Inf Syst Front (2015) 17:1253–1264
DOI 10.1007/s10796-015-9565-2

Workflow-aware attention tracking to enhance collaboration management

Shaokun Fan[1] **· Lele Kang**[2] **· J. Leon Zhao**[3]

Published online: 30 May 2015

Abstract Current Workflow Management Systems only capture the start time and end time of tasks and it is difficult for managers to track how much time is actually spent on a particular task if multiple tasks are performed concurrently. Thus, managers may have problems accurately arranging tasks and evaluating costs. In order to facilitate collaboration management in multitasking environments, we propose a workflow-aware attention tracking framework that integrates the concept of attention with workflow models. We first design the meta-model of workflow-aware attention tracking. We further derive seven rules for attention tracking in multitasking collaboration using Object Constraint Language. Based on the rules and meta-model, we develop a procedure for attention rule verification, and present an architecture for workflow-aware attention tracking systems. This research contributes to workflow management by adding attention awareness for collaboration management in multitasking environments.

Keywords Attention tracking · Collaboration management · Workflow management system · Meta-model · Accountability

✉ Lele Kang
llkang3-c@my.cityu.edu.hk

Shaokun Fan
sfan@wtamu.edu

J. Leon Zhao
jlzhao@cityu.edu.hk

1 College of Business, West Texas A&M University, Russell Long Blvd, Canyon, TX, USA

2 School of Information Management, Nanjing University, Nanjing, China

3 Department of Information Systems, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Hong Kong

1 Introduction

The increased globalization and vertical disintegration of large organizations means also that many tasks are accomplished by virtual teams (Majchrzak et al. 2000; Smith and McKeen 2011). Although today's workflow management systems (WFMSs) support defining, managing, and executing business processes in collaborative work teams (Zeng and Zhao 2005), they often assume that one actor is working on a single task at a time (Spink et al. 2008). In reality, each task may be collaboratively accomplished by several participants and a worker may engage in several tasks, which are dependent on each other (Zhao et al. 2006). Participants work cooperatively to achieve the common goal. In such a multitasking environment, it is usually difficult to track how much effort team members spend on a particular task because people may work on several tasks that may span over several projects simultaneously.

In order to address the issues in managing multitasking collaboration, we propose a workflow-aware attention tracking approach, which captures the attention resource paid to an activity by a particular person. Psychologists have intensively studied human attention. Some of them view attention as a set of cognitive and perceptual processes (Norman 1968) that guide individuals to react to environmental stimuli (Roda and Thomas 2006), while others treat attention as a type of cognitive resource that guides actions (Davenport and Beck 2001). In this paper, we define attention as an important cognitive resource that enables people to act, reason, and communicate when performing activities to accomplish tasks (Nabeth 2008). One's attention to a task can be described by time and percentage of cognitive capability that are spent on the task. Since attention is a necessary resource for performing any activity, the attention resource allocated to a task provides a good indicator of the effort that one puts into a particular task.

Attention allocation information provides managers with an effective way to monitor the real-time progress of tasks even when task deliverables are not ready. In addition, different from prior discussion about human or machine resource allocation (Delias et al. 2011), attention resources have unique features such as the attention deficit problem (Davenport and Beck 2001).

Though existing information systems such as WFMSs usually record the start and end time of tasks and the people who are assigned to each task, it is very difficult to calculate how much time and effort is actually spent on a particular task if multiple tasks are performed at the same time. For example, if Task A and Task B are assigned to Tom at the same time and he finishes the two tasks 2 h later, it is hard to know how Tom distributed his time and effort between the tasks. Note that this is only the simplest example of multitasking in teams and real cases would be much more complicated. The missing information of Tom's time and effort distribution means collaborators have no ability to monitor and audit task progress, and making it difficult for managers to manage projects effectively.

The basic assumption of WFMSs is that a workflow will transit from one state to another when a task is completed; otherwise, nothing happens. WFMSs usually do not capture what has been done to accomplish a task. Accounting for individual time and effort in multitasking environments may cause several problems:

1. Managers cannot accurately estimate the cost for each task or project because workers may work on other tasks as well during the same time period.
2. Managers may not be able to monitor progress effectively because they do not get updated progress information until the task is completed (e.g., deliverables are submitted). Unexpected difficulties in one task may affect the outcome of all projects that the worker is involved in.
3. The productivity of workers may be affected if they work on too many concurrent tasks, due to switching between tasks and interruptions.

In this paper, we propose a workflow-aware attention tracking framework to facilitate multitasking collaboration management. We use UML to represent the meta-model and Object Constraint Language (OCL) to define the constraints of attention tracking. Our study tries to improve current WFMSs by enabling attention tracking to improve accountability in multitasking collaborative work teams. The attention tracking function treats attention as a unique cognitive resource of actors that is necessary for performing activities in workflow tasks. Attention allocated to an activity can be described by the time period and percentage of cognitive capacity, which could be self-reported or captured by tracking technologies such as visual tracking and keystroke logging. We also derive seven rules for attention rule verification in multitasking collaboration. Finally, we propose a basic architecture for an attention tracking system based on the attention tracking meta-model and rules. The remainder of the paper is organized as follows. Section 2 reviews related literature. In Sections 3, we describe attention allocation problem in multitasking collaboration. In Section 4, we formally define the workflow-aware attention meta-model and derive rules for attention tracking in multitasking collaboration in Section 5. In Section 6, we propose an architecture of our workflow-aware attention tracking systems. We give examples of attention tracking in multitasking collaboration and apply principles in Section 7. Finally, the conclusion discusses contributions and limitations.

2 Literature review

2.1 Attention

Collaboration requires a group of people to engage in a task and complete the task during the required time period. Attention is one of the key resources for collaborators to perform any actions to accomplish a task. Davenport and Beck (2000) found that managing human attention significantly influenced the performance of the collaboration team. Prior studies interpreted attention from both process and resource perspectives.

From the process perspective, attention is generally considered as a set of processes that guide the selection of environmental stimuli to be attended to (Roda and Thomas 2006). It emphasizes the set of processes by which collaborators select related information and then decide to allocate their cognitive resources (e.g., memory) on a particular subject (Nabeth 2008). For example, attention is developed from this perspective in Human-Computer Interaction (HCI) research, and the design of attention systems focuses on supporting human attention processes (Vertegaal 2003).

From the resource perspective, attention is a part of human cognitive and perceptual resources. In knowledge-intensive environments, attention is an especially scarce resource (Hansen and Haas 2001). In a collaborative project, collaborators need to allocate their attention resources to tasks that they participate in. Usually, more difficult tasks require more attention resources from collaborators. The resource perspective is also adopted in organizational research, which states that the limited attention resource of humans results in their bounded capability to perform activities (Ocasio 1997). Attention has been found to be closely related to task performance (Davenport and Beck 2001). Existing literature shows that the performance of collaborators might be compromised if too many tasks are competing for attention resources from a person (Hansen and Haas 2001; Ocasio 1997).

In this research, we define attention as a type of cognitive resource required for people to conduct activities to accomplish assigned tasks (Davenport and Beck 2000, 2001; Nabeth 2008). Different from cognitive resources such as long-term and short-term memory, attention resource can be measured from two dimensions: time and percentage of cognitive capacity, which could be reported by people themselves. Time evaluates how long individuals allocate their attention resource to the tasks. Percentage of cognitive capacity assesses the degree or level of attention allocated to a particular task. Attention describes how much cognitive effort one puts into an activity during a specific time period. Percentage of cognitive capacity can also be automatically captured by technologies (e.g., keystroke logging or visual tracking).

2.2 Workflow technology

Workflow is the automation of a business process (Fan et al. 2012). Basu and Kumar (2002) summarized three predominant views of workflows. First, a process is regarded as the combination of individual tasks which are conducted by various actors within organizations. The integration of tasks represents a workflow for the business process. Second, workflow is the automatic operation of a business process, during which information, documents, or tasks are transferred from one actor to another for execution, according to the defined procedural rules. Third, workflow is defined as a particular type of process that has many characteristics, such as predictability, repeatability, distributed and automation. Regardless of which definitions, we find that a workflow is composed of various tasks which are executed by different actors with the consumption of different resources under the predefined rules.

Process design is the fundamental step of workflow management. In process design, the tasks and their mutual relationships (e.g., task hierarchy and dependency) are analyzed. Most existing WFMSs use roles to manage task assignments (Bui and Sankaran 2001). All roles in a team should be identified before executing the tasks. As a metaphor for the individual, roles define which actors are going to execute the assigned set of tasks. The benefits of role definition can be seen from different aspects. The role not only represents responsibility of actors for a set of tasks, but also shows their expertise. For instance, the database manager should have more experience and knowledge with the database than the interface designer in a software development team. Aalst and Kumar (2001) proposed two dynamic mechanisms to assign workflow tasks to individuals, namely the push approach and the pull approach. In the push approach, WFMSs automatically assign the tasks to the specific actors based on the defined rules and process. In the pull approach, the same task is offered to different actors, then the actors decide whether they will accept it. WFMSs identify the actors who have the related expertise to conduct the tasks and then check their schedules to determine availability.

Traditional WFMSs only record which tasks are assigned to which actors and managers cannot accurately estimate cost of effort for a particular task (Wynn et al. 2013). Further, it is very difficult for managers to monitor the progress of tasks because they have no information about ongoing tasks until the tasks is completed. Our proposed workflow-aware attention tracking framework addresses the problem by adding attention as a unique resource for workflow management, which is rarely examined in prior workflow literature.

3 Attention in multitasking collaboration

Two unique characterisitcis of attention are very important for multitasking collaboration: concurrency and switching frequency (Salvucci et al. 2009) (see Table 1). When participating in collaboration, the attention of actors may be at either low or high level of concurrency (or switching frequency). The frequency of switching represents how often individuals need to switch their attention from one task to another. In a multitasking context, people need to switch and resume tasks frequently (Altmann and Gray 2008). Switching one task to another requires participants to rehearse the problem representation and recall the task context (Altmann and Trafton 2007). Thus, this switching process reduces work productivity and engages actors in a high-level cognitive load (Kim and Kim 2011). The task switching cost and interruption overheads have been well established in the literature on attention models (Salvucci et al. 2009). The switching cost may vary considerably depending on the nature of tasks (e.g., similarity among tasks and complexity). The switching cost is closely related to the way that multiple tasks are represented and arranged (Kim and Kim 2011). The similarity of problem representation and task context significantly influences their switching cost. Switching from different tasks towards the same or similar goals will be easier than switching to different types of tasks that require different knowledge.

Concurrency represents how many tasks collaborators need to engage in during a specific time period (Patten et al. 2004). People can pay attention to multiple tasks simultaneously. For instance, a project manager can write a report while he participates in a remote meeting. In low-level concurrency, people work on a single task at the same time. The concentrated attention enables them to fully engage in one particular task. High-level concurrency requires people to work on different tasks at the same time. Similiar to switching frequency, too many concurrent tasks may distract attention allocation and reduce productivity of collaborators. Thus, people who work on tasks that require intensive knowledge should try to avoid high concurrency of attention.

Table 1 Concurrency and Switching Frequency of Attention

	Low concurrency	High concurrency
Low switching frequency	An actor works on one task at a time. A new task will be assigned only when the current task is finished. Example: Tom participates in a remote meeting to discuss a project plan from 9:00 am to 11:00 am.	An actor works on several tasks at the same time. Example: Tom writes a customer requirement report while he participates in a remote meeting to discuss a project plan from 9:00 am to 11:00 am.
High switching frequency	An actor has many tasks assigned to him. He works on one task at a time but needs to switch from task to task frequently. Example: Tom interviews customers from 9:00 am to 10:00 am and 11:00 am to 12:00 pm. From 10:00 am to 11:00 am, he participates in a remote meeting to discuss a project plan.	An actor has many tasks assigned to him. He works on several tasks at a time. He also needs to switch from task to task frequently. Example: Tom interviews customers from 9:00 am to 10:00 am and 11:00 am to 12:00 pm. From 10:00 am to 11:00 am, he participates in a meeting to discuss a project plan and writes a customer requirement report.

The unit of attention can be measured at different levels of granularity, such as months, days, hours, and even seconds. The granularity of attention assessment should be chosen based on the project demands. For instance, if we report attention allocation by days in a several-month project, we have enough information to know what collaborators do every day. It will be feasible for team leaders to track task progress and audit attention allocation of team members. However, if attention allocation is measured with a finer granularity (e.g., hourly), it is costly to collect and analyze the attention allocation data. Therefore, practitioners need to choose the appropriate level of granularity according to the practical requirement of projects.

Figure 1 describes the scenario of attention allocation in multitasking context. In WFMSs, the project is hierarchically separated into different tasks, which are further decomposed into activities. Activites are performed by different actors. Each activity may require participation of multiple actors, while each actor may need to accomplish several activities (Russell et al. 2005). Different activities will compete for attention of actors. As stated Fig. 1, Actor 1 needs to participate in three activities (i.e., Activity 1,2, and 3) during the time period of (1,3). Compared to Actor 2, Actor 1 works on three tasks during (1,2) and engage in a high level of concurrency. In this scenario, Actor 2 could conduct Activity 4 after he/she finishes Activity 3. Actor 2 only needs to switches his attention once (e.g., switch from Activity 3 to Activity 4 at time point 3). Actor 1 has to switch among three different activities and has a high level of switching frequency in this project. However, attention allocation of actors are not captured by existing WFMSs and it is very difficult for managers to know how much effort Actors 1 and 2 spend on each task. Next, we will introduce the work-aware attention tracking approach to facilitate collaboration management in multitasking contexts.

4 Workflow-aware attention tracking meta-model

Figure 2 shows the attention tracking meta-model. It extends the existing workflow meta-models (Aalst and Kumar 2001; Chiu et al. 1999; Rosemann and Muehlen 1998) by adding the concept of attention. The classes in the meta-model are introduced in the rest of this section.

Workflow process is case-based, which means that every instance of workflow is executed for a specific case (Aalst 1998). A case of workflow usually solves a project. Examples of cases are customer complaint, purchase order, tax declaration, and so on. Cases are usually initiated by external or internal customers. The goal of workflow management is to standardize the process of case handling and handle cases as efficiently as possible. The most important aspect of a workflow model is tasks and relationship among tasks, which are usually represented as flow diagrams in the workflow management literature (Georgakopoulos et al. 1995). Many cases can be handled by following the same workflow model. For example, a customer order handling workflow can handle different customers who may purchase different items. Each customer order can be treated as a case.

Definition 1 A case is an instance of a problem that requires execution of the workflow.

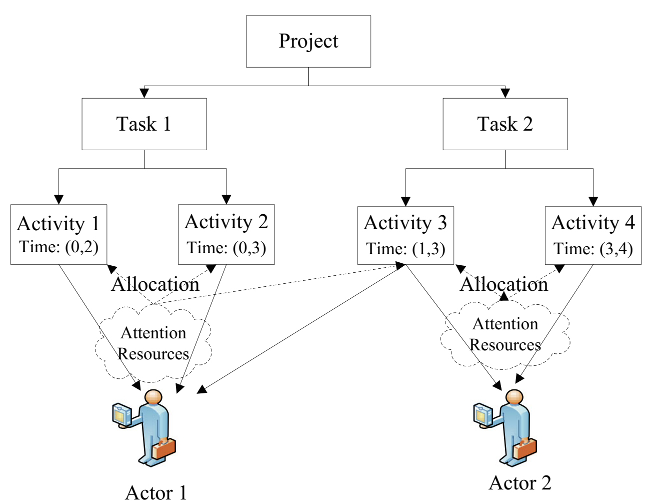

Fig. 1 Task Assignment and Attention Allocation in Collaborative Work Teams

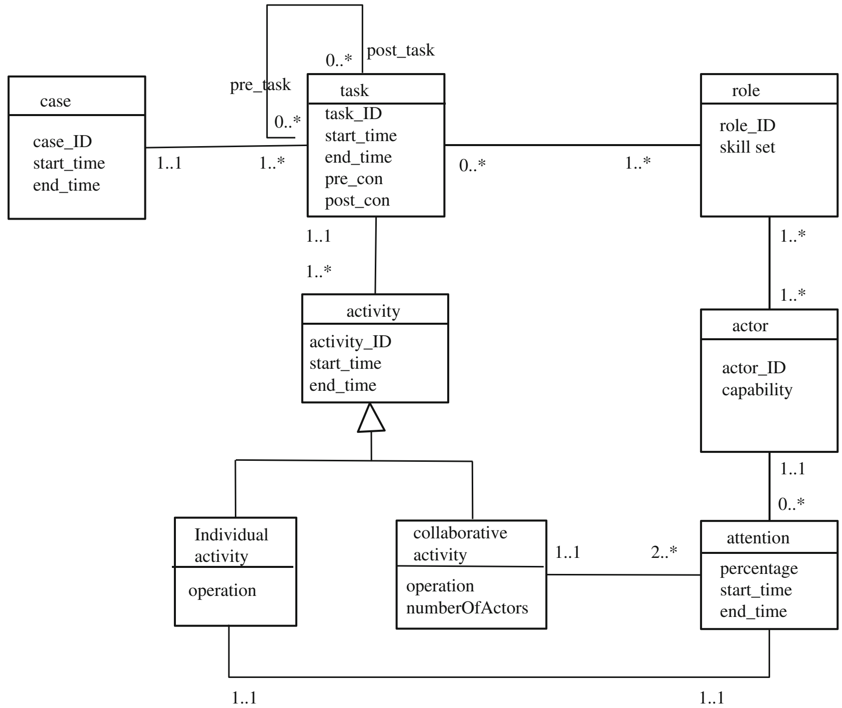

Fig. 2 Workflow-aware Attention Tracking Meta-model

Tasks in workflow models are modeled as nodes, and the dependency relationships are modeled as links among nodes. Activities and tasks are both used to represent workflow steps in the literature. But they emphasize different perspectives. The theory of organizational routines conceptualizes routines as consisting of ostensive aspects and performative aspects (Feldman and Pentland 2003). The ostensive aspect is the ideal or schematic form of a routine. It is the abstract, generalized idea of the routine, or the routine in principle. The performative aspect of the routine consists of specific actions, by specific people, in specific places and times. It is the routine in practice. Both of these aspects are necessary for an organizational routine to exist. Aalst and Kumar (2001) use activity and task to represent these two perspectives. One major difference between activities and tasks is that an activity is the smallest unit of work while tasks can be decomposed into activities.

Definition 2 A task is a unit of work that transforms input information objects into output information objects.

Definition 3 An activity is an instantiation of a task and corresponds to the smallest unit of actual execution of a task. If an activity is only performed by one person, then it is an individual activity. Otherwise, it is a collaborative activity.

Actors in organizations are people who perform activities to solve problems. Each actor has the capability, knowledge, and skills to solve certain problems in the organization. Actors in an organization can be classified by the roles they play. A role can be used to represent the required skills and capabilities for that position. One actor may have multiple roles, and several actors may share the same role. For example, the role can be a clerk, and the actors qualified to perform that role are Mary, Tom, and Bill. Note that, in general, a given actor may be qualified to perform one or more roles and a role should have at least one qualified worker. The workflow system maintains a mapping from roles to actors.

Definition 4 An actor is a human that can perform activities.

Definition 5 Role is a generic identifier for a group of workers with the required skills.

Attention belongs to actors in an organization and is needed for performing any activities. Attention allocated to an activity can be described by the time period and percentage of cognitive capacity, which is used to estimate the amount of attention spent on an activity. Since the measurement of attention resource is difficult, we can use several indicators to estimate how much attention is allocated to a particular activity. For example, user self-reported percentage of attention is usually a good indicator because the user knows better than anyone else how much cognitive resource is spent on which activity. Other potential indicators include keystroke logging and visual tracking. For instance, the document monitoring functions could report how many words each writer has typed. The detailed technology for attention indicators is out of the scope of this paper. *Definition 6. Attention* is the cognitive resource of actors, which is needed when performing activities. Attention paid to an activity is described by its starting time, ending time, and percentage of cognitive capacity as (percentage, (start_time, end_time)).

For example, from 8:00 am to 9:00 am, a worker can spend 50 % of his attention on participating in a remote meeting and the other 50 % of his attention on writing a report. The worker's self-estimation could represent how he distributed his attention among different tasks. We describe this attention distribution with the following expression: attention paid to meeting (0.5, (8:00, 9:00)), and attention paid to writing a report (0.5, (8:00, 9:00)).

Figure 2 also list some representative attributes. Note that these attributes are by no means exhaustive. We only listed some illustrative attributes and those that are considered to be directly related to attention awareness in the meta-model.

5 Rules of attention tracking in multitasking collaboration

The attention tracking meta-model does not express constraints that need to be satisfied in workflow management. In this section, we will model constraints of attention tracking in Object Contraint Language (OCL) (Aalst and Kumar 2001). OCL started as a complement to the UML notation with the goal of overcoming the limitations of UML in terms of precisely specifying detailed aspects of a system design (Cabot and Gogolla 2012). OCL uses invariants to state all necessary conditions that must be satisfied in each possible instantiation of the model.

The meta-model in Fig. 2 shows cardinality constraints among classes. For example, each case may have multiple tasks and each task can be performed by multiple activities. However, the cardinality constraint cannot precisely define the requirement for attention tracking in multitasking collaboration. For example, a collaborative activity requires three participants. The cardinality constraint only specifies that multiple attention resources are allocated to the activity. This rule may not be precise since the exact number of attention to be allocated to the activity. Based on the attention concept we discussed in Section 3, we propose seven rules in this section. Note that the rules proposed in this paper are not exhaustive. Future work can explore other aspects of attention (such as team attention) and propose more rules for attention tracking in multitasking collaboration.

Rule 1 (Minimum time unit of attention) Attention allocated to perform a certain activity should last for at least a minimum amount of time before the actor switches to other activities.

$$(R1)\quad Context\ :\ attention$$
$$self.End\_time - self.Start\_time > c$$

Here, c is a predefined minimum unit of continuous working time. Integrity constraints in OCL are represented as invariants defined in the context of a specific type, named the context type of the constraint. Its body, the Boolean condition to be checked, must be satisfied by all instances of the context type. The self variable represents an arbitrary instance of the context class. As discussed in Section 3, switching attention from one activity to another requires cognitive cost. Thus, an actor must allocate attention resource that is more than the switching cost to perform an activity. The constant will be different depending on the complexity of tasks. For complicated cases that require intensive knowledge and high cognitive load, more time will be needed. The minimum time unit of attention can also be chosen for practical concerns. For instance, it will be very costly to keep track of a one-year project in terms of hours. Defining the minimum time unit of attention could reduce complexity of attention tracking.

Rule 2 (Compatibility with case schedule) All actors must allocate their attention resource to the activities within the start and end time of the workflow case.

$$(R2)\ Context:\ case$$
$$self.task.activity.attention->forAll\,(a|a.start\_time >= self.start\_time\ and\ a.end\_time <= self.end\_time)$$

Another important concept of OCL is collections. The arrow "->" is used to access collections. forAll specifies constraints that should be satisfied by all elements in a collection. self.task gives all tasks for a workflow case. Similarly, self.task.activity.attention gives all attention resources allocated to a case. The forAll expression tests whether the time periods of all attention allocated to tasks belonging to a case fall into the range of the case.

Rule 3 (Limited attention resource) In any minimum time unit of attention, the addition of one person's attention percentage is equal to or less than 1. We use *st* and *et* as the start time and end time of any minimum time unit and *st-et=c*.

$$(R3)\ Context:\ actor$$
$$self.attention->select\ (start\_time=<st\ and\ end\ time>=et)->collect(percentage)->sum()<=1$$

Assume attention is measured by the percentage of total available cognitive resources, in any time periods, the total percentage of one's attention should be less than or equal to 100 %. The select (start_time=<t1 and end time>=t2) expression selects all the attention allocation that includes the time period. The collect expression extracts the variable of percentage from all selected attention allocation. Sum () is used to calculate the total percentage of attention from *st* to *et*.

Rule 4 (Appropriate stress level of concurrency) In any minimum time unit of attention, an actor should NOT work on more than m activities, where m is the maximum level of concurreny. We use *st* and *et* as the start time and end time of any minimum time unit and *st-et=c*.

$$(R4)\ Context:\ Actor$$
$$self.attention->select\ (start\_time<st\ and\ end\ time>et)->size()<m$$

Rule 4 specifies that an actor can allocate attention to limited activities at the same time. It is true that people can work on a report while attending conference. But the cognitive limit of human beings constrains the number of activities people can get involved in at the same time. The number *m* depends on different organizational contexts and individual ability.

Rule 5 (Appropriate stress level of switching frequency) From *t1* to *t2* (*t1* and *t2* are any two time points and *t1*<*t2*), an actor should NOT work on more than *f sequential* activities, where *f* is the maximum number of switching frequency.

$$(R5)\ Context:\ Actor$$
$$self.attention->select\ (start\_time=<t1and\ end\_time>=t2)->size()<=f*(t2-t1)$$

Attention stress is related to both the concurrency level and the switching frequency level. Rule 5 further specifies the total amount of sequential activities that an actor needs to participate in within a time period. Conformance to this rule guarantees that actors do not need to switch their attention from one activity to another very frequently. *f* is the maximum number of switching frequency and *f*(t2-t1)* represents the total number of activities assigned to an actor from *t1* to *t2*. Similar to *m*, the number *f* also depends on different organizational contexts and individual ability. The switching frequency is also influenced by the similarity among tasks. The switching frequency could be higher if actors engage in similar tasks rather than heterogeneous tasks.

Rule 6 (Collaborative attention) For a collaboration activity, the number of actors who pay attention to the activity should be equal to the number of actors required for the activity.

$$(R6)\ Context:\ collaborative\ activity$$
$$attention.actors->size()=self.numberOfActors$$

Workflow activities are classified into two types, namely individual activity and collaborative activity. For a collaborative activity, collaborative efforts are required to achieve the goal of the activity. This rule guarantees that each collaborative activity is conducted in a collaborative way in which the required number of people pay their attention to the activity.

Rule 7 (Attention allocation based on task dependency) Attention allocated to tasks should be consistent with task dependencies.

$(R7)\ Context : attention$

$$self.allinstances()->forAll(a1, a2|a1.start\_time <= a2.end\_time implies a1.activities.tasks.pre\_task->excludes(a2.activities.tasks))$$

If Task A depends on Task B (A needs input from B), then actors should first pay attention to Task A and then pay attention to Task B. In Rule 7, we define that if two attention resources (a1 and a2) are allocated to two tasks and a1 starts before a2 ends, then a2's corresponding task cannot be the pre_task of a1's corresponding task. The time relationship of attention resources should be consistent with task dependency. Violation of this rule can lead to idle time in the process and reduce the time efficiency of the whole case.

Based on the rules defined in this section, we propose a procedure to verify attention allocation of actors in a workflow execution log. The procedure takes workflow tasks, dependencies among workflow tasks and attention allocation as input and check whether the attention allocation violates any of the rules. Figure 3 shows the pseudocode of the procedure of attention rule verification. It can be used to audit a workflow task execution log and help managers track the attention status of workers.

6 System architecture of a workflow-aware attention tracking system

Based on the model and rules of attention tracking, we design the architecture as shown in Fig. 4. It consists of a user interface, a WFMS, attention monitoring, and attention verification. In this section we discuss how to integrate WFMSs and attention modules to facilitate multitasking collaboration.

According to the workflow reference model, WFMSs should automate business processes by managing the sequence of tasks and the invocation of appropriate human and IT resources associated with the various activity steps (Rinderle et al. 2004). It also helps to route documents, information, or tasks between participants according to a defined set of rules. A typical WFMS (such as jBPM, Lotus workflow) usually supports the whole lifecycle of process management, such as process definition and modeling, initiating a process instance, task assignment, routing information among tasks, enabling or suspending tasks, and so on. The attention tracking system is based on existing WFMSs and integrates attention monitoring and verification functions. WFMSs maintain the workflow definition database that stores rules for coordinating workflow tasks, while the attention monitoring and verification modules maintain an attention status database that keeps track of all participants' attention allocation. Once a workflow is initiated, actors will perform activities to accomplish tasks defined in the workflow.

The attention monitoring module captures the attention allocation of actors on activities and store attention allocation data in the attention status database. In traditional WFMSs, the manager cannot know how much effort one has spent on a task until the task is completed. However, actors may have performed most of the activities for accomplishing the task but have no result to report to the system yet. The attention monitoring module makes this progress information available to managers. Attention-based monitoring allows managers to track the real-time status of the task execution. It provides information about how much attention each individual allocates to activities associated with a specific task. Collecting attention data may increase the technical complexity and human cognitive load since collaborators need to spend some of their attention resources on reporting the required attention data. However, there are two potential methods to address this issue. The first one is that the team leader should control the granularity of attention self-reporting. Coarse granularity can reduce the frequency of recording and submitting data. Second, there are some technologies that could help collect attention status automatically. For instance, some software could collect data about how much time people spend on a particular document. This is a tradeoff in which we need to balance the cost and benefit of attention data reporting.

The attention verification module will use attention data in the database and automatically check whether the attention status of a workflow violates any of the predefined rules according to the procedure of attention rule verification. Managers can be notified by the system for situations of inappropirate attention allocation, such as high attention stress level or inappropriate allocation of attention to tasks. Although it might be true that multiple WFMSs may be used for managing projects in companies, we only support one WFMS in the system architecture. The integration of multiple WFMSs with attention tracking modules may not be trivial and may rely on techniques such as enterprise service bus. This will be part of our future work.

Fig. 3 The Procedure of Attention Rule Verification

- **For** all attention A= (percentage, (start_time, end_time))
  - **If** *A.end_time-A.start_time<c*
  - **Then** indicate a violation of minimum time unit of attention (Rule 1)
  - **If** (A.start_time<Project.start_time or A.end_time>Project.end_time)
  - **Then** indicate a violation of compatibility with case schedule (Rule 2)
- **For** all attention allocation A= (percentage, (start_time, end_time)) that belongs to a worker W
  - **For** all time unit (t, t+c)
    - **If** A.start_time<=t and A.end_time>=t+c
      - Calculate the summation (*SUM*) all A.percentage
      - Count the number of (*Concurrent_Attention_ Count*) of attention
  - **If** *SUM*>1
  - **Then** indicate a violation of limited attention resource (Rule 3)
  - **If** *Concurrent_Attention_ Count > m*
  - **Then** indicate a violation of appropriate stress level of concurrency (Rule 4)
  - Count the number (*Switch_Attention_Count*)of attention that belongs to W
  - **If** *Switch_Attention_Count >f*(* Project.end_time - Project.start_time*)*
  - **Then** indicate a violation of appropriate stress level of switching frequency (Rule 5)
- **For** all collaborative activity,
  - Count then number of attention associated with the activity,
  - **If** counter != activity.numberOfActors,
  - **Then** indicate a violation of collaborative attention (Rule 6)
- **For** any two activities (A and B) in the workflow that A precedes B
  - Find the maximum end_time of attention (*ME*) associated with A
  - Find the minimum start_time of attention (*MS*) associated with B
  - **If** *ME>MS*
  - **Then** indicate a violation of attention allocationbased on tasks dependency(Rule 7)

7 An example of attention tracking for software development

An example is introduced in this section to illustrate how the workflow-aware attention tracking framework can be used in multitasking collaboration. Suppose a software engineering team works on a software development project. The team consists of one project manager (PM) and three developers (D1, D2, and D3). The workflow of software development can be divided into three sequential tasks.

1. Task 1

   Requirement Analysis. At the beginning, developers need to visit the customers to collect their requirements. Each of them needs to interview one customer. The manager will monitor the customer interview and intervene when needed. Thus Task 1 is executed by three activities.
2. Task 2

   Project Planning. Based on the user requirements collected in Task 1, all team members have to meet together to discuss budget, schedule, and task assignment. A report should be submitted as the result of this task. Assume that we need four activities to finish this task: three meetings to decide budget, schedule, and task assignment, respectively, and one activity to compose the report.
3. Task 3

   Project Implementation. All developers carry out the project plan under the project manager's supervision.

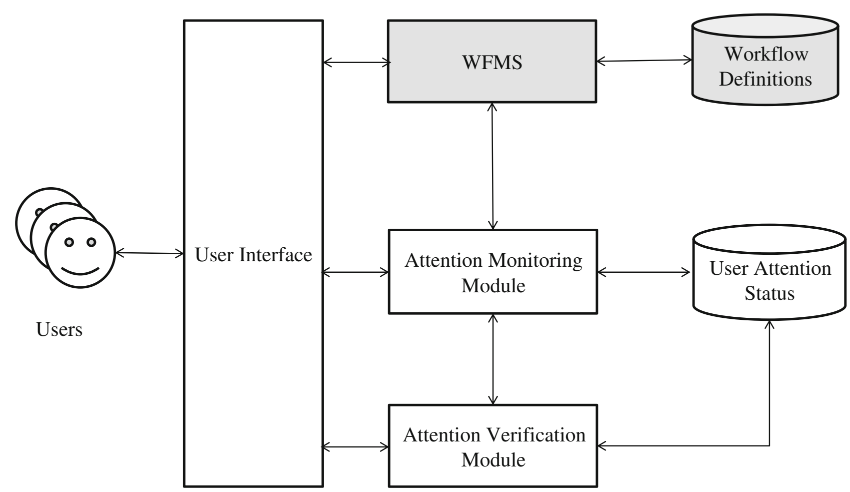

Fig. 4 Architecture of a Workflow-aware Attention Tracking System

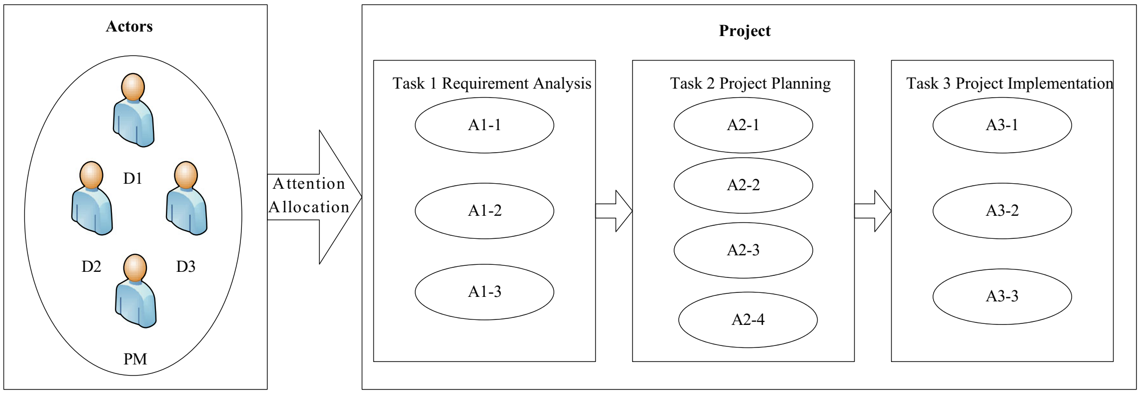

Fig. 5 Example of Software Development Project

Suppose we have three activities in this task to develop three modules of the software.

Figure 5 describes the workflow in the project and activities for each task. Table 2 gives the detailed descriptions of all activities in the project.

Suppose the attention allocation log is as shown in Table 3. Note that attention allocation is represented as a two-tuple: (percent of attention allocated, (start time, end time)), such as (1, (0,2)). The unit of time in Table 3 is day although it could be hour or minute in other cases.

Next, we will discuss how the seven rules we derived in Section 5 help attention monitoring and verification. We apply the procedure of attention verification to the case above to check whether the rules are violated.

Assume the minimum attention time unit for software development is 1 day. The team members cannot pay attention to an activity if they allocate less than one time unit on the tasks. The attention allocated on A2-4 by D1 does not meet this rule, because D1 only pays 0.5 time unit of attention to A2-4.

According to the execution log, D3 allocates his attention to the project from days 11 to 15. If the project is expected to be done before day 14, this period is out of our time frame of (0, 14). Thus, it is against Rule 2.

PM spends 1.2 (0.4+0.4+0.4) attention resources on A1-1, A1-2, and A1-3 in time period of (0, 2). This is against Rule 3 because the total attention allocation of PM exceeds a maximum value of 1.

Compared to the developers, the project manager has a higher concurrency. In time period of (1, 2) and (11, 15), he needs to pay attention to 3 tasks at the same time. If we define the appropriate concurrency as 2, attention allocation for the project manager in (0,2) and (11,14) violates Rule 4..

In the first 3 days of the project (from day 0 to day 2), all four actors only switch their attention once. But in time period (3,10), D1 and D2 switch their attention four times. The switching frequency for D1 and D2 from day 3 to day 10 is once every other day. If the switching frequency should be less than once every 3 days, then this case violates Rule 5.

For A3-3, three actors are required. Actually, only D3 and PM pay attention to the activity. Therefore, Rule 6 is violated in this situation because the number of actors who pay attention to the activity is not equal to the number of actors required for the activity.

Table 2 Descriptions of Activities in the Software Development Case

Activities	Descriptions
A1-1	D1 interviews Customer 1 and PM monitors the activity.
A1-2	D2 interviews Customer 2 and PM monitors the activity.
A1-3	D3 interviews Customer 3 and PM monitors the activity.
A2-1	All members have a meeting to discuss the project budget.
A2-2	All members have a meeting to discuss the project schedule.
A2-3	All members have a meeting to discuss task assignments.
A2-4	D1 and D2 write a report for project planning.
A3-1	D1 and D2 develop module 1 and PM monitors the activity.
A3-2	D2 and D3 develop module 2 and PM monitors the activity.
A3-3	D1 and D3 develop module 3 and PM monitors the activity.

Table 3 An Example Attention Allocation Log

	D_1	D_2	D_3	PM
A1-1	(1,(0,2))			(0.4,(0,2))
A1-2		(1,(0,2))		(0.4,(0,2))
A1-3			(1,(0,2))	(0.4,(0,2))
A2-1	(1,(3,4))	(1,(3,4))	(1,(3,4))	(1,(3,4))
A2-2	(1,(5,6))	(1,(5,6))	(1,(5,6))	(1,(5,6))
A2-3	(1,(7,8))	(1,(7,8))	(1,(7,8))	(1,(7,8))
A2-4	(1,(9,9.5))	(1,(9,10))		(0.5,(9,10))
A3-1	(1,(11,14))	(0.4,(11,14))		(0.3,(11,14))
A3-2		(0.6,(11,14))	(0.4,(9,14))	(0.2,(11,14))
A3-3			(0.6,(11,15))	(0.3,(11,14))

From the case description, we could identify that Task 3 is dependent on Task 2. Thus, attention allocation to Task 3 should be after Task 2 is finished. However, D3's attention on A3-2 violates this rule because the start time of D3's attention on A3-2 is earlier than the end time of A2-4.

This example illustrates how workflow-aware attention tracking is applied to track and verify attention status in multitasking collaboration. This approach has three benefits for managers. First, it enables team managers to monitor the project progress in an appropriate level of granularity. In traditional WFMSs, managers cannot have the progress information until the end. Attention monitoring could report the task progress to managers while the task is not fully completed. Second, the listed rules could enhance project managers' ability to detect problems of attention allocation in the team. Project managers can track each actor's attention allocation and use the attenion verification rules to check attention status of actors. Third, this approach enables managers to accurately audit contributions of individual members in multitasking collaboration.

Due to space limitations, we only provide the example with one project. In addition, we only explain the multitasking scenario at the activity level. One worker may be involved in multiple tasks across cases or projects. Our approach can also be extended to multiple-project collaboration. We only have a few simple activities in the example. We can further decompose each activity in the example into sub-activities. For instance, the interview activity may be divided into sub-activities such as making an appointment with customers, conducting a field visit, writing the memos, and summarizing customer requirements. Then, the granularity of attention will be associated with the sub-activities and it may be meaningful to measure attention in hours for these sub-activities. Practically, the granularity of attention is determined by the requirement project. management.

8 Conclusion

This paper began with the observation that accurate work accounting is important in multitasking collaborative work teams and is not supported well in current WFMSs. The accountability issue also threatens effective management and teamwork productivity. We proposed a workflow-aware attention tracking framework to facilitate collaboration management in multitasking environments. As the first paper that proposes the attention tracking concept in multitasking collaboration, we claim three major contributions. First, we proposed a meta-model to formally operationalize attention and enable attention tracking. The meta-model represents the relationship among cases, tasks, roles, actors, activities, and attention. Second, seven essential rules were proposed to express constraints that need to be satisfied for attention tracking in workflow management. We also proposed a procedure to verify the seven rules. Third, we designed an architecture for attention tracking systems and describe the functions and techniques of different modules. We also used software development as an example to illustrate how the rules constrain multitasking collaboration. In such a context, team managers can benefit from this approach in three ways: (1) monitor task progress at a finer level of granularity; (2) detect violation of attention tracking rules; (3) accurately audit individual effort.

There are several limitations in this paper. The first limitation is that attention status tracking mainly relies on self-reported results. This may introduce extra cost in team collaboration. Further, the accuracy of self-reported results may not be reliable. Other potential attention indicators (e.g., visual tracking and keystroke logging) may automate attention status tracking but can only work in tasks that are executed in information systems. We believe a combination of attention tracking techniques can help increase the accuracy and reduce the cost. The second limitation is that we only considered the situation using one WFMS while companies may use multiple WFMSs to manage different projects and tasks. The third limitation is that development of the prototype is not completed yet and field evaluation is needed to further evaluate this approach. In our future research, we would like to extend our work in two directions. First, we will try to integrate multiple WFMSs with attention tracking, implement the prototype system, and conduct field studies to further evaluate our proposed approach. Second, we will explore the usage of visual tracking and keystroke logging technologies for attention tracking.

References

Aalst, W. M. P. van der. (1998). "The Application of Petri Nets to Workflow Management," *Journal of Circuits Systems and Computers* (8:1), pp. 21–66.

Aalst, W. M. P. van der, and Kumar, A. (2001). "A Reference Model for Team-Enabled Workflow Management Systems," *Data & Knowledge Engineering* (38:3), pp. 335–363.

Altmann, E. M., and Gray, W. D. (2008). "An Integrated Model of Cognitive Control in Task Switching," *Psychological Review* (115: 3), pp. 602–39.

Altmann, E. M., and Trafton, J. G. (2007). "Timecourse of Recovery from Task Interruption: Data and a Model," *Psychonomic Bulletin & Review* (14:6), pp. 1079–84.

Basu, A., and Kumar, A. (2002). "Research Commentary: Workflow Management Issues in e-Business," *Information Systems Research* (13:1), pp. 1–14.

Bui, T., and Sankaran, S. (2001). "Design Considerations for A Virtual Information Center for Humanitarian Assistance/Disaster Relief Using Workflow Modeling," *Decision support systems* (31:2), pp. 165–179.

Cabot, J., & Gogolla, M. (2012). "Object constraint language (OCL): A definitive guide,". In M. Bernardo, V. Cortellessa, & A. Pierantonio (Eds.), *Formal methods for model-driven engineering, lecture notes in computer science* (pp. 58–90). Berlin: Springer.

Chiu, D., Li, Q., and Karlapalem, K. (1999). "A Meta Modeling Approach To Workflow Management Systems Supporting Exception Handling," *Information Systems* (24:2), pp. 159–184.

Davenport, T. H., and Beck, J. C. (2000). "Getting the Attention You Need," *Havard Business Review* (78:5), pp. 118–126.

Davenport, T. H., & Beck, J. C. (2001). *The attention economy: Understanding the New currency of business* (p. 276). Book, Cambridge, MA: Harvard Business Review Press.

Delias, P., Doulamis, A., Doulamis, N., and Matsatsinis, N. (2011). "Optimizing Resource Conflicts in Workflow Management Systems," *IEEE Transactions on Knowledge and Data Engineering* (23:3), pp. 417–432.

Fan, S., Zhao, J. L., Dou, W., and Liu, M. (2012). "A Framework for Transformation from Conceptual to Logical Workflow Models," *Decision Support Systems* (54:1), pp. 781–794.

Feldman, M., and Pentland, B. (2003). "Reconceptualizing Organizational Routines as a Source of Flexibility and Change," *Administrative Science Quarterly* (48:1), pp. 94–118.

Georgakopoulos, D., Hornick, M., and Sheth, A. (1995). "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure," *Distributed and Parallel Databases* (3:2), pp. 119–153.

Hansen, M., and Haas, M. (2001). "Competing for Attention in Knowledge Markets: Electronic Document Dissemination in a Management Consulting Company," *Administrative Science Quarterly* (46:1), pp. 1–28.

Kim, G., and Kim, H. C. (2011). "Designing of Multimodal Feedback for Enhanced Multitasking Performance," in *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI 2011*, New York, pp. 3113–3122.

Majchrzak, A., Ronald, E. R., Malhotra, A., King, N., and Ba, S. (2000). "Technology Adaption: The Case of Computer-Support Inter-Organizational Virtual Team," *MIS Quarterly* (24:4), pp. 569–600.

Nabeth, T. (2008). "User Profiling for Attention Support at School and Work," in *Profiling the European Citizen: Cross-Disciplinary Perspectives*, M. Hildebrandt and S. Gutwirth (eds.), Springer Netherlands, pp. 185–200.

Norman, D. a. (1968). "Toward a Theory of Memory and Attention," *Psychological Review* (75:6), pp. 522–536.

Ocasio, W. (1997). "Towards an Attention-Based View of the Firm," *Strategic Management Journal* (18:Summer Special Issue), pp. 187–206.

Patten, C. J. D., Kircher, A., Ostlund, J., and Nilsson, L. (2004). "Using Mobile Telephones: Cognitive Workload and Attention Resource Allocation," *Accident; Analysis and Prevention* (36:3), pp. 341–50.

Rinderle, S., Reichert, M., and Dadam, P. (2004). "Flexible Support of Team Processes by Adaptive Workflow Systems," *Distributed and Parallel Databases* (16:1), pp. 91–116.

Roda, C., and Thomas, J. (2006). "Attention Aware Systems: Theories, Applications, and Research Agenda," *Computers in Human Behavior* (22:4), pp. 557–587.

Rosemann, M., and Muehlen, M. Zur. (1998). "Evaluation of Workflow Management Systems-a Meta Model Approach," *Australian Journal of Information Systems* (6:1), pp. 103–116.

Russell, N., Van Der Aalst, W. M. P., Ter Hofstede, A. H. M., and Edmond, D. (2005). "Workflow Resource Patterns: Identification, Representation and Tool Support," in *Advanced Information Systems Engineering*, O. Pastor and J. F. e Cunha (eds.), Springer Berlin Heidelberg, pp. 216–232.

Salvucci, D. D., Taatgen, N. A., and Borst, J. P. (2009). "Toward a Unified Theory of the Multitasking Continuum: From Concurrent Performance to Task Switching, Interruption, and Resumption," in *Proceedings of the 2009 annual conference on Human factors in computing systems - CHI 2009*, New York, pp. 1819–1828.

Smith, H. A., and McKeen, J. D. (2011). "Enabling Collaboration with IT, " *Communications of the Association for Information Systems* (28: 6), p. Article 16.

Spink, A., Cole, C., and Waller, M. (2008). "Multitasking Behavior," *Annual Review of Information Science and Technology* (42:1), pp. 93–118.

Vertegaal, R. (2003). "Attentive User Interfaces," *Communications of the ACM* (46:3), pp. 30–33.

Wynn, M. T., Reijers, H. A., Adams, M., Ouyang, C., Hofstede, A. H. M. ter, Aalst, W. M. P. van der, Rosemann, M., and Hoque, Z. (2013). "Cost-Informed Operational Process Support," *Lecture Notes in Computer Science* (8217), pp. 174–181

Zeng, D. D., and Zhao, J. L. (2005). "Effective Role Resolution in Workflow Management," *INFORMS Journal on Computing* (17: 3), pp. 374–387.

Zhao, J. L., Bi, H. H., Chen, H., Zeng, D. D., Lin, C., and Chau, M. (2006). "Process-Driven Collaboration Support for Intra-Agency Crime Analysis," *Decision Support Systems* (41:3), pp. 616–633.

Shaokun Fan is an assistant professor in Computer Information Systems at West Texas A&M University. He received his Ph.D. degree in Management Information Systems from the University of Arizona in 2012. He received M.S. degree in Computer Science from the Nanjing University in 2008. His research interests include big data analytics, collaboration management and technologies, and business process management. His work has appeared in Decision Support Systems, Journal of Computer and System Sciences, Concurrency and Computation: Practice and Experience, among others, and in various conference proceedings.

Lele Kang is an assistant professor in School of Information Management, Nanjing University. He holds MS and BS degrees from the School of Business, Nanjing University. His research interests include electronic commerce, open innovation, and collaboration technologies. His work has appeared in Journal of Organizational Computing and Electronic Commerce, International Conference on Human-Computer Interaction, Australasian Conference in Information Systems, and Pacific Asia Conference on Information Systems. He is a member of the Association for Information Systems.

J. Leon Zhao is Head and Chair Professor in IS, City University of Hong Kong. He was Interim Head and Eller Professor in MIS, University of Arizona. He holds Ph.D. from Haas School of Business, UC Berkeley. His research is on information technology and management, with a particular focus on collaboration and workflow technologies and business information services. He is director of Lab on Enterprise Process Innovation and Computing funded by NSF, RGC, SAP, and IBM among other sponsors. He received IBM Faculty Award in 2005 and was awarded Chang Jiang Scholar Chair Professorship at Tsinghua University in 2009.