Expert Systems With Applications 86 (2017) 367–384

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# The specification and design of secure context-aware workflows

Hussein Zedan[a], Saif Al-Sultan[b,*]

[a] Applied Science University, P.O. Box 5055, Manama, Bahrain
[b] University of North Texas at Dallas, 7300 University Hills Blvd, Dallas, TX 75241, United States

ARTICLE INFO

*Article history:*
Received 4 February 2017
Revised 14 May 2017
Accepted 29 May 2017
Available online 31 May 2017

*Keywords:*
Workflows
Context-aware
Security
Specification

ABSTRACT

There are two major concerns in the development of current workflow system. The first is *security* considerations and the second is *context awareness*.

Modern workflow systems cross the boundaries of organisations, each may have its own security requirements, policies and constraints. Even within one organisation, activities in a workflow system may be executed, in one of its instances, within a platform, but in another instance it may be executed or performed on a different platform with completely different environment. Indeed it may not even be automated. This lends modern workflow systems to be *security* and *context-critical*. This is in addition to the fact that some of its activities must satisfy variety of *hard timing constraints*.

Current specification and design languages for workflow systems are inadequate in dealing with context-aware secure workflows. This paper presents a sound wide-spectrum language, $\mathcal{CS}-\mathcal{F}low$, for the specification and design of context-aware, secure workflow systems. As workflow systems have strong temporal aspects (e.g. activity sequencing, deadline, scheduling constraints, etc.), the proposed $\mathcal{CS}-\mathcal{F}low$ is equipped with a rich set of temporal constructs together with temporal proof rules which can be used to be integrated with temporal expert systems, hence enhancing the capabilities of current workflow software. In addition, $\mathcal{CS}-\mathcal{F}low$ is supported by a proof system which underpins the workflow engine that provides decision service using its state to make some decision and update the workflow state. $\mathcal{CS}-\mathcal{F}low$ therefore provides a sound linkage between expert systems and workflow engines.



## 1. Introduction

A business process is a set of activities that are (appropriately) arranged to achieve a given business goal. These arrangements are commonly structured and many of these processes are traditionally well understood, predictable, repeatable and have real-time constraints. The activities in these processes are normally distinct, and the control between them flows in a well defined manner and decisions involved are simple, clear and made in a deterministic fashion. For example, a workflow scenario for the management of a warehouse organises and controls the movement of goods around the warehouse and where about, and the way goods are being stored as efficiently and as safely as possible. These activities, and many others, are to be achieved through the precise definition, processing and realisation of many complex transactions, including goods' *shipping, receiving, putting away, picking up* and *delivering*. However, with the advent of ubiquitous computing environment, these traditional tasks have the extra requirement that they must decide on the next service according to the user's situation (known as *contextual* information) which is continually and dynamically changing.

Workflow execution can involve a large number of different participants, services and devices which may cross the boundaries of various organisations. This raises important issues that are related to *context-awareness* and *security*. It is important to be able to specify exact rules to prevent unauthorised participants from executing sensitive tasks and also to prevent tasks from accessing unauthorised services. For example, medical scenarios will require that only authorised doctors are permitted to perform certain tasks and that only specific machines are used in those tasks. If a workflow execution cannot guarantee these requirements, then the flow will be rejected.

Furthermore, workflows can hold and manipulate various data with different security requirements and it is important to *enforce* these requirements while the data is accessed in a workflow instance. Delegations, constraints over authorisations, audit and integrity provide additional security features. We adopt a *policy-based* approach in which rules are specified compo-

* Corresponding author.
*E-mail addresses:* Hussein.zedan@asu.edu.bh (H. Zedan), saif.alsultan@untdallas.edu, saifalsultan@yahoo.com (S. Al-Sultan).

http://dx.doi.org/10.1016/j.eswa.2017.05.078
0957-4174/

sitionally as policies which may be analysed and verified at run-time.

Currently, there is no commonly accepted model for secure workflows or even a consensus on which features a workflow security model should support. For example, Sushil Jajodia (2001), Abadi, Burrows, Lampson, and Plotkin (1993) give typical policy-based models in which policies are "static". By static we mean that policies have *no* dependency on time or the sudden occurrence of events. Such dependencies are important as they permit policies to change at run time. It is often desirable, and sometimes crucial in many workflows, that after the elapse of a period of time or the occurrence of a particular event, new policies will be adopted and enforced rendering the old ones obsolete. Workflow systems have strong temporal aspects, activity sequencing, deadlines, routing conditions and complex scheduling constraints, all involve the element of time. In addition, temporal/timing aspects of access control requirements are especially important in domains ranging from E-business to even military domain where the value of tactical information are highly dependent on *time*, for example, time to start a mission and its duration) and *events* (e.g. civilian accidents, or change in troops formation).

Furthermore, current temporal expert systems, which use knowledge-based constructs to represent and reason about time, can be used to enhance the capabilities of workflow software. The proposed $\mathcal{CS}-\mathcal{F}low$ wide-spectrum language has recognised the importance of the temporal dimension of workflow and provides a rich set of temporal/timing constructs together with their sound proof rules to enhance the functionalities and capabilities of workflow systems. The work reported in Barker and Stuckey (2003), Bertino, Bonatti, and Ferrari (2001) has recognised the need for temporal / timing-dependent policies. However, these models and others lack compositionality and efficient mechanisms for enforcing them at run-time.

In this paper we take the view that workflow systems are inherently *context-aware* and the fact that it is highly distributed and their activities cross the boundaries of many organisations lend themselves to be *security*- and *context-critical*. This is in addition to the fact that some of its activities must satisfy variety of *hard timing constraints*.

The paper presents a sound wide-spectrum language, $\mathcal{CS}-\mathcal{F}low$, for the specification and design of context-aware, secure workflow systems. The paper is organised as follows. A critical review of the literature is given in Section 2. An informal description of $\mathcal{CS}-\mathcal{F}low$ in terms of textual and graphical representations are given together with some illustrative examples are given in Section 3 ($\mathcal{CS}-\mathcal{F}low$ 's formal semantics is omitted due to lack of space). The specification and design of a health case system in $\mathcal{CS}-\mathcal{F}low$ is detailed in Section 4, and the conclusion is given in Section 5, which contains a full exposition of the algebraic characterisations and temporal and timing proof rules. These are indeed the formal basis for enhancing the functionalities of workflow systems by adding decision support (hence linking between workflow and expert systems technologies).

## 2. Security and context requirements

### 2.1. Security requirements

As we mentioned earlier that there is a no commonly accepted secure workflow model. We take the view that a workflow security model should support some basic requirements. These include:

- Activities are only executed by authorised users and that authorised activities are to access particular services. Provision for conflict resolution should be provided.
- Context-awareness is important as context and the surrounding environment influence the security decisions and may force mobility and adaptation.
- Due to the highly distributed nature of current workflows, a provision for the secure distributed workflow execution should be given and that the ability to specify constraints over workflow migrations and the distribution of activities.
- Adaptability is important and a security framework should allow for the modification of security policies through, for example, dynamic policy changes.

Although it is beyond the scope of this paper, the following addition features are required for any workflow security model:

- Ability to specify privacy constrain over data which are to be manipulated by activities.
- Ensure trusted platforms over which users (human and/or activity) can be authenticated.

In a policy-based approach, a security policy typically reflects the security requirements for the workflow. A policy is a set of rules and procedures controlling the use of information, from processing, storage, to their distribution and presentation (Hung, 2002). Security requirements can describe the types and levels of protection needed for equipment, data, information, applications, and facilities to meet a security policy (Hung, 2002). For example, a policy can be as simple as *a nurse at a grade X is authorised to measure blood pressure*. Here we review some of the existing models that have adopted and proposed within the discipline of *Computer Security* in general.

#### *Existing models*

*Access Control.* Role-Based Access Control (RBAC) models (Ravi et al., 1996) lie at the heart of any authentication process which allocate users into roles. Some work on security in BPEL (Business Process Execution Language) processes, e.g. Bhatti, Joshi, Bertino, and Ghafoor (2003) and Koshutanski and Massacci (2003), rely on authenticating a user through *credential* checking, but they do not specify what credentials or how they are checked.

WS-Security (Novak, Rollo, Hodík, & Vlček, 2003; Nadalin, Kaler, Hallam-Baker, & Monzillo, 2004; Moses, 2005) and SAML (Hughes & Maler, 2005) are two popular web service standards. These can be used to provide secure authentications between different services. A security token, for example in the case of WS-Security, is appended (such as certificates) to SOAP Messages. This can be viewed as a protocol to securely exchange messages between web services by providing confidentiality and integrity of SOAP messages. The SAML standard attempts to solve the Single-Sign-On problem, where a user is authenticated once by a service which is in turn gives an assertion (or a capability) that other services use to authenticate the user – the user does not need to supply his/her credentials again.

Discretionary Access Control (DAC) (Pernul, 1992), is another access control mechanism which has been used widely in file systems. DAC was designed to model security of objects (e.g. *files* on the basis of a single subject's defined privileges (i.e. users). Within this framework, predicates are used to describe the types of access that a subject has over an object. These predicates which have to be evaluated to "true" for granting access or "false" otherwise. Within a file system, for example, privileges can be *read, write, delete, create* and *copy*[1].

[1] We note here that the support required in workflow security relates to the activity's granularity. This has motivated DAC's extension to cater for the security properties of Integrity and Authorization for workflow.