Context-aware Workflow Management Engine for Networked Devices

Y.C. Ngeow[1], A. K. Mustapha[2], E. Goh[1], and H. K. Low[1] and [3]D. Chieng

[1,2]*Faculty of Engineering, Multimedia University, Malaysia*
Asian Research Centre, BT GCTO, British Telecommunications Plc, Malaysia
[1]*{ngeow.yen.churn05, erin.goh.yii.yin05, low.hsien.kung05}@mmu.edu.my,*
[2]*azhar@alum.mit.edu,* [3]*david.chieng@bt.com*

Abstract

In this paper, we present a novel Context-aware Workflow Management Engine (CWME) which offers seamless control and coordination of ubiquitous networked devices in a heterogeneous, autonomous, and distributed (HAD) environment. CWME adopts the concept from contemporary Business Process Management (BPM) systems where end users are allowed to design or customize activities and workflows involving network devices. A working CWME prototype has been developed as a proof of concept and tested using a device simulator. During the workflow design phase, a Feature Interaction Detection Engine (FIDE) is adopted in our system to detect all possible operation conflicts between the devices. During workflow execution phase, we propose Justified-Event-Condition -Action (JECA) rule and a Case-Based Reasoning (CBR) to enhance CWME's exception handling capability.

1. Introduction

Workflow, as defined by Workflow Management Coalition (WfMC) is an automation of procedures where documents, information or tasks are passed between participant according to a defined set of rules to achieve, or contribute to, an overall business goal [1]. Most Workflow Management Systems (WFMSs) has been widely developed as Business Process Management (BPM) applications to improve the efficiency and effectiveness of any organization by bringing business processes, people and information together. Some of the prominent BPM solutions such as Fujitsu’s I-flow, IBM’s FlowMark as well as SAP NetWeaver have been well implemented in various enterprises to allow the integration of and alignment of people, information and business processes across business and technology boundaries.

Workflow modeling supports a variety of design and management approaches for enterprises. The most commonly known are BPM, risk management, and enterprise architecture design. Managing hardware devices through workflow is still considered as a new area of research. More research could be done to extend the capabilities of workflow in this area. We propose a novel Context-aware Workflow Management Engine (CWME) that supports seamless control and coordination of hardware devices such as air-conditioner, printer, projector, etc., in heterogeneous, autonomous and distributed (HAD) environment. We intend to show how workflow model, with its richer semantics is able to be applied in a commercial products to manage devices such as configuring device and setting up activity execution preferences, for generic consumers.

Most existing workflow systems lack of the capability to manage exception events. A good exception handling mechanism is important to enhance the whole workflow's efficiency and effectiveness. In this paper, we have categorized those exceptions that might occur in CWME and designed the exception handling mechanism in the context of hardware device management.

The rest of the paper is organized as follows. Motivation and related work are discussed in Section 2. Section 3 describes the context-aware workflow model including the CWME's architecture. Section 4 describes our case study and simulator results. Section 5 identifies the possible exceptions whereby exception handling mechanism is described in Section 6. Lastly Section 7 discusses this system and its future work.

2. Motivation and related work

The workflow model presented by WfMC supports service automation in processing tasks using a sequence of rules. It has been successively applied to traditional computing environments such as business processes and distributed computing in order to perform service composition, flow management, parallel execution, and time-driven services [2]. However, a context-aware workflow engine has to be implemented in order to support automatic service on dynamically changing environment. Furthermore, users of ubiquitous computing environments may want to receive services in an appropriate form, at the appropriate time, and without user intervention [3].

The workflow reference model introduced by the WfMC does not consider the initial creation of process definitions to be an area of standardization as it is considered to be a major distinguishing area between products in the marketplace [1]. Therefore, workflow engine in the market has various process definitions which vary according to its application purposes. Workflow engine such as OpenWFE [4] is a business process management suite. It interprets and runs Business Process Definitions that describe the transition of work items between participants. Eventually it does not require context information in their process definitions. Another example of similar workflow engine that does not depend on context information for their system would be Business Integration Enterprise (BIE) [5] and jBPM [6].

Gaia [7] implemented a service environment by making its ubiquitous applications to communicate with each other using context information. However, it is built on CORBA middleware and it depends on a specific protocol that is not widely used. Moreover, LuaOrb, the script language which Gaia used cannot express dependency or parallelism among the services because it describes only a sequential flow of specific services.

Many of the workflow systems do not provide satisfactory solutions to the problems of exception handling [8]. Recovery mechanism introduced by C. Hagen [9] depended on the sphere concept that integrates transaction concepts directly into his system. Conversely, G. Alonso [8] tried to show that transactional properties are not adequate to cope with failures since they do not address many of the issues that have made workflow systems so popular. An exception handling mechanism which uses Event-Condition-Action (ECA) rule is commonly used and adopted in several workflow prototypes as a modeling tool. However, Zongwei [10] introduced Justified ECA (JECA) rule that eliminates the limitation of ECA rule and supports context dependent reasoning processes in dealing with uncertainties. Additionally, he introduced Case-Based Reasoning (CBR) as he believed that designing an integrated human-computer process may provide better performance than moving toward an entirely automated process in exception handling. Knowledge-based approach is another similar method to CBR which is implemented by M. Klein [11].

As the workflow and devices involve user's interaction, a major concern on handling exception is that a workflow designer cannot predict every single possible case that may occur when a process is being executed. In order to reduce exceptions occurrence and to enhance the system's exception handling capabilities, we developed a workflow engine which supports Feature Interaction Detection Engine (FIDE) and Context Engine (CE). FIDE plays an important role in detecting possible conflicting task execution that could exist during the workflow designing phase whereby CE updates CWME and FIDE with the latest context information of the devices such as device availability, device activity services, device location and etc.

Lastly, JECA rule and CBR are proposed to be implemented as the exception handling mechanism for our system. We believe JECA rule model and CBR could provide a wider range of exception handling options.

3. Context aware workflow model

3.1. Adoption of BPM workflow

BPM workflow represents the next generation in the evolution of Information System. Although its main purpose is used to help organizations formalize specifications of business processes, the capabilities of BPM which enables analysis, monitoring and execution of underlying processes suit our research purposes. BPM will be an emerging trend in the area of business process automation and system design. It is slowly creating an impact that was once created during the advent of Database Management Systems and new industry is expected to grow around process management. According to an April 2001 report by Gartner Research, enterprises which continue to hard-code all flow control, or insist on manual process steps and do not incorporate BPM's benefits, will lose out to competitors that adopt BPM.

3.2. The need of context-aware workflow

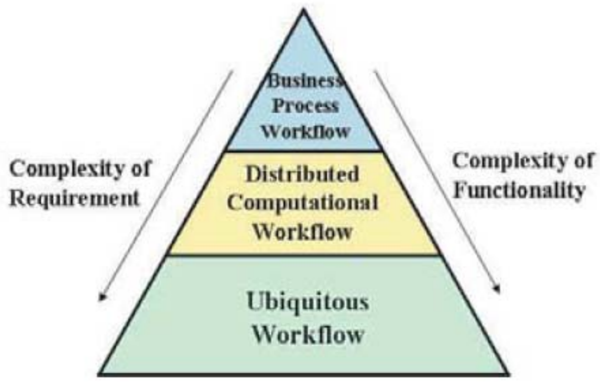

Figure 1. Evolution of workflow

The early workflow was initially applied to the automation of business processes, whereby the business process is static, and the requirements and functionalities are simple. Figure 1, which is taken from [2], shows the evolution of workflow from top to bottom. From business processes, the workflow has been evolved into the field of distributed computational workflow such as Grid services. In future, more computing work will be carried out. Thus, the complexity of requirement as well as the functionality will increase. The workflow must be evolved and it will be evolving to support service automation in ubiquitous computing. Workflow will be the best tools to resolve automation problems in dynamically changing environments.

In order for the workflow to operate in ubiquitous environment and to provide context-aware services, it relies on the context information from its environment. Context is referred as any information that can be used to characterize the situation of an entity such as person, place, event, object, etc. Taking an office environment for example, context can mean :

- Identity: a person or an object such as telephone, computer, fax machine, projector, photocopy machine, coffee maker, etc.
- Location: specific places such as conference room, pantry, etc.
- Service: a function offered by an identity or a collection of identities such as lighting control, temperature control, printing of documents from a remote location, voice communication, etc.
- Event: referring to an event that occurs on a specific time of the day such as monthly meeting, sales or product presentation, technical training, etc.

3.3. Context-aware workflow engine

The context information is collected and generated by the CE. We apply the latest developments of Semantic Web [12] which offers a richer description of web resources. By using standards to represent machine-interpretable information such as Resource Description Framework (RDF), computing entities can easily exchange and infer context based on explicit context representation. RDF is in fact ideal for representing the entities in order to add semantic richness as well as offers unique resource naming and expandability. The CE applies declarative language based on RDF for querying contexts and a rule-based reasoner for inference purposes. By hosting CE's functions via web services, the workflow engine is able to connect to the CE's functionality and consequently creating an efficient context-aware workflow engine.

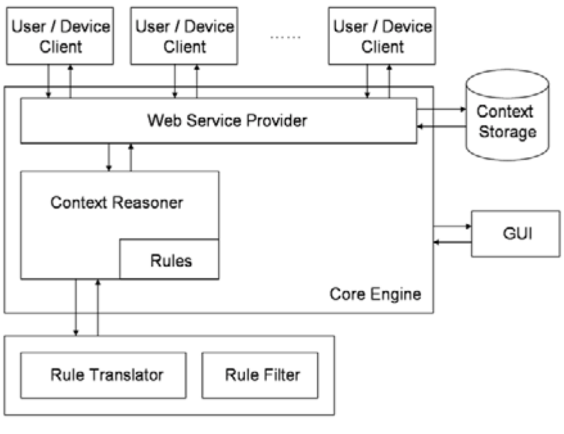

Figure 2. CE's context architecture

As shown in Figure 2, the CE architecture is built using Jena Semantic Web Toolkit [13], a Java framework which provides the APIs to create and manipulate RDF. It is consisted of the following collaborative components:

- Core engine: is consisted of a web service provider and a reasoner. The web service provider allows the CE's services to be utilized through web services whereby the reasoner provides logic reasoning services to process context information.
- Context storage: stores the context information of every single entity in the system.
- Rules repository: stores the context information of every single entity in the system.
- Graphical User Interface (GUI): allows system administrator to manually register physical entities such as user or device to the system.

3.4. Context-aware workflow design process

Figure 3 shows the context-aware workflow design process. Firstly, user will have to login into the CWME through any web browser. Then, the user accesses the CWME through its GUI and designs his workflow activities in the workflow workspace. Each of the devices inside the workflow can be further specified through the dynamic activity form templates. As an example from the meeting scenario, the user could select the available printers, select the time of execution and define the printing task to be executed.

The form template of the device changes dynamically according to the context information obtained from CE, which provides the form template with the most current and updated context information of devices such as the device status, services availability, device location and etc. via the Context WebService Client. Thus, the dynamic change of the form template has extended the flexibility and extensibility of the CWME to manage any device's services in a dynamic and heterogeneous environment.

The completed workflow design will be sent for error checking with the help of Form Validator and FIDE. Further explanations about the functionality of these two components are described in Section 5. Once the FIDE and the Form Validator verified that the workflow design is error-free, the Core Engine will schedule those selected activities and execute them accordingly.

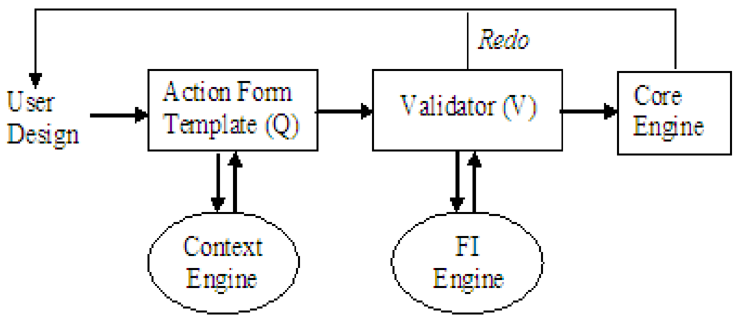

Figure 3. Context-aware workflow design process

3.5. CWME's system architecture

The CWME is build by extending the functionality of an open source workflow application, Business Integration Engine which incorporated JBOSS/J2EE application server as well as Tomcat Web Server. Our working CWME's system architecture and its supporting engines as shown in Figure 4, contains the following components:

- Core Engine: contains the workflow GUI including the workflow workspace and controls the workflow enactment process.
- Activity Form Template: contains all the device form templates which dynamically change according to the information retrieved from the CE.
- Form Validator: validates the completed workflow design.
- Service Providers: provides device services to the Core Engine.
- Scheduler: schedules the execution which has been defined by the user in the workflow design.
- Device WebService Client: provides web services obtained from Device Simulator (act as an intermediate communication between Scheduler and Device Simulator by using SOAP).
- Context WebService Client: provides web services obtained from CE (act as an intermediate communication between Activity Form Template and CE by using SOAP).
- FI WebService Client: provides web services obtained from FIDE (act as an intermediate communication between Form Validator and FIDE by using SOAP).

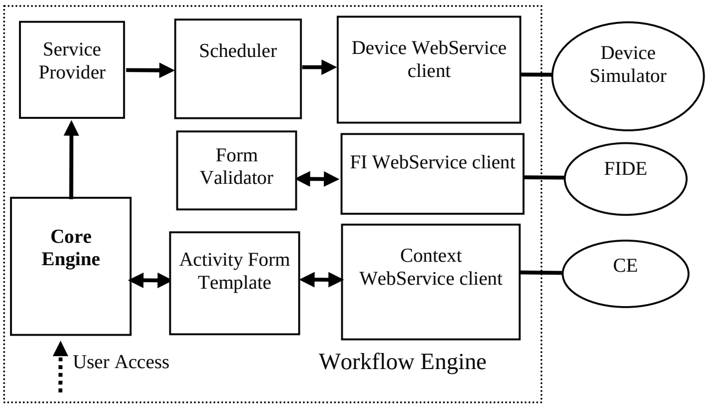

Figure 4. CWME's system architecture

4. Case study and simulator results

To demonstrate the CWME's capability on managing web-based devices, a device simulator was created to capture the execution supposedly to be carried out in the devices. The following paragraphs will describe a typical meeting scenario in office atmosphere and the graphical user interface of the workflow engine.

4.1. Scenario

Adam is the chairman and would like set up a meeting with his subordinates. The activities can be configured prior to the meeting where invitation can be sent out to all the participants; meeting notes can be printed out; air-conditioner will be functioning right before the meeting started; lighting in the meeting will be controlled according to user preferences; projector will be automatically switched on with the uploaded slides during the meeting.

4.2. Context-aware workflow engine

Firstly, Adam logins into the CWME through any web browser. Then, he could access the workflow workspace in order to design a workflow of activities. Each of the activity represents a device and he could design his workflow, which is to set up a meeting by dragging and dropping the activity into the work space as shown in Figure 5. It shows a workflow of activities that sets up an office meeting scenario. In order to configure each of the activity according to Adam's preferences, he could access the device's dynamic activity form template through the activity icon. Figure 6 shows the projector's activity form template page with the dynamic context information from CE. By accessing the printer's dynamic activity form template, Adam is able to choose the current services which are provided by the available printers in his office. The context information shown on the template will assist Adam in picking the right printer. Furthermore, even if Adam is still in the process of configuring the printer, any new printers detected by the CE will dynamically change the printer's form template.

The completed configuration will be checked by the Form Validator in order to make sure Adam has completed the configuration successfully. Then, it will be sent to the FIDE to verify that Adam's workflow will not have any conflicting execution with other devices' execution. Once the error-free workflow design is verified by the FIDE, the activities will be scheduled to be executed accordingly on the meeting day.

The Device Simulator is created to simulate our working scenario. It contains the available devices at a specific location and deploys Web service technology to the web-based device. Any execution of the activities from CWME will be captured by Device Simulator via Web service technology and current activity of the device will be updated. The Device Simulator is designed based on Web service technology as it is a standardized communication method and it is independent of heterogeneous platforms, protocols, and languages. Web service will serve as the middleware that enables remote procedure calls (RPCs) for a device execution to be carried out over the Internet.

The captured execution commands as shown in Figure 7 demonstrated the working and executable CWME.

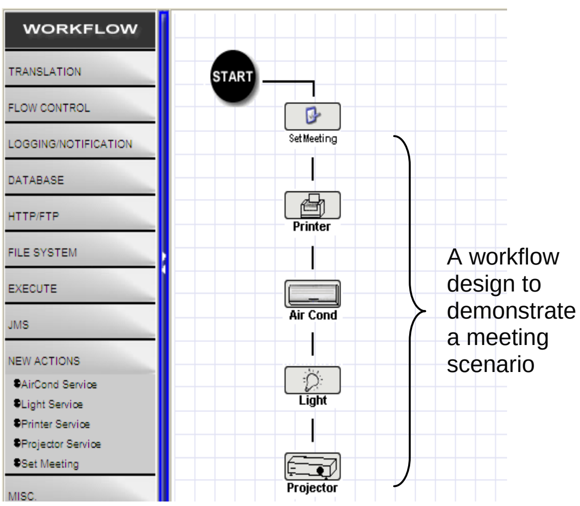

Figure 5. Graphical user interface of the workflow workspace

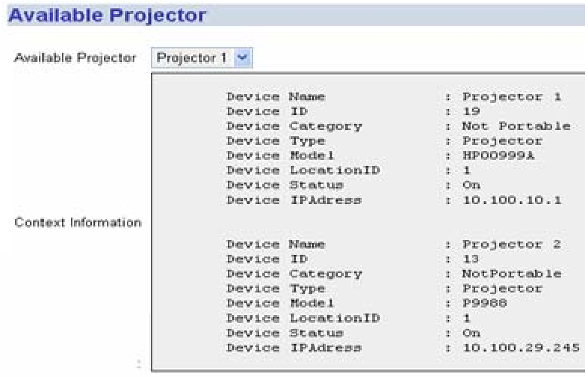

Figure 6. Projector's dynamic activity form template page with dynamic context information

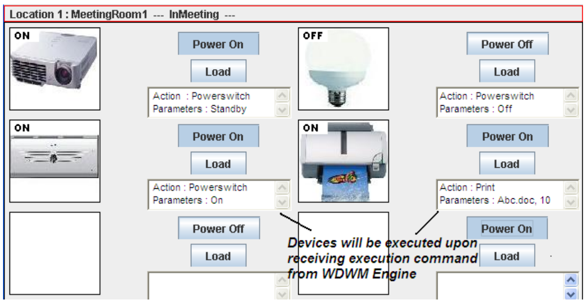

Figure 7. Graphical user interface of the device simulator

5. Exceptions

A deviation from a normal process in workflow may occur at anytime and it may cause a breakdown that could stop all the processes from being executed orderly. We identify some of the possible exceptions which could occur in our CWME. They can be classified into a few different categories: program/system failures, design and semantic failures, feature interactions failures, device failures. These exceptions would be explained as follows:

- Program/System failures :
  A program/system failure could cause unsuccessful abort of an external application. Several components which supported the workflow engine may be unable to communicate or perform their function due to internal system error in the components. Communication error occurs when workflow engine fails to receive response from device or other engine. One of the methods to resolve this failure is by retrying execution at a later point in time.
- Design and Semantic failures :
  Design and semantic failures could exist during designing of the workflow. Occurrence of these failures due to the mistakes done by user during the process of designing workflow may happen when the required data input from the designer is invalid or insufficient. For example, the available device is not selected by the designer and causes the task configuration to be incomplete. Such failure could be resolved by checking workflow design for errors each time a workflow design and its task configuration are completed. Configuration changes may also lead to

design and semantic failures such as incorrect parameters for program invocation or impossible program execution. A Form Validator could be deployed to identify design and semantic failures and notify the designer to make appropriate changes.

- Feature Interactions failures :
  Although the workflow process can be executed successfully without any design and semantic errors, a feature interaction failure could be occurred due to unforeseen constraints when devices are executed concurrently. A useful scenario to demonstrate this failure is when an air-conditioner and a heater are being switched on at the same time. The workflow design would not return any errors as these two devices are able to execute individually but it would serve no purpose when they are executed concurrently. The completed workflow design will be consulted with the FIDE in order to check for any possible feature interaction failures before the designed workflow is going to be executed. FIDE will notify the designer if an exception might occur. However, the threats from feature interaction failures are subjective. Some of these failures can be ignored depending on user preferences.
- Device failures :
  When a task to print a document is executing, malfunctioning of the printer could cause device failure to occur. The user of that particular task will be notified and is requested to select another available printer or schedule another time. Therefore, the CE plays an important role by keeping track of the device status and providing the CWME with the accurate and up to date information on devices. Any problematic device will be removed from the list of available services in CWME.

6. Exception handling mechanism

Exception handling mechanism is an important part of the whole prototype system to ensure a reliable and consistent execution of the processes in the presence of failures and exceptions. It is used to prevent and to solve exceptions that would occur in CWME. To analyze how exception handling mechanism can be applied in our system, we divided the workflow process into two phases, which are workflow design phase and execution phase. Workflow design phase is defined as the stage when user is designing and configuring his workflow of devices' activities and the execution phase is when the activities are being executed by the devices. During these two phases, an exception which is defined in previous Section 5 could happen.

6.1. Workflow design phase

Failures during the workflow design phase usually can be easily detected and solved by notifying the user to amend the causes of the failures. We developed Form Validator and FIDE to prevent any error created during workflow design phase. Design and semantic failures can be captured by the Form Validator whereas the feature interaction failures are detected by the FIDE. The Form Validator uses a generic comparison functions provided by Java programming language and checks for any invalid or insufficient data input from the designer.

Additionally, the FIDE is more complex and contains a few components in its engine. FIDE contains a main core engine that coordinates the whole conflict detection function. The main core engine collects the request from CWME, checks for the possible feature interaction failures and returns back the results to the CWME. Based on the rule repository, which is a database containing all the conflict detection rules, FIDE will use a checking algorithm to check for any possible conflicting activities which has been configured or designed by the user. The checking is written using logic programming language, Prolog. A more detailed explanation of this FIDE can be found in [14]. Figure 8 shows the process of exception handling mechanism during workflow design phase that was developed in our CWME. The figure also shows how the reliable and up to date context information of the devices such as the device status is provided by the CE.

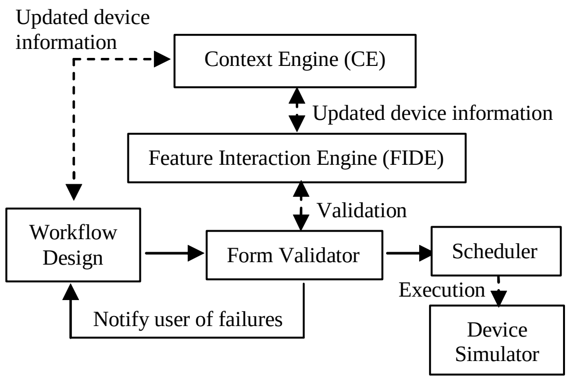

Figure 8. Workflow exception handling process during workflow design phase

6.2. Workflow design phase

Successful execution of the activities is important as it determines the completion of the execution phase. We propose to implement JECA rule as well as CBR as our approach to handle exceptions which occur during workflow execution phase. Figure 9 shows the usage of JECA and CBR as our proposed approach to handle exceptions.

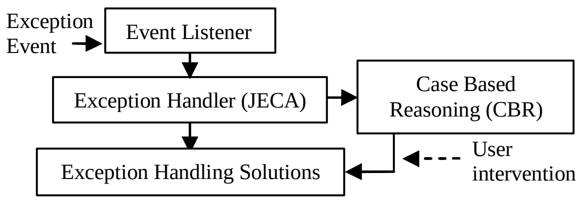

Figure 9. JECA and CBR exception handling mechanisms during workflow execution phase

6.2.1 JECA rules : Event-Condition-Action (ECA) is commonly used and adopted in several workflow prototypes as a modeling tool. The limitation of capturing rules evaluation context in ECA rules leads to the usage of JECA rules where justification (J) provides a reasoning context for evaluations of ECA rules in order to support context dependent reasoning processes in dealing with uncertainties, especially in our case, the device failures. In other words, JECA is able to capture more workflow contexts and is able to support context-dependent reasoning processes especially in the dynamic and uncertain environment of the device behavior. Each JECA rule, R contains four components as follow :

- Justification (J) :

Justification forms the reasoning context in which evaluation of the specific JECA rule to be performed. It is frequently used as a disqualifier. For example, the rule is disqualified if J is evaluated to true.

- Event (E) :
  If event occurs, related JECA rules will be evaluated.
- Condition (C) :
  Condition is used as a logic constraint to be satisfied so that the action (A) in the rule can be executed if the rule is not disqualified.
- Action (A) :
  If this rule, R is not disqualified, where an event occurs with condition (C) is met, and justification (J) is not satisfied, then actions (A) will be carried out to resolve the exception event. The Action, which is the exception resolution, is taken according the specifications that are configured earlier by the workflow designer. There is no requirement to implement forward recovery as each of the device activities, in nature, are not atomic and can be individually executed. We proposed the following exceptions resolutions in our system:
  Ignore - No further action and carry on the next activity.
  Abort and Compensate – Complete abort the parent process and remove all work depended on the aborted task.
  Retry – Re-execution the task for a certain number of attempts.
  Notify – Notify user of that particular task

The below Figure 10 shows JECA rule execution algorithm. The exception event will be propagated to the CBR for evaluation if the JECA rule, R is disqualified. It will be explained further in the next paragraph.

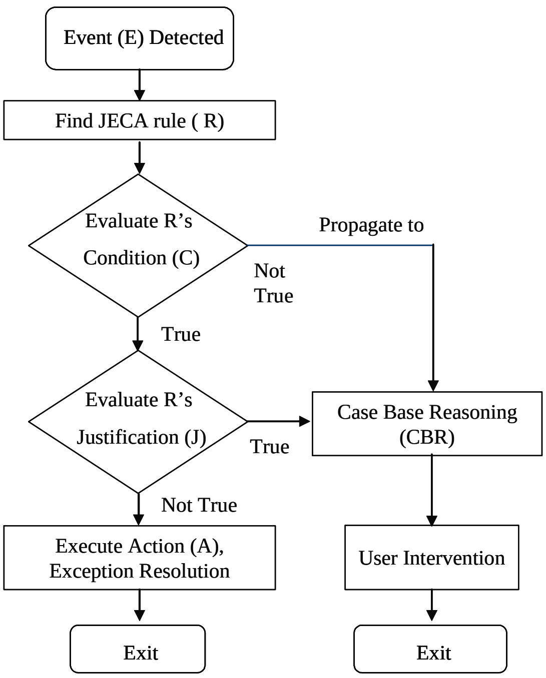

Figure 10. JECA execution algorithm

Consider the following example of JECA rule, R1 in dealing with an exception related to printing device. In this rule, the exception mechanism is related to an exception returned by the printer. The initial Action (A) will be carried out if the justification does not disqualify the JECA rule.

Rule R1 :
Event : PrintingException
Condition : Failure of printing
Action : Retry printing execution <5
Justification : Paper tray failure

The justification provides a reasoning context that if the failure of the printing device is related to paper tray failure, then it requires human attention to resolve the problem. If not, the printing task will be executed again for a maximum tries of 5 times. When paper tray failure occurs, rule R1 will not be evaluated as it will be disqualified base on the justification condition. Then, the exception will be sent to the CBR component to resolve the exception.

6.2.2 CBR : There could be a variety of unique device failures because there is a wide range of possible devices with different individual functionality and capability. As a result, the exception handling mechanism using JECA rules is impossible to resolve all the possible exceptions that could occur in our CWME. We propose to include a CBR to provide an alternative solution when local exception handler unable to solve the exceptions and user intervention is needed to handle them. The CBR approach is based upon a generic and reusable body of knowledge concerning what kinds of exceptions can occur in collaborative work processes and how these exceptions can be handled. CBR involves in collecting cases each time an exception event is solved successfully, retrieving similar prior exception handling cases and reusing the knowledge captured in those cases in new situations. There could be more than one exception handler or resolution from CBR for handling a particular exception. In order to help the workflow designer to select among them, all handler processes are annotated with their characteristic preconditions and performance properties.

Taking example of the paper tray problem, the CBR will assist user to select appropriate solution. The example below shows the exception and its possible solutions.

Exception : PrintingException
Problem : Paper tray failure
Printer : HP 1015
DeviceID : HP2525
Location : Panasonic Lab

Problem 1 : Insufficient paper
Resolution : Fill in the paper tray
Relevancy : 90%
Occurrence : 9/10

Problem 2 : Paper tray jammed
Resolution : Remove trapped paper from tray A
Relevancy : 10%

Occurrence : 1/10

Problem 3 : Paper tray jammed
Resolution : Remove trapped paper from tray B
Relevancy : 0%
Occurrence : 0/10

7. Discussion and future work

In this paper, we showed how context-aware workflow model, with its richer semantics is able to be used in a commercial products to manage generic devices for consumers especially in office environment. The scalability of this CWME enables the possible usage in other scenarios such as home environment. For example, home user is able to design workflow that manages home security by switching on the alarm systems, and recording the network camera or closed-circuit television according to user specified time.

Different types of exceptions which could occur in our CWME are being identified in this paper. To enhance our system's exception handling capability, FIDE and CE is used to detect possible exceptions occurrence by checking the workflow design during the workflow design phase. During the workflow designing phase, the FIDE detects possible conflicting activity execution that could exist whereas CE helps in eliminating unwanted device failures by updating the latest device status.

However, our current prototype is not able to resolve exceptions during the workflow execution and we propose the usage of JECA rule and CBR as the exception handling mechanism of the workflow in our future work. The current relevancy in CBR depended on the occurrence and the exception which are detected. The repository is still in the design stage and we hope to expand the repository and the relevancy algorithm in future work. Besides that, exception between concurrent workflow executions is not being analyzed in this system and should be carried out in future.

The device task executions are tested using a device simulator with graphical user interface that captures execution command from the CWME and it proofs the feasibility of our concept of managing devices through an adaptive CWME. Although the test was done using a simulator, CWME would be able to work for any hardware or devices connected in the network as a research done by Wong [15] demonstrated that devices can be operated through web services. Future work will be carried out to expand more devices scenarios and to identify more device failures.

References

[1] Workflow Management Coalition (WfMC) site, http://www.wfmc.org.

[2] J. Han, Y. Cho and J. Choi, "Context-Aware Workflow Language Based on Web Services for Ubiquitous Computing", Intl. Conference on Computational Science and Its Applications (ICCSA), Singapore, 2005.

[3] G. Chen and D.Kotz, "A Survey of Context-Aware Mobile Computing Research", Dartmouth Computer Science Technical Report. TR2000-381. Dartmouth College, 2000.

[4] Open source WorkFlow Engine (OpenWFE) site, www.openwfe.org.

[5] Business Integration Engine (BIE) site, http://www.brunswickwdi.com/dashboard.

[6] JBoss jBPM site, http://www.jboss.com/products/jbpm.

[7] M. Roman, C. Hess, R. Cerqueira, A. Ranganat, R. H. Campbell and Klara Nahrstedt, "Gaia – A Middleware Infastructure to Enable Active Spaces", IEEE Pervasive Computing, pp. 74-83, 2002.

[8] G. Alonso, D. Agrawal, A. El Abbadi, M. Kamath, R. Gunthor and C. Mohan, “Advanced Transaction Models in Workflow Contexts,” in Proc. of Intl. Conference on Data Engineering (ICDE), 1996.

[9] C. Hagen, G. Alonso, “Beyond the black box: Event-based Interprocess Communication in Process Support Systems”, 19th Intl. Conference on Distributed Computing Systems, May 1999.

[10] Z.Luo, A.Sheth, K.Kochut, J.Miller, “Exception Handling in Workflow Systems,” Applied Intelligent, vol. 13, no. 2, pp. 125-147, Sept.-Oct. 2000.

[11] M.Klein, C.Dellarocas, “A Knowledge-based approach to handling exceptions in workflow systems,” Computer Supported Cooperative Work, vol. 9, no. 3-4, August 2000.

[12] W3C Semantic Web site, http://www.w3.org/2001/sw/.

[13] Jena2 Semantic Web Toolkit: http://jena.sourceforge.net.

[14] H.K. Low, D. Chieng, A. K. Mustapha, Y. C. Ngeow, and E. Goh, “Feature Interaction Conflicts Detection in Ubiquitous Networked Appliances”, International Symposium on Information and Communications Technologies (M2USIC) 2005.

[15] S.C. Wong, S.N. Cheong, Azhar, “Hardware Community through the use of Web Services and Hardware Interpreter”, 7th WSEAS CSCC, July 2003.

[16] Y. C. Ngeow, D. Chieng, A. K. Mustapha, H.K. Low, and E. Goh, “Dynamic Devices and Workflow Management for Ubiquitous Multimedia Devices”, International Symposium on Information and Communications Technologies (M2USIC) 2005.

[17] M. Kamath and K. Ramamritham, “Failure Handling and Coordinated Execution of Concurrent Workflows,” in Proc. of Intl. Conference on Data Engineering (ICDE), 1998.

[18] C.Hagen, G.Alonso, “Flexible Exception Handling in Process Support Systems”, International Conference on Distributed Computing Systems, 1998.

[19] M.Brambilla, S.Ceri, S.Comai, C.Tziviskou, “Exception Handling in Workflow-Driven Web Applications,” International World Wide Web Conference, pp. 170-179, May 2005.

[20] C.Hagen, G.Alonso, “Exception handling in workflow management systems”, Software Engineering, IEEE Transactions, vol. 26, no. 10, pp. 943-948, 2000.

Authors

Yen-Churn Ngeow

Received a B. Elecs. Eng. degree in computer engineering from Multimedia University, Malaysia, 2004. In 2004 he joined the Faculty of Engineering, Multimedia University, Malaysia as a research scholar in the area of multimedia network devices. His research interests include workflow engine, exception handling mechanism, networking device application and multimedia communication systems.

David Chieng

Joined BT Asian Research Centre in 2006 as a technical lead in broadband wireless research group. His current research interests include wireless networking and ubiquitous computing. Prior to joining BT, David was involved in a European EPSRC Cordless Access to Multimedia Applications (CAMA) project from 1997-1998. From 1999-2002, he was the principal researcher in BONDS (Brokering Open Network Differentiated Services), a joint research project between Queen’s University, Ulster University and Fujitsu Telecommunications Europe Ltd. From 2003-2006, he was a lecturer, then senior lecturer in

Multimedia University, Malaysia and was leading a number of research projects in the areas of Computer Telephony Integration and ubiquitous computing. David received his MSc and PhD in Communication Networks from Queen's University of Belfast, Northern Ireland, U.K., in 1998 and 2002 respectively. He has authored and co-authored numerous publications in international journals, conferences and book chapters.

Azhar Mustapha

Received a B.Sc. and M. Eng degree in Electrical Engineering and Computer Science from Massachusetts Institute of Technology, USA. He joined Multimedia University of Malaysia where he is currently a Research Specialist and leads a team conducting advanced research & development work funded & sponsored by the Matsushita Corporation of Japan. Azhar has written & published numerous scientific papers at the International level. His area of research involves speech coding, pervasive computing, interactive multimedia, web semantics and enterprise computing.

Erin Goh

Received a first class honour degree in Bachelor of Information Science and Technology, majoring in Software Engineering from Multimedia University, Malaysia, 2004. In 2004, she joined the Faculty of Engineering, Multimedia University, Malaysia as a research scholar in the area of multimedia network devices. Her research interests include ubiquitous computing, internet technologies and telecommunication networks.

Hsien-Kung Low

Received a first class honour degree in Bachelor of Information Science and Technology, majoring in Software Engineering from Multimedia University, Malaysia, 2004. In 2004, he joined the Faculty of Engineering, Multimedia University, Malaysia as a research scholar in the area of multimedia network devices. His research interests are feature interaction and networking devices.