



Title	From process logic to business logic - A cognitive approach to business process management
Author(s)	Wang, M; Wang, H
Citation	Information And Management, 2006, v. 43 n. 2, p. 179-193
Issued Date	2006
URL	http://hdl.handle.net/10722/135593
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Wang, M., & Wang, H. (2006). From process logic to business logic: a cognitive approach to business process management. *Information & Management*, 43(2), 179-193.

From Process Logic to Business Logic — A Cognitive Approach to Business Process Management

Minhong Wang^{*} and Huaiqing Wang

Department of Information Systems, City University of Hong Kong

Received 20 December 2003; received in revised form 5 April 2005; accepted 18 July 2005

Abstract

The unpredictability of business activities means that business process management should provide a way to adapt to change. The traditional workflow approach, based on predefined process logic, offers little support for today's complex and dynamic business environment. Therefore, a cognitive approach is proposed to help manage complex business activities, based on continuous awareness of situations and real-time decisions on activities. In this approach, the business environment is seen as capturing events that occurred and the state of tasks and resources; business logic involving process routing, operational constraints, exception handling, and business strategy is used to determine which actions are appropriate for the current situation. By extending process management from process logic to business logic, the methodology offers flexibility, agility, and adaptability in complex business process management.

^{*} Corresponding author. Tel.: +852-2788-9962; fax: +852-2788-8535.
Email address: maggie_mh_wang@yahoo.com (M. Wang)

Keywords: Cognitive approach; Business process management; Workflow; Business logic; Process logic

1. Introduction

In recent years businesses have been facing the challenges of rapidly changing environments which rapidly change from centralized and closed to distributed and open with the business processes displaying more complexity because of interactions between their internal components and interactions of the processes with the environment. As a result, organizations are paying more attention to supporting business process management able to adapt to the new complex environments. Management of the business process is commonly considered to be the development of business applications that directly follow the execution logic of the underlying business process. Traditional approaches use workflow technology to model and manage the process, based on a predefined logical procedure of activities from a centralized perspective; i.e., a complete list of all the activities and all the paths are provided, the criterion for following a particular path is specified, and the ordering constraints on the actions are given. Such rigid and exact specification works well for simple and stable business processes; however, it is inadequate for complex and dynamic business activities due to its lack of flexibility and adaptability [2, 9, 12].

Realizing the need to provide sufficient flexibility and adaptability in business processes, many researchers are investigating adaptive workflow techniques, knowledge-based approaches, etc. [10, 19]. Most of these techniques emphasize the improvement of conventional workflow models by explicitly representing alternative paths through the process or by redesigning systems to deal with anticipated exceptions. They offer limited

flexibility and may exact a high cost in terms of business redesign or reconstruction. Another approach involves using rules in workflow models to enforce additional operational control of processes (e.g., the ECA approach). This technique uses predefined workflow schemas as well as rules in a static bind to build so-called rule/knowledge-based workflow systems. Rules make data repositories react to certain events and trigger special actions. By employing rules, the traditional workflow approach may support runtime processes when unusual events occur. For example in supply chain management, orders for components or raw materials should be placed as soon as the inventory level drops below the base-stock level. This activity can be scheduled by adding a rule to the workflow model. However, if we need a more sophisticated control mechanism for a complex situation that may have more uncertainty and unexpected combinations or interactions, this approach is not effective. When dealing with inventory management, in addition to the status of current inventory and actual orders, other issues such as order requests from prospective customers, supply capability of existing suppliers, bids from new suppliers, and changes from actual orders, should also be considered. If a large order arrives while the inventory is low, the promised supply is delayed, or a bid is received from a prospective supplier, all of the relevant information must be digested and sound and a fast decision must be made about the order processing with the customer, the bidding policy with the new supplier, and the coordination with existing suppliers. Moreover, all these activities may interact in the whole process without a routine procedure. Therefore, it is difficult to model such a complex process using traditional workflow even with a rule support addition.

We believe that a novel conceptualization of process management is needed to overcome the limitation of conventional approaches. As addressed by Simon in [23], the challenge of a changing and complex environment requires management of an organism using adaptive mechanisms to respond to the demands. This kind of mechanism is used to handle problems that are unstructured to the extent that there is an absence of routine procedures for dealing with them. In non-procedure paradigms we do not depend on systems giving exact details on how to accomplish a task, but let the system determine how to solve a problem [5]. How do the systems solve the problems? To answer this question, cognitive science has provided a conceptual framework about human thinking and decision-making [22]. Problem solving is an interaction between the behaving organism and the environment under the guidance of the control system [20]. During interaction, information is input, represented in memory as declarative knowledge, and then used in problem solving following algorithmic or heuristic steps.

In a business process context, an exact execution order of activities may be impractical, while the interaction or relationship between the environment and activities is more reliable in determining how to orchestrate tasks; i.e., the question of which task to execute and when to execute it is dependent on the current environment and underlying business logic rather than a static process schema. From this point of view, the rule/knowledge-based approach improves upon the traditional workflow approach by supplying real-time reaction to certain events that occur. Our method tries to go further by incorporating events into a context that watches over the environment and exercises timely and decision-based control of the execution of business processes that are unstructured because there is no routine procedure for dealing with them.

A cognitive approach is used to manage complex business activities based on situational awareness and real-time decisions. In a cognitive system, the relationship of the system to its environment is of vital importance [6], while, in a business process context, information about the process environment (such as events and the state of activities and resources) is collected as perception-based knowledge; business logic concerning process routing, operational constraints, exception handling, and business strategies is used to make decisions about tasks based on situation awareness. By extending process management from process logic to business logic, the approach offers more flexibility, agility, and adaptability in complex process management.

The traditional workflow approach focuses on specifying the execution order of activities, while the cognitive approach declares knowledge or rules about how to manage activities based on the current state of the environment. In today's business climate, business activities can be delegated to a number of autonomous problem-solving agents. They may be machines or software applications. The right-hand side of Figure 1 presents the mechanism of one agent, which is equipped with business rules to manage its own process by interacting with the environment, i.e., perceiving and reacting to the system, communicating with other agents, etc. In addition, the cognitive approach supports proactive actions on the basis of prediction of future state of the environment.

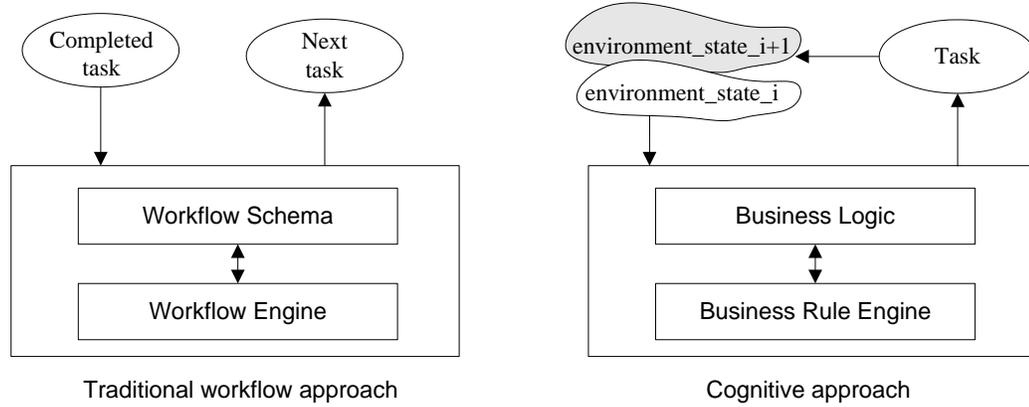


Figure 1. Traditional workflow *versus* a cognitive approach

2. Literature review

From the management point of view, business process management can be seen as a collection of methodologies, techniques, and tools supporting the analysis and improvement of business processes [17]. Organizational leaders today recognize process management as an essential element in organizational performance [14]. From a technology viewpoint, business process management supports business processes using methods, techniques, and software to design, enact, control, and analyze operational processes [28]. Workflow technology usually refers to the development of a Workflow Management System (WFMS), an information system that defines, manages, and executes “workflows” through the use of software whose order of execution is driven by a computer representation of the workflow logic (see <http://www.wfmc.org>). The WFMS has become a fundamental building block in the organization and has advanced considerably in recent years [1]. One of its fundamental assumptions is that workflow schemas (i.e., control flows) are predefined. This is a significant weakness.

In [11], several technical papers on adaptive workflow systems were collected. Among them, a number of techniques supported exception handling in workflow systems allowing dynamic adaptation of to the changing environment. They emphasized the improvement of conventional workflow models to deal with dynamic workflow adaptation. While they provided mechanisms for the seamless integration of exception handling into workflow descriptions, these approaches lacked the practical aspects that become important in workflow systems, such as the participation of autonomous, heterogeneous legacy systems, and the increased interactions among business activities.

An event-based model for workflows was employed in some systems [3]. For example, the ECA (Event-Condition-Action) rules were used to support exceptions and asynchronous behaviour during workflow execution by enhancing database technology and extending transaction management [7]. ECA rules were advocated by database practitioners to make data repositories react to internal or external events and trigger a chain of activities. The ECA approach uses predefined workflow schema as well as ECA rules; these are used as a supplement to the workflow schema for reaction to certain events, and thus they support adaptation to environments by responding to certain expected changes using events and triggers. However, if we need a more flexible control mechanism for complex or unstructured process environments, characterized by increased rate of expected or unexpected changes and increased volume of interactions with an absence of routine procedures for dealing with them, a novel approach to process management is required.

The application of agents for managing business processes has been studied in a flurry of research [4, 21, 27, 30]. The term “agent” denotes an encapsulated software-

based computer system that has autonomy, social ability, reactivity, and pro-activity. A multi-agent system consists of a group of agents, interacting with one another to achieve their goals [16, 29]. As defined in work on distributed artificial intelligence (DAI), a multi-agent system is a loosely coupled network of problem-solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of each entity [24]. By modularizing a complex problem in terms of multiple autonomous components that can act and interact in flexible ways, agent technology is well suited for complex, dynamic, and distributed software systems [31]. An agent-based workflow system usually consists of multiple agents, each responsible for specific work items. The management of the whole process is either performed by one routing agent through its mediating with other agents or the whole process is broken into pieces and embedded into each agent.

3. Cognitive approach to complex business process management

3.1. Background theory

According to Hammer and Champy, “A business process is a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer” [8]. Rather than viewing a business process as an assembly of interchangeable components, Melão and Pidd recognized the complex, dynamic and interactive features of business processes. Changes in their environments result in dynamic and uncertain business processes. While business environments are rapidly changing from centralized-and-closed to distributed-and-open, mainly due to the proliferation of World Wide Web,

the dynamic and interactive feature of business processes is increasingly seen as the complexity of this domain.

Complexity is created and driven by the perception of increased rate of changes and increased volume of interactions. A complex problem is usually semi-structured or unstructured -- there is an absence of routine procedures for dealing with it. Cognitive research has worked on a conceptual framework to deal with questions of ever-increasing complexity, problem solving, thinking, and language. Meanwhile, computer scientists studying cognition concern themselves with problem-solving skills and developing new domains of cognitive research.

3.2. Mechanism

Based on the theories about cognition and complexity, we can design a cognitive and adaptive mechanism that manages processes by responding flexibly to the demands of the environment. In the business environment, process management resembles real-time, dynamic decision-making tasks (RTDDM), defined as “a decision task that requires a series of interdependent decisions in a continuously changing environment” [13, 26]. It has four key characteristics: (1) the tasks require a series of decisions; (2) the decisions are interdependent; (3) the environment changes autonomously and as a result of decisions; and (4) decisions are made in real time. The researchers studying real-time decision tasks developed a concept called “situational awareness,” which is the ability to know what is going on in a real-time dynamic environment. There are three levels of situational awareness. The first is the ability to perceive the status, attributes, and dynamics of relevant elements in the environment. The second refers to an individual’s

selection of an action on the basis of the current situation. The third involves the capacity to select an action based on the predictions of the future state of the system. In our approach, the first is to perceive and understand the workflow environment through message communication and monitoring. The second and third represent the reasoning of reactive and proactive tasks based on the environment.

The cognitive approach here is characterized by an ability to perceive the business process environment and make real-time decisions about tasks based on underlying business logic. It orchestrates business processes dynamically at runtime and continues the evaluation of the process throughout execution, during which time business changes occur and business rules are dynamically bound to process orchestration (see Figure 2). Furthermore, the cognitive approach not only provides real-time reaction to the process environment but also supports proactive activities, which refer to the exhibition of goal-oriented behaviours taking an initiative.

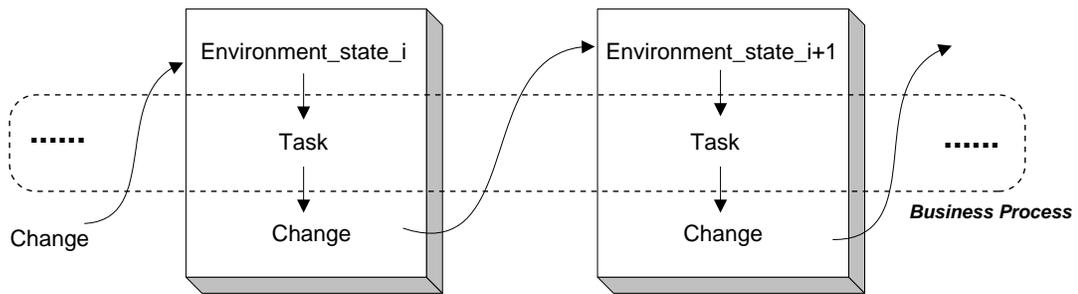


Figure 2. Business process in the cognitive approach

Compared with ECA rules used for reaction to certain events, the business logic in the cognitive approach works as knowledge for a process decision in the current process environment. It attempts to incorporate events into a context that observes the whole process environment and exercises timely and decision-based control of the

execution of unstructured processes. The major difference between the reactive mechanism in the ECA approach and the cognitive mechanism in our approach is shown in Table 1. Besides the difference at the conceptual level, the two [approaches differ](#) in their constructs, notations, ease of use, and other practical aspects. The ECA approach involves the context of active databases. It lends itself to a workflow implementation in a database environment, which supports events and triggers. While the ECA approach is a blend of predefined workflow schemas and ECA rules, the cognitive approach has no specific procedures. It deals with the way that the system organizes incoming information objects, maintains representative objects in memory, and accesses them to orchestrate and perform various tasks. The model consists of input-processing-output, storage or memory, and operators that represent rules or strategies for process management and problem solving.

Table 1. The reactive mechanism in the ECA approach versus the cognitive mechanism in the proposed approach

function Reactive-BPM (percept) return action	function Cognitive-BPM (percept) return action
static: rules,	static: environment,
.....	knowledge-base,
event <= Interpret-input (percept)
rule<= Rule-Match (event, rules)	environment <= Update-World-Model (environment, percept)
action<= Rule-Action[rule]	state<= Update-Mental-State (environment, state)
return action	action<= Decision-Making (state, knowledge-base)
	environment <= Update-World-model (environment, action)
	return action

3.3. Knowledge model

Given the problem of complexity, one of the societal goals should be to enable people to convert information into knowledge through understanding. Furthermore, the knowledge level is useful for designing systems whose internal workings are yet to be determined. The special feature of the knowledge level is that it can be given before anything about the internal workings of the system is determined. In a cognitive system, information input and represented in memory is termed declarative knowledge (knowledge of facts). However, procedural [knowledge](#), drives cognitive processes, such as problem solving and decision-making. For process management, the knowledge is concerned with perception of the environment, and business logic for process management.

- **Environment state**

The environment state is composed of a set of information representing the current status of the process environment. It is mainly used for the tracking of events that have occurred in the environment, or the state of tasks or resources. In most situations, such information comes from input from external systems, intervention from users, output from completed tasks, and status of key tasks and resources. These data are collected and transformed into information or facts used for the reasoning of tasks.

- Event

The information about events that have occurred usually refers to the messages that can be captured by the system and then interpreted into event facts for reasoning. For

example, the event of receiving an order request No.15421 from a customer can be interpreted as:

(event (e-type order_request) (ord-no 15421))

- State of resources and tasks

The state of resources and tasks can be continuously detected by the system, and represented as facts for reasoning. The state of an order can be presented as a fact, from which we know the validation and confirmation status of order No.15412.

(order (ord-no 15412) (validation validated) (confirmation NULL))

The state of a task is described as the fact that the validation activity on order No.15420 has started and is not yet finished:

(task (t-type validate_order) (ord-no 15420) (start-time 1.009598657E9) (finish-time NULL))

- **Business logic**

In general, business logic is a set of formal or informal statements about how business is done. It can be represented in the form of business rules, each of which represents a small unit of knowledge of business management. Business rules could be controlling workflow, reducing business risk, making efficient use of resources, or improving customer service [18]. In our approach, rules of process management cover process routing, operational constraint, exception handling, and business strategy. Such rules form the knowledge base of the cognitive approach to process management.

- Process routing

Rules on process routing are used for scheduling tasks or activities. If an order request is received from a customer, it is scheduled for validation. This can be specified by a rule which activates the task of order validation based on the order

request. The information about this order is generated as well for the further tracking of its status.

```
(defrule order-request
(event (e-type order_request) (ord-no ?no))
=> (assert (task (t-type validate_order) (ord-no ?no) (start-time (time))))
(assert (order (ord-no ?no) (validation NULL) (confirmation NULL))))
```

Four basic types of process routing are generalized; these include sequential routing, conditional routing, parallel routing, and iterative routing. Each type is specified using AND/OR operators and split/join blocks, based on which process routing is mapped into the business rules.

- Operational constraint

Rules about operational constraints may be used to exercise control of tasks, and prohibit unauthorized operations. They can be specified on the basis of business requirements. The rule below rejects the cancellation of an order that has been confirmed.

```
(defrule reject-confirmed-order
(event (e-type order_cancel_request) (ord-no ?no))
(order (ord-no ?no) (confirmation ?cfm))
=> (if (= ?cfm confirmed) then
(assert (task (t-type reject_cancel_order) (ord-no ?no)))
else
(if(= ?cfm NULL) then
(assert (task (t-type cancel_order) (ord-no ?no))))))
```

- Monitoring rule

In order to keep track of changes and reduce business risks, monitoring rules can be applied to capture the real-time status of tasks or resources and take proactive actions when necessary. For example, if an order request has not been

validated or rejected in 20 minutes (1200 seconds), the system will remind the user of the validation task unless this reminder has already been delivered.

```
(defrule remind-validation
(task (t-type validate-order) (ord-no ?no) (start-time ?t1) (finish-time NULL))
(test (> (- (time) ?t1) 1200))
(not (remind-history (r-type remind_validate_order) (ord-no ?no)))
=> (assert (task (t-type remind_validate_order) (ord-no ?no)))
(assert (remind-history (r-type remind_validate_order) (ord-no ?no))))
```

- Business strategy

Business logic could be used both for controlling workflow and for reducing business risk, making efficient use of resources and improving customer service. They can be seen as one type of business logic, though most are implicit and difficult to translate into business rules. The following is a promotion strategy, in which a gift will be sent to a customer who has made a large order.

```
(defrule send-gift
(event (e-type order_confirmed) (ord-no ?no))
(large-order (ord-no ?no))
(not (gift-sent (ord-no ?no)))
=> (assert (task (type send_gift) (ord-no ?no)))
(assert (gift-sent (ord-no ?no) (send-time (time)))))
```

3.4. Framework

Based on this discussion of the mechanism and knowledge model of the cognitive approach, we designed the framework shown in Figure 3. Considering the large-scale and distributed settings in today's business environment, business activities can be delegated to a number of autonomous problem-solving agents, each of which plays a specific role. For example, in an order management system, several kinds, such as a customer agent, validation agent, payment agent, etc., can be specified. Each agent can possess certain

resources and is equipped with specific business logic. The agents perceive changes in the internal and external environment and make decisions on tasks based on their business logic. From the perspective of the whole environment, each agent works both autonomously and collaboratively to achieve business goals.

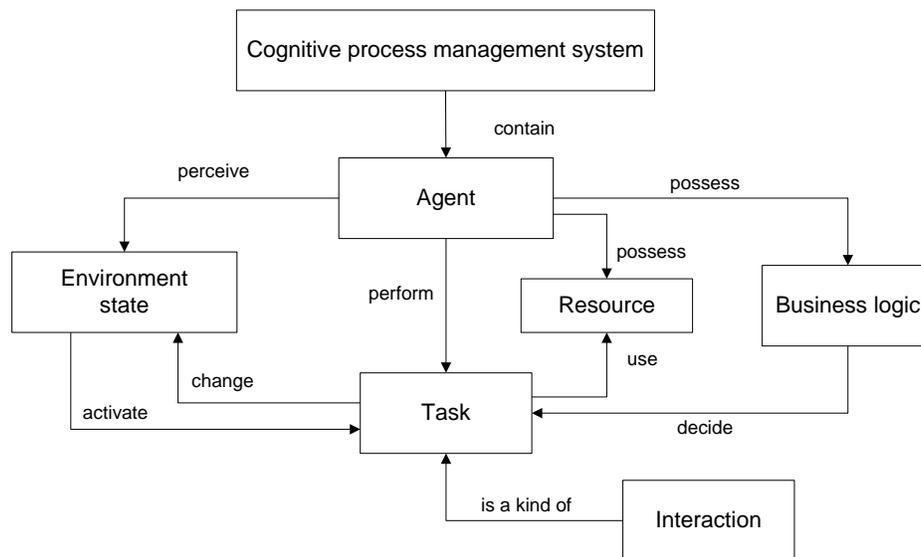


Figure 3. The cognitive process management framework

4. A Case Study

To illustrate the proposed cognitive approach, a case involving exception management in securities trading is presented. For this case study, a prototype system was developed to automate the identification and resolution of exceptions so as to assist securities industry gain quicker results and thus a competitive advantage. This method provides more intelligence, flexibility, and collaboration in business exception management. Considering the complexity and interactive processes in exception management, the approach involved delegating tasks to a number of cognitive agents. The system was built using the Java Web Services Development Package (JWSDP) and

Java Expert System Shell (JESS). The effectiveness of this prototype was evaluated through a use case and demonstration feedback [25]. Although the case study does not precisely reflect real-world situations, the results reveal the success in using a cognitive-agent-based DSS for exception management.

4.1. Case description

With rising trading volumes and increasing risks in securities transactions, the securities industry is shortening the trade lifecycle and minimizing transaction risks. While attempting to achieve this, exception management is critical to pass trade information within the trade lifecycle in a timely and accurate fashion.

- **Trade Detail Monitoring** captures unusual components in trade details, such as a trade dealt at a price different from the market price, incorrect calculation of trade cash value, missing components, etc.
- **Trade Status Monitoring** watches the agreement status of each trade and detects those trades that have not been agreed upon by the trading parties in a timely manner (or trades in which the two parties disagreed).
- **Diagnosis** investigates the nature of exceptions and offers resolution advice to settle problems.
- **Resolution** carries out actions on exceptions based on the output of the diagnosis.

Generally speaking, the process starts with the monitoring of trade details and trade agreement status. Any exception will result in the diagnosing activity, and, subsequently, a diagnostic report with resolution advice. Once the advice is validated by the manager, resolution action will remove the exception. However, the process in real-world

situations is more complex. The following are some possible situations that are not included here:

- 1) Information contained in an error report may not be enough to make a diagnosis, and additional data may be required.
- 2) The resolution advice on an error is not produced within a normal time frame due to lack of information, etc.
- 3) In order to check exception resolutions to large trades, the advised resolutions have to be confirmed by the diagnostic expert.
- 4) The diagnostic expert may perform an adjustment on resolution advice.
- 5) The chief manager needs to query related information before validating resolutions to exceptions.
- 6) The chief manager may be unavailable to validate the resolution advice.
- 7) An emergent error should be reported to the chief manager for instant action.

Besides being complex, exception management is also a kind of collaborative process, in which multiple organizations and a mixture of human activities and automated tasks may be involved. The cognitive approach is applied by delegating complex exception management tasks to a collection of agents. Each agent plays a role in exception management and is in charge of specific tasks. The framework of the system is portrayed in Figure 4, which describes the internal interactions among agents and the external relationship between the exception management system and legacy trading and settlement systems.

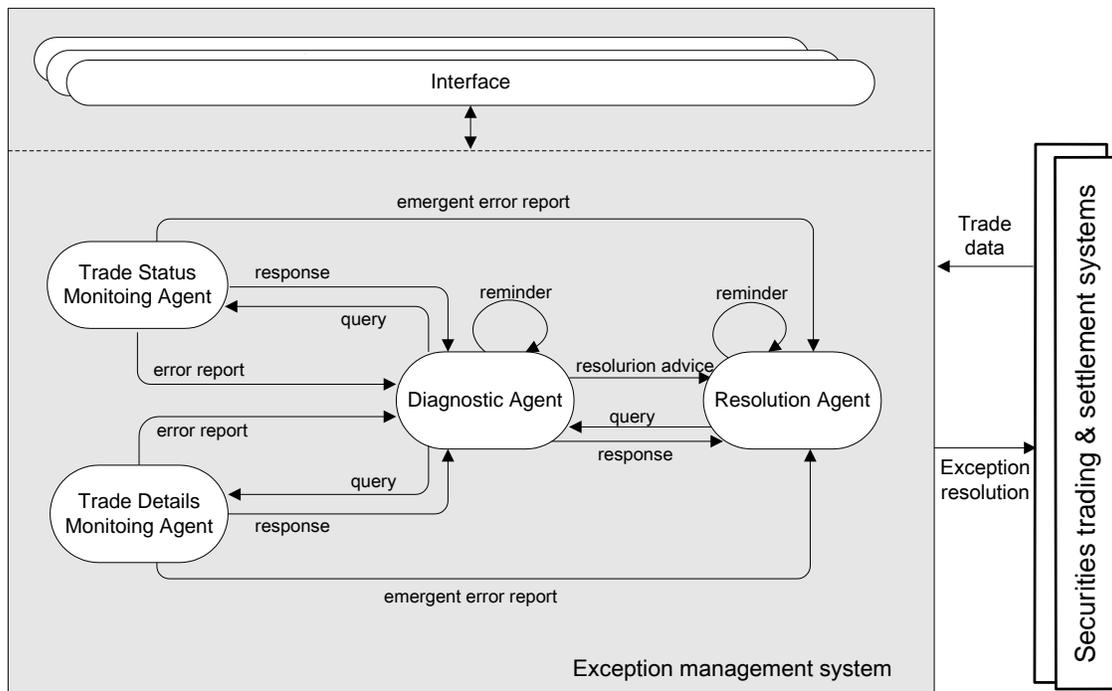


Figure 4. The exception management system for securities trading

The agents are distributed in organizations or departments involved in securities transactions; they communicate with each other through the Internet. The deployment of the agents may differ depending on the layout of the STOs (Securities Trading Organizations). Commonly, the Trade Detail Monitoring Agent (TDMA) is placed with the trading system, while the Trade Status Monitoring Agent (TSMA) is placed with the settlement system. They monitor exceptions. The Diagnostic Agent (DA) can be set up in the risk management department to examine the nature of the exceptions and provide resolution advice. Based on the report of the DA, the Resolution Agent (RA) placed in the chief management department can carry out actions to resolve exceptions. Besides agents, interface facilities are employed in relevant locations to support the interaction between the system and users.

The exception management prototype was based on the approach already discussed, where each agent was built as a cognitive entity equipped with specific business logic, from which it controls its process and interacts with others. Only the Diagnostic Agent is illustrated here as an example of a cognitive agent in the system. The details of this agent are:

- Environment state
 - Event

The major events that occur to the Diagnostic Agent are listed in Table 2.

Table 2. Events that occur to the Diagnostic Agent

Event	Meaning
(event (e-type error_report) (err-no 102) (trd-no 12362))	An error report No.102 is received from a monitoring agent.
(event (e-type trade_detail_request) (err-no 102) (trd-no 12362))	A data request for trade details is reported; this request is related to error No.102 in trade No.12362.
(event (e-type trade_status_request) (err-no 105) (trd-no 12569))	A data request for trade status is reported; this request is related to error No.105 in trade No.12569.
(event (e-type data_receive) (err-no 102) (trd-no 12362))	The requested data of trade No.12362 are received for diagnosing error No.102.
(event (e-type resolution_advice) (err-no 102) (trd-no 12362))	The resolution advice for error No.102 is generated to resolve the error in trade No.12362.
(event (e-type modify_resolution) (err-no 102))	The request to modify the resolution to error No.102 is received.

- State of task and resource

As shown in Table 3, the information regarding tasks and resources of the Diagnostic Agent are used in its real-time decision regarding process control.

Table 3. State of tasks or resources of the Diagnostic Agent

Task or resource	Meaning
(task-ongoing (t-type diagnose) (err-no 102) (start-time 1.009598657E9) (finish-time NULL))	The diagnostic task on error No.102 has started and is not yet finished.
(held-error-report (err-no 102))	The diagnosis on error No.102 is suspended for lack of data.
(remind-history (r-type remind_diagnose) (err-no 101))	The reminder message has been sent out for the delayed diagnosis on error No.101.
(large-trade (trd-no 13258))	Trade No.13258 is of large value.
(confirmed-resolution (err-no 109))	The resolution advice on error No.109 has been confirmed.

- Business rules

Rule-1 starts a diagnostic task when an error report is received.

```
(defrule rule-1 "start diagnosis"
(event (e-type error_report) (err-no ?e_no))
=> (assert (task (t-type diagnose) (err-no ?e_no) (start-time (time))))))
```

Once a data request for trade detail or trade agreement status has resulted from a diagnostic task, rule-2 or rule-3 is fired to activate the request task.

```
(defrule rule-2 "request trade detail"
(event (e-type trade_detail_request) (err-no ?e_no) (trd-no ?t_no))
=> (assert (task (t-type request_trade_detail) (err-no ?e_no) (trd-no ?t_no))))
```

```
(defrule rule-3 "request trade status"
(event (e-type trade_status_request) (err-no ?e_no) (trd-no ?t_no))
=> (assert (task (t-type request_trade_status) (err-no ?e_no) (trd-no ?t_no))))
```

When requested data are received, rule-4 is fired to start the corresponding diagnostic task that was held for lack of data.

```
(defrule rule-4 "continue diagnosis"
(event (e-type data_receive) (err-no ?r_no))
?held-error <= (held-error-report (err-no ?e_no))
=> (assert (task (t-type diagnose) (err-no ?e_no) (start-time (time))))
(retract ?held-error))
```

In order to check exception resolutions to those trades of large value, the advised resolutions to such trades are required to be confirmed by the diagnostic expert.

```
(defrule rule-5 "confirm resolution advice"  
(event (e-type resolution-advice) (trd-no ?t_no))  
(large-trade (trd-no ?t_no))  
=> (assert (task (t-type confirm_resolution) (err-no ?e_no) (start-time (time))))))
```

The diagnostic expert can adjust the resolution only before it is confirmed. This policy is specified in rule-6.

```
(defrule rule-6 "reject modification of resolution"  
(event (e-type modify_resolution) (err-no ?e_no))  
(confirmed-resolution (err-no ?e_no))  
=> (assert (task (t-type reject_modify_resolution) (err-no ?e_no))))
```

Rule-7 keeps watch on diagnostic tasks, and send out a reminder message if they have not been completed within a specified time frame.

```
(defrule rule-7 "remind diagnosis"  
(task-ongoing (t-type diagnose) (err-no ?e_no) (start-time ?t1) (finish-time NULL))  
(test (> (- (time) ?t1) 1200))  
(not (remind-history (r-type remind_diagnose) (err-no ?e_no)))  
=> (assert (task (t-type remind_diagnose) (err-no ?e_no)))  
    (assert (remind-history (r-type remind_diagnose) (err-no ?e_no))))
```

The operation of the Diagnostic Agent is (see Figure 5):

- When an error is reported by the Monitoring Agent (1), this event is transformed (2) as the fact: $(event (e-type error_report) (err-no 102))$.
- This fact fires rule-1 (3). As a result, a diagnostic task is set in motion (4), and then activated (5).
- However the expected resolution advice is not created for lack of data.
- Subsequently, the data request is sent out (6) and this error is recorded in a held-error-report waiting for diagnosis until the data are received (7). The data

request is interpreted as a corresponding fact (8), which fires rule-2 or rule-3 (9).

- The data request task is scheduled (10), activated (11), and the request is sent out to the Monitoring Agent (12).
- When the requested data arrives (13), this event is translated into a fact (14) and this fires rule-4 (15). Thus, the diagnostic task on the suspended error in the held-error report is recalled (16) and activated (17), and resolution advice is sent to the Resolution Agent (18). If the diagnostic task is not completed within a specified time frame (19), this fact may fire rule-7 (20), schedule, and activate the task (21) of sending a reminder message to the relevant manager (22).
- This reminder is recorded in the history, and will not be sent again.

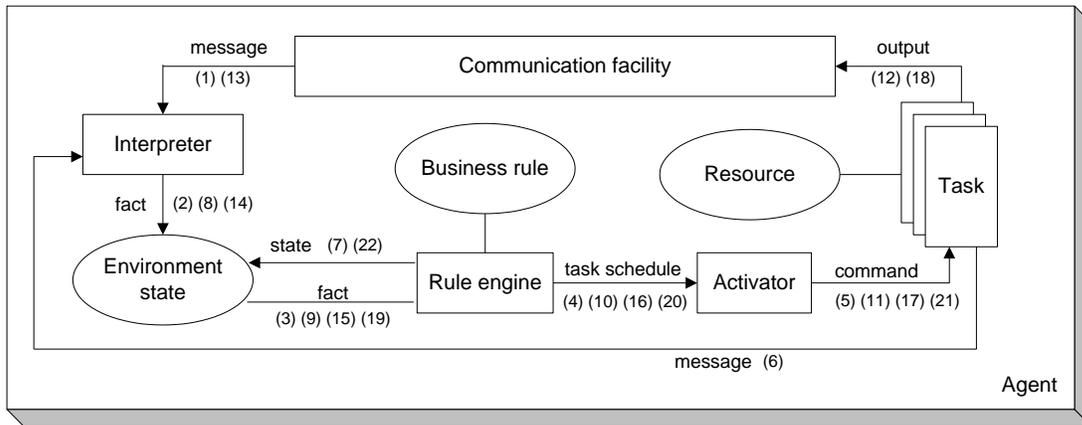


Figure 5. Operation of the cognitive agent

4.2. Case discussion

This case illustrates how the cognitive approach can be applied to complex process management with the purpose of increasing the ability of managers to organize,

monitor, and control dynamic business activities. In order to assess our approach, a comparison of this approach with others is discussed here.

Table 4. Comparison of cognitive approach with workflow approach

Features	Workflow approach based on predefined process schema	Workflow approach based on predefined process schema with ECA rules	Cognitive approach
Task routing	++	++	+
Operational constraints	+	++	++
Reaction to certain events	-	+	++
Continuous perception of environment	-	-	+
Decision-based control of process	-	-	+
Manipulation of business strategies	-	+	++
Support of interactive tasks	-	-	+

As shown in Table 4, traditional workflow technology focusing on predefined process logic is good at task routing, with the limited function of operational constraints, by using conditional branches in workflow models. However, it falls short of having the features of flexible workflow management. By supplementation with ECA rules, workflow technology improves its function in its reaction to events and constraints on operations, however there is no way to adapt to more dynamic processes of high rates of change (expected and unexpected) and increased volume of interactions as there is no routine procedure in place. The cognitive approach was designed to overcome the limitations of the workflow approach by supporting the capability of continuous perception of the process environment and real-time and decision-based control of the process. Moreover, business strategies can be employed to process management by mapping them into rules. By using the cognitive approach, a complex process may be modeled and controlled through the rules for task routing. Furthermore, constraints on

operations, monitoring of task progress, and manipulation of business strategies are easy to perform by including them in business logic for process management.

The cognitive approach is applicable to a number of complex—dynamic, open, and interactive—domains like electronic commerce (EC), manufacturing resource planning (ERP), supply chain management (SCM), project management, etc. This approach is particularly well suited for situations that are not all known a priori, cannot all be assumed to be fully controllable in their behaviors, and must interact on a sophisticated level of communication and coordination. For example, a typical supply chain faces uncertainty in terms of supply, demand, and process, etc. Moreover, in a supply chain, business entities are highly interdependent in order to achieve coherence among them [15]. A large number of interacting decisions may take place between different entities, most of which are impossible to foresee at design time. Furthermore, with mixed and often conflicting objectives, processes are sophisticated and difficult to use closed-form analytical solutions. In this kind of environment, business managers have no way to deal with the situation, but fall back on whatever general capacity they have for intelligent, adaptive, problem-oriented action. The cognitive approach is directly applicable to this type of application domain.

5. Conclusion

There has been considerable interest in the development of process management systems that offer a flexible and dynamic mechanism for routing and operational controls in workflow management. Our paper has developed a cognitive approach for implementing real-time routing and strategic control of dynamic and complex process

management. In this, the business process is not specified in a routine procedure, but is managed through business logic represented by a set of business rules that control tasks. Considering the complex process in a dynamic environment with unstructured activities and volumes of interactions, this approach is designed to achieve the flexibility and adaptability in process evolution by continuous perception of the whole environment and real-time and decision-based control of the process. By providing a rich conceptual model with implementation architecture, our work contributes a novel approach for adaptive process management practice. Compared with other related approaches based on static routing and limited alternatives, this work is characterized by extension from process logic to business logic, continuous perception of the process environment, and decision-based control of the business process.

One limitation of this approach arises from the complexity of business logic, such as identification of business logic from business practice, correct representation of rules in a business sense, and maintenance of rules. In other words, this approach was developed on the assumption that business logic for process management is accurately defined and represented.

Acknowledgement

The authors thank the editors and reviewers of *Information & Management*, and Professor Doug Vogel for their constructive comments on this paper. This research is supported by a UGC CERG research grant (No. 9040823) from the Hong Kong Government and a Strategic Research Grant (No. 7001309) from City University of Hong Kong.

Reference

- [1] W.M.P. van der Aalst, Loosely Coupled Interorganizational Workflows: Modeling and Analyzing Workflows Crossing Organizational Boundaries, *Information and Management* 37(2), 2000, pp.67-75.
- [2] W.M.P. van der Aalst, A. Kumar, XML-Based Schema Definition for Support of Interorganizational Workflow, *Information Systems Research*, 14(1), 2003, pp.23-46.
- [3] A. Basu, A. Kumar, Research Commentary: Workflow Management Systems in e-Business, *Information Systems Research* 13(1), 2002, pp.1-14.
- [4] M.B. Blake, H. Goma, Agent-oriented compositional approaches to services-based cross-organizational workflow, *Decision Support Systems*, 40(1), 2005, pp.31-50.
- [5] J. Giarratano, G. Riley, *Expert systems: principles and programming*, PWS Publishing Company, Boston, 1998.
- [6] G. Gorry, M.S. Morton, A Framework for Management Information Systems, *Sloan Management Review*, 12(2), 1971, pp.55-70.
- [7] P. Grefen, B. Pernici, G. Sanchez, *Database support for workflow management: the WIDE project*, Kluwer Academic Publishers, 1999, Boston, Mass.
- [8] M. Hammer, J. Champy, *Reengineering the Corporation: A Manifesto for Business Revolution*, 1993, Brealey, London.
- [9] N. R. Jennings, P. Faratin, T. J. Norman, P. O'Brien, B. Odgers, Autonomous Agents for Business Process Management, *Int. Journal of Applied Artificial Intelligence*, 14 (2), 2000, pp.145-189.
- [10] P. J. Kammer, G. A. Bolcer, R. N. Taylor, A. S. Hitomi, M. Bergman, Techniques for Supporting Dynamic and Adaptive Workflow, *Computer Supported Cooperative Work*, 2000, 9(3-4), pp.269-292.
- [11] M. Klein, C. Dellarocas, A. Bernstein, Introduction to the Special Issue on Adaptive Workflow Systems, *Computer Supported Cooperative Work*, 9(3-4), 2000, pp.265-267.

- [12] M. Klein, C. Dellarocas, A Knowledge-based Approach to Handling Exceptions in Workflow Systems, *Computer Supported Cooperative Work*, 9(3-4), 2000, pp.399-412.
- [13] F. J. Lerch, D. E. Harter, Cognitive support for real-time dynamic decision making, *Information systems research*, 12(1), 2001, pp.63-82.
- [14] K. Linderman, K.E. McKone-Sweet, J.C. Anderson, An integrated systems approach to process control and maintenance, *European Journal of Operational Research*, 164(2), 2005, pp.324-340.
- [15] J. Liu, S. Zhang, J. Hu, A case study of an inter-enterprise workflow-supported supply chain management system, *Information & Management*, 42, 2005, pp.441-454.
- [16] F.P. Maturana, P. Tichý, P. Slechta, F. Discenzo, R.J. Staron, K. Hall, Distributed multi-agent architecture for automation systems, *Expert Systems with Applications*, 26(1), 2004, pp.49-56
- [17] N. Melão, M. Pidd, A conceptual framework for understanding business processes and business process modelling, *Information Systems Journal*, 10, 2000, pp.105–129.
- [18] T. Morgan, *Business rules and information systems*, Addison-Wesley, New York, 2002.
- [19] N.C. Narendra, Flexible Support and Management of Adaptive Workflow Processes, *Information Systems Frontiers*, 6(3), 2004, pp.247–262.
- [20] A. Newell, *Unified theories of cognition*, Cambridge, Mass., Harvard University Press, 1990.
- [21] P.D. O'Brien, W.E. Wiegand, Agent based process management: applying intelligent agents to workflow, *The Knowledge Engineering Review*, 13(2), 1998, pp.1-14.
- [22] W.W. Reeves, *Cognition and complexity: the cognitive science of managing complexity*, Lanham, Md., Scarecrow Press, 1996.
- [23] H.A. Simon, *The new science of management decision*, Englewood Cliffs, N.J., Prentice-Hall, 1977.
- [24] C. Varela, G. Agha, A hierarchical model for coordination of concurrent activities, *Lecture Notes in Computer Science*, 1594, 1999, pp.166-182.

- [25] M. Wang, H. Wang, D. Xu, K.K. Wan, D. Vogel, A Web-service Agent-based Decision Support system for Securities Exception Management, *Expert Systems with Applications*, 27(3), 2004, pp.439-450.
- [26] M. Wang, H. Wang, Intelligent Agent Supported Business Process Management, *Proceedings of 38th Hawaii International Conference on System Science (HICSS-38)*, 2004, Hawaii, US.
- [27] M. Wang, H. Wang, D. Xu, The Design of Intelligent Workflow Monitoring with Agent Technology, forthcoming in *Knowledge-based Systems*, 2005.
- [28] M. Weske, W.M.P. van der Aalst, H.M.W. Verbeek, Advances in business process management, *Data & Knowledge Engineering*, 50 (1), 2004, pp.1-8.
- [29] M. Wooldridge, *An introduction to multiagent systems*, 2002, England, J. Wiley
- [30] H. Zhuge, J. Chen, Y. Feng, X. Shi, A federation-agent-workflow simulation framework for virtual organization development, *Information & Management*, 39, 2001, pp.325-336.
- [31] H. Zhuge, Workflow- and agent-based cognitive flow management for distributed team cooperation, *Information & Management*, 40, 2003, pp.419-429.



Minhong Wang (Maggie) is a senior research associate at the Department of Information Systems at City University of Hong Kong, from which she obtained her PhD in Information Systems in 2005. Dr. Wang's research interests are in the areas of agent oriented software engineering, business process / workflow management, supply chain management, and knowledge-based systems. She has published papers in *Expert Systems with Applications*, and *Knowledge-based Systems*. She has also presented several papers at international conferences, such as HICSS, CAiSE, AMCIS, PRICAI, among others. She can be reached at maggie_mh_wang@yahoo.com.



Huaiqing Wang is a professor in the Information Systems Department at the City University of Hong Kong. He specializes in research and development of business intelligence systems, intelligent agents and their applications (such as multi-agent supported financial information systems, virtual learning systems, knowledge management systems, and conceptual modeling). He received his PhD in computer science from the University of Manchester in 1987.