing offer promising perspectives in this respect and a growing interest in aligning information systems in a process- and service-oriented way can be observed\textsuperscript{19}. In contrast to data- or function-centered information systems (IS), PASA are characterized by a strict separation of process logic and application code. In particular, most PASA describe the process logic explicitly in terms of a process model providing the schema for process execution. Usually, the core of the process layer is build by a process management system (PMS), which provides generic functionality for modeling, executing, and monitoring processes. This approach allows for a separation of concerns, which is a well established principle in computer
science for increasing maintainability and reducing cost of change \cite{20}. In many cases changes to one layer can be performed without affecting the other layers. Therefore, separation of concerns \cite{25, 26} is an important concern of self-adaption \cite{21, 22}, self-adaption theory include process orchestration and process choreography \cite{23}, separation of concerns \cite{25, 26}, cross-organizational information sharing-and-hidden \cite{24}.

- Process choreography \cite{27} from the global perspective describes the business processes involved by each other in collaborative peer message exchange.
- Separation of concerns \cite{25, 26} means when the system needs to evolve, simply modifying the process model without modifying the underlying application. For example, if the original process model in order to perform parallel execution of two activities, the change does not need to bind to the active entities on the implementation of changes.
- Cross-organizational information sharing-and-hidden \cite{24} uses the process-view approach for cross-organizational PASA design. A process-view is an abstraction of an implemented process. E-business enterprise can design various process-views for different partners based on diverse commercial relationships and, in doing so, establish an integrated process that consists of internal processes and process-views provided by each partner. Participatory enterprises can obtain appropriate progress information from their own integrated processes, allowing them to collaborate effectively. Cross-organizational PASA is coordinated through virtual states of process-views. As long as the implementation of the activities of the entity meets the interface requirements, binding entity to follow the implementation of activities of information hidden the principle, implementation and evolution of the entity’s internal implementation process and will not affect the process instances.

### 1.3.2 Self-Evolution Theory

Process should have the ability of self-evolution \cite{27} include static evolution, dynamic evolution or dynamic change (dynamic evolution include ad-hoc evolution and evolutionary evolution), dynamic refinement. It contains some components to efficiently check whether the process instances can be migrated and guarantee that the process instances can be correctly migrated to the modified process model.

Through the above described self-adaption \cite{21, 22}, self-evolution \cite{27}, in principle, PAIS facilitate changes significantly. A pattern-based analysis \cite{28, 29} combines self-adapting and self-evolution theory in PAIS, we adopt a set of changes patterns and change support features to put forwards four kinds of model of PASA according to the situation of the needs business processer facing and changeable environment.

Different models require different features to support. Based on these change patterns and features, we provide a detailed mechanism analysis and case study evaluation in the healthcare industry of the relationship between cross-organizational dynamic process-aware software architecture (CD-PASA) and change patterns of business processes. We give the structural framework of each part by describing CD-PASA, meanwhile introduce self-adaptive implementation under the mode of component and explain the representation of component and self-adaptive evolutionary processes. In summary, we identified change patterns and change support features facilitate the comparison of change support frameworks.

Section 2 summarizes a set of changes patterns and change support features to put forwards four kinds of model of PASA according to the situation of the needs business processer facing and changeable environment. Section 3 describes change support in cross-organizational dynamic process-aware software architecture. Mechanism analysis the relationship between cross-organizational dynamic process-aware software architecture and change patterns of business processes is described by Section 4. Section 5 deals with evidencing the existence relationship between change patterns and cross-organizational dynamic process-aware software architecture in healthcare industry. We conclude with a summary and outlook in Section 6.
2. PROCESS CHANGE PATTERN AND CHANGE SUPPORT FEATURES

This section it is to complement existing workflow patterns with a set of change patterns and change support features suitable to assess a PASA ability to effectively deal with process changes. Respective patterns and features should not only allow PASA engineers to assess the expressiveness of a PASA change framework, but also ensure that changes can be demonstrated in a correct, consistent and efficient way.

2.1 Pattern Description

The business process is a set of logically related business activities, which according to the order of execution, and business activities in support of software architecture, to complete specific business tasks. There are a variety of changes in business processes, can have a variety of classification methods, the variability of the process we have to analyze from the perspective of system design. Changes in business processes, mainly the changes of the control structure and business activities, so we will change in characteristics of business processes are divided into the following categories: business activity, business rules, business logic. Business activity ensemble includes all the elements you need to provide business activity monitoring (BAM) as part of an enterprise integration project. Business rules allow non-technical users to change the behavior of ensemble business processes at specific decision points. Business logic describes ensemble features that enable non-technical users to assess and direct the flow of application logic in order to meet business goals.

To realize e-business process flexibility, there are 18 characteristic relevant patterns for business processes changes in PASA [28, 29] (Figure 1). All these change patterns constitute solutions for realizing commonly occurring changes in PASA. We divide the change patterns into two major categories: adaptation patterns and patterns for changes in predefined regions. Thereby, adaptation patterns support structural process adaptations, whereas patterns for changes in predefined regions allow for built-in flexibility [28]. A set of adaptation patterns which can be used to accomplish changes at the process type or process instance level [28, 29]. Process instance level also means ad-hoc evolution; process type also means evolutionary evolution. For both process type and instance level changes, patterns speed up the change process by reducing the efforts for conducting a respective change through raising the level of abstraction. In case of ad-hoc changes not only cost of change can be reduced, but changes also become more applicable for end users as complexity of change is hidden from them. Patterns AP6 and AP7 mainly occurred in inter-organizational business processes. Patterns for Changes to Predefined Regions have identified four patterns. These four patterns differ regarding the parts that can remain unspecified results in a different degree of freedom during run-time. The relevant business processes change types with business processes changes pattern see Table 1.

Figure 1. Overview of identified change patterns [28]
Table 1. Relating business process change types with change patterns

<table>
<thead>
<tr>
<th>Change Types</th>
<th>Change Patterns</th>
<th>Related Change Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Activity Change</td>
<td>AP1,AP2,AP3,AP4,AP5,AP6,AP7,AP14</td>
<td>Adding / Deleting Fragments; Moving / Replacing Fragments; Adding / Removing Levels</td>
</tr>
<tr>
<td>Business Rules Change</td>
<td>AP8,AP10,AP13,PP1,PP2,PP4</td>
<td>Adapting Ctrl Dependencies; Change Transition Conditions; Patterns for Changes to Predefined Regions</td>
</tr>
<tr>
<td>Business Logic Change</td>
<td>AP9,AP11,AP12,PP3</td>
<td>Adapting Ctrl Dependencies; Patterns for Changes to Predefined Regions</td>
</tr>
</tbody>
</table>

2.2 Support Features Description

A pattern-based analysis combines self-adapting and self-evolution theory in PAIS that is the support for business process-aware feature, including the evolution of the static, dynamic evolution or dynamic change, dynamic refinement [25, 26, 34]. Relevant the evolution support features include process schema evolution, version control and instance migration, change correctness, change traceability, access control, change reuse, and concurrency control [28].

- Static evolution process is mainly based on demand and changes in the environment, the use of the evolution operator allow the process model to be modified, making the process model from a version upgrade to another version of the process. Such a schema evolution often necessitates the propagation of respective changes to ongoing process instances, it must stop all ongoing treatments, abort them and re-starting them.
- The dynamic process of evolution means that when the process model is changed, the corresponding changes in the dynamic process model then spreads to the running process on the process instance.
- The dynamic refinement mainly use Patterns for changes in predefined regions allow for dealing better with uncertainty by deferring decisions regarding the exact control-flow to run-time. Instead of requiring a process model to be fully specified prior to execution, parts of the model can remain unspecified. Dynamic refinement mainly occurred in inter-organizational business processes.

2.3 Process-Aware Software Architecture

![Diagram](image)

Figure 2. Internal organizational process and external organizational process [30]

Initially, process-aware software architecture was mainly oriented towards internal-organizational settings. Focus was on the use of process support technologies (e.g., workflow systems) to automate operational processes involving people and applications inside an organization (or even within an organizational unit). Over the last few years, there has been a push toward processes that cross-organizational barrier [30] (Figure2). Such cross-organizational processes can be one-to-one (i.e.,
bilateral relations), one-to-many (i.e., an organization interacting with several others), or many-to-many (i.e., a number of partners interacting with each other to achieve a common goal). 

3. CROSS-ORGANIZATIONAL DYNAMIC PROCESS-AWARE SOFTWARE ARCHITECTURE

The trend toward cross-organizational PAISs is marked by the emergence of B2B integration standards that define collections of common B2B integration processes (e.g., for procurement) or support the definition of such processes. It is also apparent in the emergence of notion Web Service composition, whereby applications are exported as services and composed by means of process models. A number of tools implementing these standards are now emerging, and established tools for cross-organizational application and process integration are being extended to support these standards. Therefore, CD-PASA is an extremely important form of process-aware software architecture.

3.1 The Important of Cross-Organizational Dynamic Process-Aware Software Architecture

A pattern-based analysis, through of CD-PASA realize to support managed e-business processes include process change in six common phases: market recognition, partner exploitation, technology matching, partnership formation, operation, and termination. CD-PASA reduced business processes uncertainty and processes variability and cost of change. As indicated in Figure 4, during a cross-organizational information system (IOS)-mediated collaboration episode, e-business enterprise may collaborate with one or more partners by configuring and reconfiguring IOS resources to collectively produce a product or deliver a service. Concurrently, e-business enterprise may engage in several collaboration episodes with the same or different partners. E-business enterprise may engage in joint marketing arrangements with its partners through shared repositories. It may allow its customers to place orders through the company’s intranet. It may also use...
groupware or an extranet for joint product design with its partners. As more and more IOS links are established, these links come to create the patterned networks of electronic partnerships, in which e-business enterprise and its IOS partners are embedded.

![An IOS Mediated Collaboration Episode](image)

**Figure 4. Cross-organizational dynamic process-aware software architecture realize to support managed e-business processes** [33]

Supporting the evolution of process model at run-time stage and propagating the changes of process model to the active process instances are fundamental requirements for flexible CD-PASA. Instance migration is regarded as the mainstream technique that deals with the dynamic evolution of process-aware software architecture. By using this technique, active process instances are migrated to the modified process model, so that they can benefit from the optimization process. CD-PASA appears in two forms: public and private. Public processes are those whose definitions are visible to parties outside the organization which implements the process. On the other hand, the definition of a private process is only visible to the organization that owns it. The rationale behind this distinction is twofold. On the one hand, organizations do not wish to expose the full details of their processes to other organizations. Instead, they would only expose the parts of the process that relevant for establishing a given collaboration. On the other hand, it allows for partners to be replaced. An organization A partnering with an organization B in the context of an cross-organizational process is able to substitute B for another organization C, so long as C provides a public process compatible with the requirement of A. However, simply looking at the supported patterns and counting their number is not sufficient to analyze how well a system can deal with process change. One characteristic of an information system is that different components depend on each other for data and processing. The pattern of these dependencies is the information systems architecture. CD-PASA needs some change support feature to support managed process.

### 4. MECHANISM ANALYSIS THE RELATIONSHIP BETWEEN CROSS-ORGANIZATIONAL DYNAMIC PROCESS-AWARE SOFTWARE ARCHITECTURE AND CHANGE PATTERNS OF BUSINESS PROCESSES

A pattern-based analysis combines self-adapting and self-evolution theory in CD-PASA shape prototype system. As indicated in Figure 5, CD-PASA components included the following aspects: self-adaptation [21, 22], such as process choreography [23], inter-organizational information sharing-and-hidden [24], separation of concerns [25, 26] and self-evolution [27]. This section is about the relationship between the collection of software components that comprises an organization’s information system and the organization’s ability to respond to changes in its business and technology environment. CD-PASA is a combination of components that function as
a complete whole. We divide the design of CD-PASA into components and architecture. The components are the parts of a CD-PASA that the software developers implement in computer code; architecture describes the overall form of the CD-PASA.

4.1 Change Support in Cross-Organizational Dynamic Process-Aware Software Architecture

The kernel of the process server is realized as a multi-layered architecture. As indicated in Figure 5, the Low-level Services Layer is a thin abstraction on SQL, enabling a DBMS independent implementation of persistency. The Basic Services layer is responsible for storing and locking different entities of the process management system (e.g., process templates and process instances). The Execution Layer encapsulates essential process support functions including process enactment and change management. The User Interaction layer provides different build-time (BT) and run-time (RT) tools to users, including a process editor and a process monitoring component. A process instance may either be controlled either by a single server or by multiple servers if that is favorable. Different clients can be connected with each server covering their work list programs, monitoring components, and modeling tools.

In CD-PASA all patterns are relationship with relevant change support features, for implementation of non-standard clients, CD-PASA offers a rich API. In basic components, the Execution Layer, processes client API calls (e.g., to start an activity or to perform a change). It uses execution manager and runtime environment component. Execution manager component coordinates the execution of process instances in an efficient and correct way. Runtime environment component provides the container for executing arbitrary applications. Each call is decomposed into a set of service requests from the Basic Service Layer components such as activity repository, process repository, process manager and data manager comprises services designed along the described features (e.g., for scheduling process activities, dynamically changing process instances, managing user work lists, or handling temporal constraints). ActivityRepository component manages the activity templates based on which processes can be composed and executed. ProcessRepository component manages process templates and their Meta data. While the above components manage buildtime data, the Process-Manager component provides exactly that information needed for process enactment. For each process instance the DataManager maintains all process (relevant) data created during process enactment. Each component of the Service Layer itself decomposes calls into basic operations (e.g., to read, to create, or to modify process objects) through log Manager, persistence (DBMS), configuration & registry framework and communication for the Low-level Services Layer. Configuration & Registry Framework component provides the basic infrastructure for configuring and managing the different system components of the architecture, and for enabling
inter-component communication. LogManager logs all system events occurring at build- and runtime.

In dynamic evolution or dynamic change components must be combinable in a flexible way to realize higher-level services like ad-hoc flexibility or process schema evolution. This includes build-time tools for modeling, verifying and testing processes, runtime components for monitoring and dynamically adapting process instances, and work list clients for accessing upcoming tasks. Many applications, however, require adapted user interfaces and functions to integrate process support features in the best possible way. The dynamic evolution components include process editor, org model editor, monitor, simulation/test, work list manager, progress coordinator, change operations, resource manager, and org mode manager. Org model manager and resource manager are to define potential actors for a particular activity; it can be associated with an actor assignment. Control center provides advanced buildtime and runtime components for user interactions. Worklist manager component manages work lists.

In dynamic refinement components include change operations kernel component comprises high-level change operations that can be applied for processes in different context (e.g., to add or delete activities), included correctness of change, traceability and analysis of changes, access control for changes, another schema evolution, version control and instance migration, instance-specific changes and change reuse, change concurrency control. Version control and instance migration is primarily relevant for changes at the type level, support for instance-specific changes and change reuse is particularly useful at the instance level.

In cross-organizational information sharing-and-hidden components include virtual process engine, base process definition tool, virtual process definition tool, base process engine, role designer. The role designer maps role definitions onto the organizational units, groups, and users. The role definitions contain job descriptions and obligations, and are consulted during base/virtual process design. A process modeler employs the role designer and the virtual/base process definition tool to specify workflow participants (internal workers and trading partners), base processes, process-views, and integrated processes. During run-time, the processes manager consults org model manager to obtain clients’ responsibilities, and then submits/receives messages to clients’ work list manager or progress coordinator. Execution manager supports publish/subscribe asynchronous communication mechanism in the process-view model. The client-side work list presents the activities that must be performed by the client. The progress coordinator displays abstracted progress information of the workflow in which the client participates (progress monitoring). Moreover, a client can alter the execution state of a public process view via the coordinator (progress control). In e-business cooperation, trading partners can monitor and control the progress of internal processes through client-side progress coordinators.

5. EVIDENCING THE EXISTENCE RELATIONSHIP BETWEEN CHANGE PATTERNS AND CROSS-ORGANIZATIONAL DYNAMIC PROCESS-AWARE SOFTWARE ARCHITECTURE IN HEALTHCARE INDUSTRY

Escalating costs pushed dramatic changes in the healthcare industry, with a move to warding managed care. Managed care seeks for integrate healthcare delivery processes and continuously improve them through feedback based on evaluation of care outcomes. The success of managed care depends critically on the collection, analysis and seamless exchange of information within and across organizational borders. This section examines the business processes unique to managed care, and identifies its architecture components. Healthcare is characterized by high flexibility demands. Additionally, both require tools that are easy to use since domain specialists have no IT knowledge. By supporting domain-specific views on processes (e.g., clinical pathways) and services, the CD-PASA enables end-users to actively shape the different phases of the process as well as the service life cycle. Another important aspect concerns about application integration. In both domains system integrators are confronted with heterogeneous, autonomous applications, to be integrated in a
process-oriented way.

CD-PASA enabled the service-oriented provision of advanced process support features and simplified the integration of existing application systems. Fig. 7 shows a mobile client needing patient assistance in a hospital being connected to process server. For representing clinical pathways the CD-PASA uses editable tree structures, which are automatically mapped to (block-structured) process models. In particular, adaptations within such tree structure are translated into corresponding changes of the underlying process model and can be automatically applied for the considered instance using ad-hoc changes. This empowered domain experts (e.g. nurses) to change processes at a high level of abstraction. Additionally, features like robust process execution and user assistance in connection with ad-hoc process instance changes where considered being extremely useful features. Thereby one or more low-level change operations can be combined into form higher-level change patterns in order to perform domain-specific operations, such as continuous quality improvement, service integration, Medicare contracts, shared incentive with doctors, system-wide clinical service, community and so on. Table 2 summarizes relevant business process change with architecture components. We divide the relationship into three levels: strong, weak and no relationship. We measure the cost of making an architectural component generate with process change pattern \([39]\): the lower the cost the greater weak the relationship. Cost data were collected through a survey administered to the IT Service Owners. In CD-PASA, Adding / Removing Levels are strong relationship with Cross-organizational information sharing-and-hidden relevant change support features, and other patterns are weak. Patterns for Changes to Predefined Regions are strong relationship with dynamic refinement relevant change support features, and other patterns are weak. Adding / Deleting Fragments, Moving / Replacing Fragments, Adapting Ctrl Dependencies and Change Transition Conditions are strong relationship with dynamic evolution relevant change support features, and other patterns are weak.

<table>
<thead>
<tr>
<th>E-BUSINESS PROCESS CHANGE PATTERN</th>
<th>SOFTWARE ARCHITECTURE COMPONENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Basic Information sharing-and-hidden</td>
</tr>
<tr>
<td>Adding / Deleting Fragments</td>
<td>■</td>
</tr>
<tr>
<td>Moving / Replacing Fragments</td>
<td>■</td>
</tr>
<tr>
<td>Adding / Removing Levels</td>
<td>■</td>
</tr>
<tr>
<td>Adapting Ctrl Dependencies</td>
<td>■</td>
</tr>
<tr>
<td>Change Transition Conditions</td>
<td>■</td>
</tr>
<tr>
<td>Patterns for Changes to Predefined Regions</td>
<td>■</td>
</tr>
</tbody>
</table>

6. CONCLUSIONS

In this paper, we describe the emergence of different process support paradigms and the lack of methods for comparing existing change approaches have made it difficult for process-aware software architecture (PASA)
engineers to choose the adequate technology. This makes it difficult for PASA engineers to assess the maturity and the change capabilities of those technologies, often resulting in wrong decisions and expensive investments. Therefore, we need to provide a detailed mechanism analysis of the relationship between PASA and change of business processes.

Respective pattern-based analysis (adaptation patterns and patterns for changes in predefined regions) combines self-adapting theory (process orchestration and process choreography, separation of concerns, cross-organizational information sharing-and-hidden), self-evolution theory (static evolution, dynamic evolution or dynamic change, dynamic refinement) and technologies (e.g., workflow management systems, case handling tools) enable the definition, execution, and monitoring of the operational processes of e-business enterprise. In connection with Web service technology, in addition, the benefits of business process management from within a single e-business can be transferred to cross-organizational business processes as well.

A pattern-based analysis, divided from the organization-wide, e-business process-aware can be described as support of Internal Process and External Process. A pattern-based analysis combines self-adapting and self-evolution theory in PAIS that is the support for business process-aware feature, including the evolution of the static, dynamic evolution or dynamic change, dynamic refinement. We have proposed 18 change patterns and change support features which – in combination – allow for assessing PASA change frameworks. We put forwards four kinds of PASA, such as Internal-Organizational Static Process-Aware Software Architecture, Internal-Organizational Dynamic Process-Aware Software Architecture, Cross-Organizational Static Process-Aware Software Architecture, Cross-Organizational Dynamic Process-Aware Software Architecture. Different models require different features to support. Because CD-PASA is extremely important in e-commerce which reduced business processes uncertainty and processes variability, based on these change patterns and features, we provide a detailed mechanism analysis and case study evaluation in the healthcare industry of the relationship between CD-PASA and change patterns of business processes. A pattern-based analysis combines self-adapting and self-evolution theory in CD-PASA shape prototype system. CD-PASA includes dynamic evolution or dynamic change, dynamic refinement. We give the structural framework of each part by describing CD-PASA, meanwhile introduce self-adaptive implementation under the mode of component and explain the representation of component and self-adaptive evolutionary processes.

In the healthcare industry case study, we divide the relationship into three levels: strong, weak and no relationship. We measure the cost of making an architectural component generate with process change pattern: the lower the cost the greater weak the relationship. Cost data were collected through a survey administered to the IT Service Owners. We conclusion in CD-PASA, Adding / Removing Levels are strong relationship with Cross-organizational information sharing-and-hidden relevant change support features, and other patterns are weak. Patterns for Changes to Predefined Regions are strong relationship with dynamic refinement relevant change support features, and other patterns are weak. Adding / Deleting Fragments, Moving / Replacing Fragments, Adapting Ctrl Dependencies and Change Transition Conditions are strong relationship with dynamic evolution relevant change support features, and other patterns are weak, and all patterns are strong relationship with basic components. It consequently will support PASA engineers in selecting the right technology for realizing flexible PASA. In addition, this work can be used as a reference for implementing more flexible PASA.

PASA research has great significant impact on practice of relationships from operation mode over business processes to deployment. Change support features must be considered to make change patterns useful in practice. Our futures works include valuate additional academic and commercial systems with Internal-Organizational Static Process-Aware Software Architecture, Internal-Organizational Dynamic Process-Aware Software Architecture and Cross-Organizational Static Process-Aware Software Architecture.
REFERENCES