Towards Distributed Collaborative Workflow Management for Mobile Devices

Anna Kocurova, Samia Oussena, Peter Komisarczuk, Tony Clark, Dean Kramer

Towards Distributed Collaborative Workflow Management for Mobile Devices

Anna Kocurova [1], Samia Oussena [1], Peter Komisarczuk [1], Tony Clark [2], and Dean Kramer [1]

[1]School of Computing and Technology
University of West London, UK
[2]School of Engineering and Information Sciences
Middlesex University, London, UK
{Anna.Kocurova,Samia.Oussena,Peter.Komisarczuk,Dean.Kramer}@uwl.ac.uk
T.N.Clark@mdx.ac.uk

Abstract. Using mobile devices enhances overall collaboration and communication between team workers. Co-workers can collaborate and share content regardless of their location. Workflow management is a widely accepted technology that supports collaboration; however, existing workflow approaches provide only a limited support for content management. Moreover, constantly changing collaborators' contexts and needs demand more adaptable and flexible workflows. The aim of this paper is to examine how collaborative workflows can be adapted for mobile devices, be capable of dealing with current collaborators' context and support content manipulation. As a result, a conceptual reference framework for the definition, management and execution of mobile context-aware content-centric workflows is introduced.

Key words: Distributed workflow management system, context-awareness, content lifecycle, mobile collaboration

1 Introduction

The demand for an intelligent electronic environment which responds to people′s needs is growing with worldwide proliferation of portable mobile electronic products. With mobile devices such as smartphones, tablets or PDAs becoming ubiquitous, people are more and more dependent on using mobile computing capabilities in everyday's life. The unique mobile's benefits are that the mobile devices are the always-on and always-carried mass media with ability to create and distribute content, take and upload a picture, capture an important event and publish the video in near real time [1]. People often want to share content or information privately by using mobile devices in various situations such as in business meetings. They expect to have tools to collaborate, share and publish content as required by the situation [2].

Workflow management is a technology that can support collaboration and content sharing between participants in mobile settings. Mobile devices reside in extremely dynamic contexts. Frequent changes of user, device or environment state can influence behaviour of mobile information systems. Mobile workflow capabilities can be enhanced by making workflow systems more context-aware and adaptable to changing conditions. Context awareness might have a number of meanings, depending on the domain to which it is applied. We use the following definition of context [3]:

> *'Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. Context is typically the location, identity, and state of people, groups, and computational and physical objects.'*

In this paper, context is mainly related to two workflow concepts: content-related context and context driven collaboration. The concepts are examined in details below.

Content-related context: Content shared by mobile devices differs from content circulated in traditional systems. Mobile content such as image, document or video/audio file is usually user-generated or adapted for use on mobile devices and can be context-aware. Context-aware content is content semantically enriched by context. For example, a picture is associated with information about the physical location or time when it was taken. Creating, processing and disposition of context-aware content are the basic stages of its lifecycle. However, content can flow through a more complex management process. The process might involve content editing, reviewing or publishing. In addition, context information associated with content can be changed at any time. The context change can trigger an action, cause content movement between two different states or influence content behaviour. In process-centric workflows, there is only a little support for recognition of context-aware content lifecycle. Integration of the lifecycle into workflows and extending mobile workflow management systems by additional context-aware content management functionalities can enhance content manipulation and sharing. In this paper, a context-aware content-centric approach with focus on key collaboration-related context-aware contents, their lifecycles and management is described.

Context driven collaboration: Content sharing between collaborators might be a costly operation in terms of transfer cost, devices resource usage and users time consumption, especially when task can be performed by a number of participants with the same role but not all of them can really accomplish it within a required time. Moreover, collaborators expect to have solutions for content sharing adapted for their current needs, preferences and situation. Collaboration driven by collaborator's current context can improve team work. However, the underlying management topology has a significant impact on the way how context can drive collaboration. In centralised workflow management approaches, management decisions are made by servers and context driven collaboration would be hardly achievable. In peer-to-peer (P2P) workflow management, management decisions are made by mobile devices. The decision can be based on local context-related information. By using the P2P management topology, workflows can become more context oriented processes adapted for collaborators needs.

The P2P management topology requires the development of a mobile distributed management system that consists of a set of mobile devices, connected through wireless links. High-speed wireless connections (Wi-Fi) and open wireless technology standards such as Bluetooth enable connection between devices in various environments. In P2P workflow management, data and content are exchanged directly; administration and responsibility for operations are distributed among the devices [4]. No device has complete information about the workflow state. By using P2P management technology, communication processes can be accelerated and collaboration costs are reduced through the ad hoc administration of working groups [5]. Moreover, each device can consider a variety of local information to determine what will happen next.

In this paper, a context-aware content-centric collaborative workflow tailored for mobile P2P collaboration is introduced and a reference framework for its definition, management and execution is elaborated. This framework can bring benefits to designers and developers of mobile collaborative applications who will be able to define own workflows according to their needs. In particular, the integration of a context-aware content-centric perspective into a workflow process is examined. Content behaviour is expressed by its lifecycle. The content lifecycle is composed of content states and transitions between content states. The transitions are driven by context guard conditions. A logical architecture of a workflow management system which is capable of executing such workflows is outlined.

The structure of the paper is as follows. Section 2 describes related work. The research methodology is highlighted in Section 3. A case study is described in Section 4. Context categorisation and management are presented in Section 5. Section 6 offers the definition of context-aware content-centric workflows. The logical architecture of the workflow management system is discussed in Section 7. The final section 8 summarizes the current research stage and results achieved so far.

2 Related Work

The need for adaptive workflow management demanded by practical situations is widely known and acknowledged. We review related work to workflow adaptation according to three categories: context-aware workflow modelling, context-aware workflow adaptation and artifact-centric workflow approaches.

Wieland et al. pointed out that workflow meta-models should support context modelling and its use in workflows [6]. Earlier works such as Context Toolkit [3] provided rather simple frameworks for context modelling. Later, more structured and complex frameworks that facilitate the development of context-aware applications were presented [7, 8, 9]. The frameworks address concepts of context management such as differentiation of lower-level raw context data from higher-level derived context information, context aggregation and context querying. Recently with increased context data dissemination, shared global context models for context-aware applications in large-scale scenarios such as Nexus approach have been developed [10]. However, we believe that a context modelling approach need to be customised and adapted for the problem it is applied to. In this paper, no new domain concept is introduced but a customised modularisation approach to context management is proposed.

Workflow contextualisation has been addressed in a number of works, however, there are many research challenges to make context-aware workflow systems ready for practical use. From the modelling perspective, COMPRO, a methodological approach for context-aware business process modelling and discovering relevant properties and variants has been developed [11]. A more role-driven process contextualisation and an explicit definition of the context related knowledge for adequate role-based business process modelling has been addressed in another work [12]. The works offer methodological, general or abstract approaches for context awareness which are tailored to a wide spectrum of business processes. However, context awareness should be domain-related with focus on the execution environment. Thus we propose a contextualised workflow definition designed for mobile devices.

In early stages, adaptive workflows have been described as workflows able to deal with certain changes. Later, process flexibility has been seen as the ability to deal with foreseen and unforeseen changes and four types of flexibility have been recognised: flexibility by design, flexibility by deviation, flexibility by underspecification and flexibility by change [13]. Three objectives for context-aware workflow adaptation have been identified: customisation, correction and optimisation [14]. Process flexibility can be related to design time, configuration time, instantiation time or run time. No modification of process definition at build-time is needed in the case of the flexibility by selection (a priori), as opposed to the flexibility of adaptation (a posteriori) in which modification of the process definition or its instances at run-time is allowed [15]. It has been pointed out that despite to a large number of proposals for flexible workflow support, the workflow models are still composed of activities and flexibility is usually achieved by two specific approaches: allowing runtime deviations or minimalistic specification of flow dependencies [16]. Various techniques for workflow flexibility have been designed. We do not propose another approach, rather we combine existing techniques in a specific way. Our objective is to customise workflows to user's needs with focus on workflow flexibility at run-time.

Context adaptation in a distributed collaborative execution environment is challenging because a context-aware data dissemination technique needs to be considered. Reachability, availability and roles of workflow participants are the factors that should be monitored. Ghayam et al. adopted a mobile agent paradigm and proposed a distributed solution for context management by using a decision tree-based approach [17]. A notification-token mechanism for dynamic data dissemination for Vehicular Ad Hoc Networks (VANETs) is based on opportunistic publish/subscribe system [18]. However, the mechanisms are not entirely role-based. Importance how local mobility and context awareness can influence interactions among the hospital workers is shown in [19]. The work offers a conceptual insight and outlines the significance of the inquiry for context aware collaboration but lacks in providing more details.

Workflows have become complex and better solutions are needed to manage complexity and event-driven behaviour. Whilst process-centric models are still used for most workflows, a data-centric approach to workflows has become more popular. Focus on key business-relevant objects, their lifecycles and how services invoke on them has emerged [20], howbeit, object-awareness in workflow processes is still very limited [21, 22]. The duality of information-centric and activity-centric workflow models has been emphasised as a sophisticated solution for workflow management [23]. Content integration is usually limited on the inputs and outputs of certain workflow activities, and impact on its content lifecycle would be neglected or hardly visible; therefore, a basis for an artifact-centric workflow model comprising four concepts has been proposed [24]. The four concepts are: business artifact information model, business artifact macro-level lifecycle, services (tasks), and the association of services to business artifact. In our work, the aim is to integrate a context-aware content-centric perspective into the mobile collaborative workflow process in order to support content management and context-driven behaviour.

So, in summary, there is no existing mobile workflow approach for day-to-day collaborative activities that would support context-aware content manipulation, integrate a context-aware content lifecycle and operate in a P2P context driven manner. We present our conceptual framework that can be used to define, manage and execute mobile context-aware content-centric workflow processes.

3 Research Methodology

The objective of the paper is to introduce a conceptual reference framework. The framework outlines information needed for the mobile context-aware content-centric workflow process definition, management and execution.

Firstly, we describe a case study to illustrate the use of the framework and targeted class of workflows. Secondly, context related to that class of workflows is determined. A modularisation approach to context management and aggregation is proposed. Thirdly, a workflow can be seen as an abstraction of real work. A workflow definition is basically an abstraction of phenomena in the real world and is used to create actual workflow operational instances. Workflow needs to be well defined because the success of a workflow management system is based on the quality of the workflow models put into it [25]. In this paper, we describe all constituents that form the context-aware content-centric workflows at a metamodel level. Finally, workflows are carried out by a mobile workflow management system. Therefore, we describe a logical architecture of the system.

The framework is expressed at a higher level of abstraction, therefore, it can be used as a tool for application domain experts who will be able to define the concrete workflow cases for a particular domain. The area of framework's application is exemplified in the following scenario.

4 Case Study

In this section, we present an example scenario to illustrate the motivation for utilising workflow management technology for mobile devices. The framework is tailored specifically to a certain class of workflows and this simplified scenario describes a particular workflow case that belongs to the class.

Consider a small business which focuses on residential and commercial interior design of houses, schools, offices, hotels and shops. Design projects are carried out by a team of six interior designers. Business success depends on close cooperation between designers and close collaboration with customers. Designers usually work out in the field and communicate with each other by using their smart phones. With mobile devices, designers can make important decisions right away, share images of design patterns with as little delay as possible.

Designers work on a number of concurrent projects. Although each project is assigned to a particular designer, design decisions are never done by a single person. Jane, as a specialist on private houses, has been assigned to a project to redesign a living room for a private client. The client wishes to optically enlarge the room by repositioning the furniture. In order to successfully complete the project, the following steps need to be taken:

1. Jane knows few tricks how to optically enlarge rooms so the furniture is repositioned according to her ideas. Jane takes a picture of the new room design by using her smart phone.
2. A simple rating system is used to quickly assess design ideas. Jane adds her own rating to the picture.
3. The picture is sent to her fellow co-workers. They work out in the field so receive the picture on their phones. Each of them can review the picture.
4. Reviewer's subjective opinion can be captured by adding a comment.
5. Reviews and comments are sent back to Jane. She finally reassesses her idea according to opinions of other designers.
6. If the idea is good, the picture is sent for final approval to client.
7. Approved picture is added to Jane's completed work.

From a user's point of view, the real pattern can be abstracted into a collaborative workflow as illustrated in the middle in Fig. 1.

It illustrates how a picture lifecycle can be integrated with the collaborative workflow and how the workflow can be context-driven. For example, context information about *location* and *time* are needed when the picture is taken. Latitude and longitude are coordinates that are usually captured as default by most mobile devices. However, high-level context information such as name of place or building might be more useful. Secondly, the picture goes through a number of states such as *Initial, Reviewed, Assessed, Approved or Final.* Jane can specify

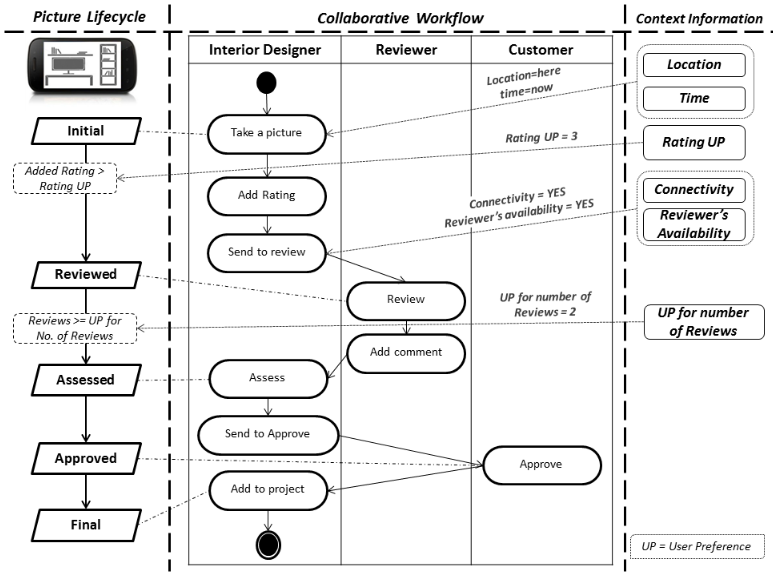

Fig. 1. Collaborative Workflow Case

that only pictures with rating larger than her *Rating User Preference* can move from the *Initial* state to the *Reviewed* state. So if the *Rating User Preference* is set to 3 and the rating added to the picture is 4, then the picture can be sent for review. A similar situation is between the *Reviewed* and the *Assessed* states. The picture can move to the *Assessed* state only if a number of obtained reviews has reached Jane's requirement. Jane sets *User Preference for no. of reviews* to 2, so two reviews are needed for her to assess the picture. Finally, sending the picture only to reviewers who are currently available would make content sharing more efficient (*Reviewer's Availability = YES*).

5 Context

5.1 Context Categorisation

People use mobile devices in a more personalised way and mobile information systems need to be more customisable and adaptable to user behaviour and needs. Personalisation and adaptation of information systems is usually accomplished by considering additional information about user, for example, by including more detailed information based on personal preferences [26]. Using user preferences in the workflow concept can lead to more flexible and dynamic workflow structures that can respond to collaborators' needs.

Secondly, smart mobile phones are capable of capturing various types of contextual information. Context information can be related to the context of device or environment. Connectivity and battery level are examples of device's context. Environmental context might be time, location or information about surrounding devices. Finally, significant context to establish cooperative effort is social context. Social context awareness relies on knowing the work context of fellow collaborators, such as their availability, current activity and location [27].

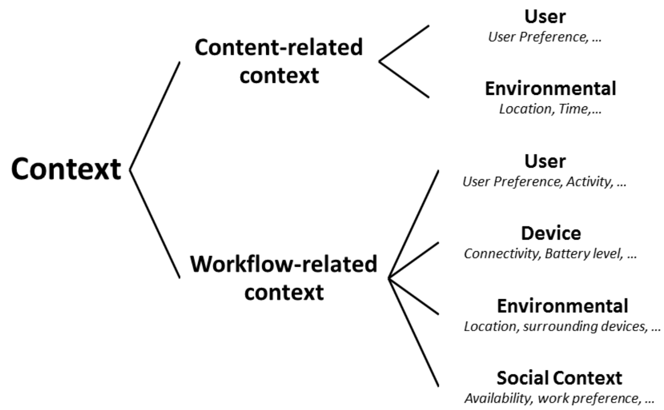

Fig. 2. Context Categorisation for the Framework

Context categorisation considered in the framework is shown in Fig. 2. We assume that whilst environmental and device's context information can influence overall workflow execution, coordinated social context-dependent workflow activities can effectively mediate the constraints of content sharing between workflow participants.

5.2 Context Management

Context management itself is not a trivial concept. We have separated context acquisition and aggregation from context adaptation. Whilst context adaptation can be considered workflow specific, context acquisition and aggregation can be workflow independent. Therefore, context acquisition and aggregation can be handled by a generic standalone mechanism that monitors, manages, aggregates and disseminates contextual information to a workflow management system and other systems running on the same mobile device. Describing the mechanism is out of scope of the paper but can be found in our previous work [28]. In the mechanism, context is described by its name and is associated with a finite set of context values. For example, context values for Bluetooth state are $\{ON, OFF\}$. In case of numeric data, high level context values are derived from raw values, e.g. if battery level is between 0 - 10%, its context value is defined as *LOW*. Context values for Battery level can be defined as {*LOW, MEDIUM, FULL*}. High-level context values can be defined by user. Context information is broadcasted in a unified form as a *(KEY-VALUE)* pair.

5.3 Context Aggregation

Context aggregation is not a new concept in the research field. This section describes its accommodation in this framework by illustrating the following examples (Fig. 3).

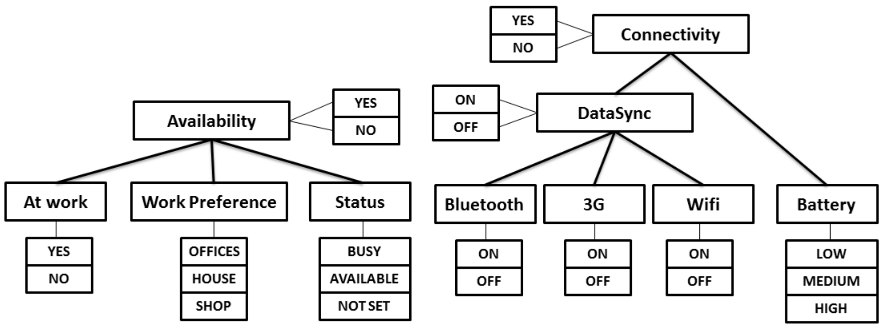

Fig. 3. Examples of Context Aggregation

Work context of fellow interior designers is expressed as a context aggregation: At work, Work Preference and Status. Context values for the 'At work' context are *YES* and *NO* representing the designer's work status. Each designer can specify own work preference, for illustration purposes only *OFFICES*, *HOUSES* and *SHOPS* are shown. The 'Status' is used to show whether collaborator is currently busy. *Availability* of the designer has an informative character to indicate whether the task can be taken by the person. The context value of *Availability* is determined by using context values of children and associated rules. The aggregated context value is broadcasted as follows: *(Availability-YES)* or *(Availability-NO)*.

The other example shows a context hierarchy for *Connectivity*. The *DataSync* component is composed of *Bluetooth*, *3G* and *WiFi*. The value sets for all of them are specified as {*ON,OFF*}. In case that any of the context is *ON*, the context value of *DataSync* is *ON* and content can be shared between devices. Content sharing might consume battery, so ideally, this operation should be performed only if battery level is not *LOW*. Thus the *Connection* component is aggregated of the *DataSync* and the *Battery* contexts. The context value of *Connectivity* is set to *YES* iff *DataSync* is *ON* and *Battery* is *MEDIUM* or *FULL*. The workflow management system gets a notification only about the aggregated context information, therefore, either *(Connectivity-YES)* or *(Connectivity-NO)*.

These examples of context aggregations are illustrative and the context aggregation hierarchy can be uniquely defined for each workflow case as shown later.

6 Context-Aware Content-Centric Workflow Process Definition

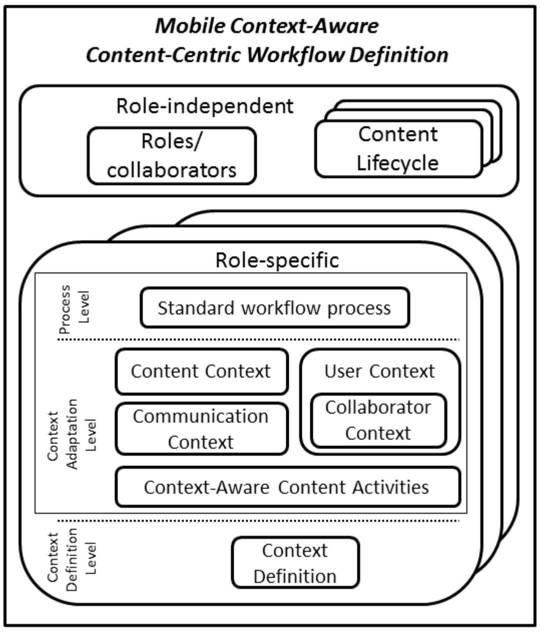

Fig. 4. Anatomy of the workflow definition

This section describes a context-aware content-centric workflow process definition designed for mobile P2P workflows. The anatomy of the workflow definition is depicted in Fig. 4. The workflow definition is composed of a role-independent part which is same for all workflow participants and a role-specific part which needs to be explicitly defined for each participating role. The parts are in details described in the remainder of this section.

6.1 Role-Independent Part

The role-independent part of the workflow definition includes a list of roles, collaborators, and content lifecycles. These workflow constituents are same for all participants.

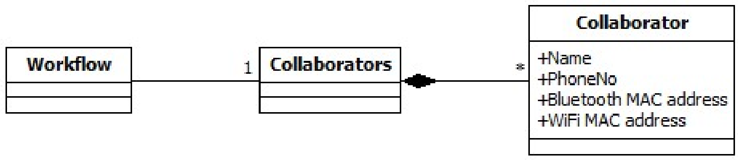

Fig. 5. Collaborators

Roles/Collaborators: In collaborative workflows, tasks are performed by collaborators. In this concept, workflow collaborator is a person who uses a mobile device to collaborate, share content and communicate with other team members in order to achieve a common goal. Each collaborator has a mobile device which is identifiable by its phone number, Wi-Fi and Bluetooth MAC addresses (assuming that SIM card is not changed). In order to execute workflows, the participating devices are aware of other fellow devices and their identifiable elements. A list of participating collaborators and their devices is predefined and available on each device. The workflow definition includes a set of collaborators as shown in Fig. 5.

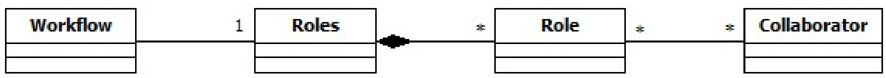

Fig. 6. Roles

A collaborator can play a number of roles in various workflow processes. For instance, Jane with the role of *Interior Designer* takes a picture and creates a workflow instance. In the meanwhile, another team member can take a picture that needs to be reviewed by Jane and creates another workflow instance. Jane will have a role of *Reviewer* in the coexisting workflow instance. Our workflow definition includes a list of roles. Each role is associated with a certain group of collaborators who can play the role (Fig. 6).

Content Lifecycle: A number of content pieces can be managed in a workflow. A set of states that each content moves through is described by a content lifecycle (Fig. 7). Transitions between a source and a target content state are associated with conditions. A condition depends on particular context. By using this structure, content management and behaviour can be context driven. A content lifecycle describes content processing across multiple mobile devices.

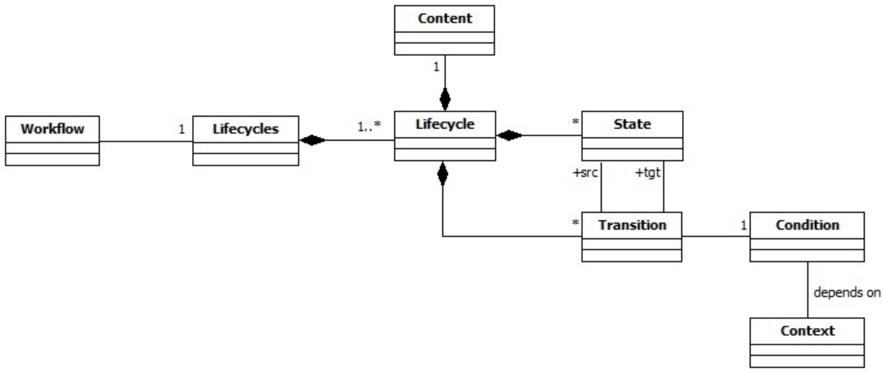

Fig. 7. Content Lifecycles

6.2 Role-Specific Parts

The workflow definition includes a number of role-specific parts specified for each participating role. A role-specific part in the workflow definition is defined for each participating role and comprises three levels: process level, context definition level and context adaptation level. The process level contains a standard workflow process definition. We use the Business Process Execution Language (BPEL) to describe our workflow processes but any workflow executable language can be used. Describing the process level in details is out of scope of the paper.

Context Definition: Context is described by its name, type and a context values set as shown in Fig. 8. For example, context can have name: *Status*, type: *User Preference* and context values set: {*Busy, Available, Not Set*}. However, in certain cases, only raw context data are obtained from context sources and high level context values need to be derived. We use *Values Descriptor* to associate high context values with raw data. Two examples: *Range* and *Coordinates* are outlined. For instance, LOW as a high-level context value for battery can be defined for a range: 0 to 10%. Composite context, dedicated for context aggregation, is designed as a context container that inherits all constructs of context. The role of the context definition constituent is to specify all atomic and aggregated contexts which influence workflow behaviour. Corresponding context values sets can be also identified. Using this constituent, context structures and hierarchies can be modelled.

Context Adaptation: As mentioned, often only aggregated context values influence behaviour of workflows. We call contexts which influence workflow execution as *active contexts*. We defined four active context types: content context, communication context, user and collaborator context (Fig. 9). Content context specifies context associated with content and its lifecycle. Communication

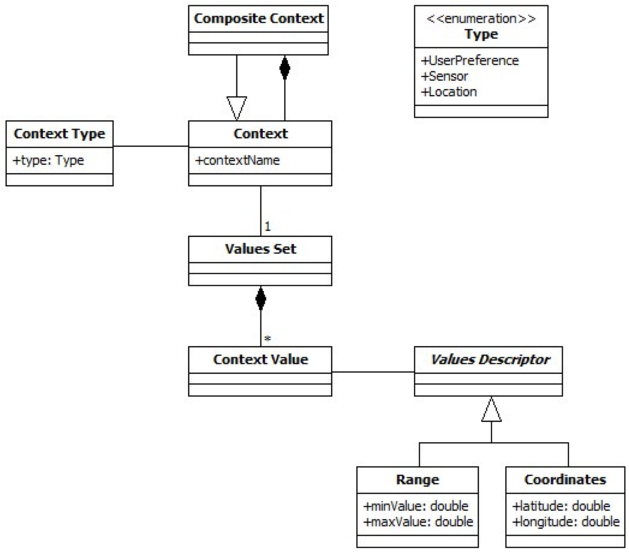

Fig. 8. Context Definition

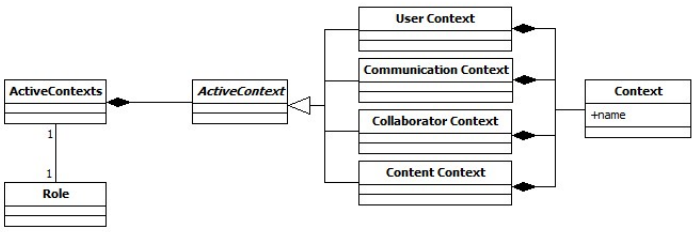

Fig. 9. Active contexts

context influences interaction between devices. User context comprises user preferences. Collaborator context specifies user's availability to perform tasks.

Context Adaptation-Content Context: Integration of content lifecycles with a particular workflow process is done by specifying content aware activities. A content-aware activity is a special type of a workflow activity. It processes content according to an action and changes its content state (Fig. 10).

7 Workflow Management

The context-aware content-centric workflows are interpreted and carried out by a system that needs to be deployed on each participating device. The system is based on cooperation between a mobile workflow management system and a

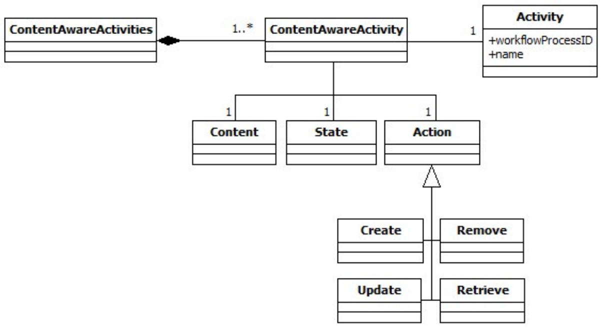

Fig. 10. Content Activities

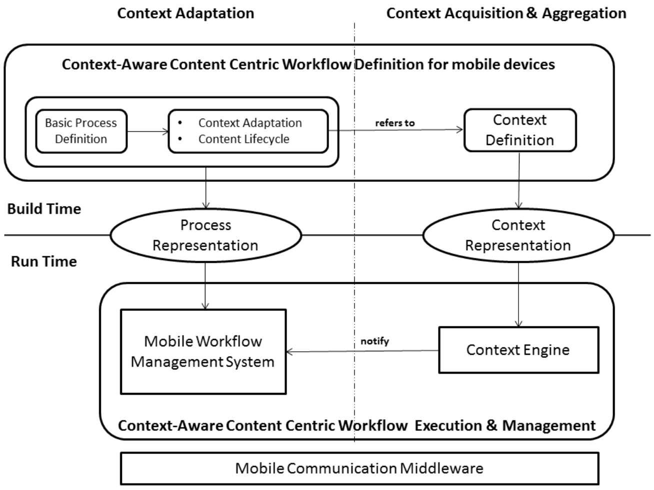

Fig. 11. Interaction between Workflow Management System and Context Engine

context provisioning engine. The interaction model, as an extended version of the reference model defined by Workflow Management Coalition [29], is outlined in Fig. 11. In the remainder of this section, we describe its logical architecture and key components.

7.1 Context Provisioning Layer

A context provisioning engine is built upon the context acquisition and aggregation mechanism. As mentioned, the details about the mechanism can be found in our previous work. The engine interprets the context definition part of workflow and operates as a context provisioning platform. The context engine supports asynchronous and synchronous communication. If a context change occurs, it is broadcasted to all listening systems and applications. On the other hand, other systems and applications can query context values at any time. A prototype of the context provisioning engine is being developed on the Android platform and is available as an open source software [1].

7.2 Workflow Management System

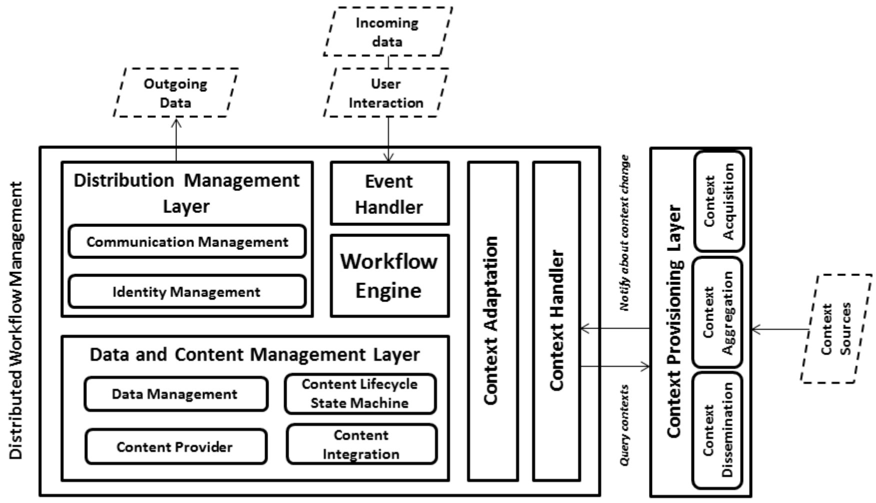

Fig. 12. The Architectural Model

A workflow management system needs to be deployed on each participating device. This section offers a logical architectural overview of the system. Describing the details of its run-time architecture is out of scope of the paper. P2P workflow management is more complex, therefore, key components grouping semantically related functions and data have been identified. The independent but interoperable components of the mobile distributed workflow management system are shown in Fig. 12. The system needs to detect, consume, and react to context events. To cope with context events, the architecture is based on an event-driven architecture pattern.

Context Handler: Interaction with the context provisioning layer is ensured by a context handler. The handler supports synchronous and asynchronous communication. The synchronous communication mechanism enables real-time interaction and context querying. Asynchronous communication is achieved by using a listener. The listener receives messages broadcasted by the context engine.

Context Adaptation: The component processes the received context messages further. Its role is to provide a run-time context adaptation support to all dependent components.

[1] https://github.com/deankramer/ContextEngine

Data and Content Management Layer: In P2P workflow management, each device needs to administer the following data and content sets:

Data Management: Most likely a mobile device will participate in a reasonable number of small-scaled workflow processes. Workflow definitions are pre-loaded on each device. As outlined, a workflow consists from a role-independent part which has to be deployed to all participating devices and a role-based part which is deployed only to the devices assigned to the role. Each workflow can have more operational instances running at the same time. In P2P collaboration, each device executes only an allocated partition of the workflow. Each device persists operational, instance and control data.

Content Provider: Sharing and manipulating of mobile content becomes a part of workflow execution. To prevent content duplication on a single device, content is stored in a read-only mode in a mobile′s default file system. This ensures that content can be seen by user at any time but cannot be modified. Only the content provider can encapsulate contents and manage access to them.

Content Lifecycle State Machine: A state machine is used to interpret a content lifecycle part of workflow and manages the content lifecycle.

Content Integration: Extended management functionalities for context-aware content are provided. Content-related context information is associated with content.

Event Handler: This component handles incoming text/media messages or events triggered by user.

Workflow Engine: The mobile workflow management system needs to be based on a good execution engine. Havey recommends the adoption of the Business Process Execution Language (BPEL) [30]. A workflow execution engine interprets workflow process definitions, instantiate them and provides logistic facilities for their completion. An existing mobile BPEL workflow engine called Sliver has been adapted [31]. Adding context awareness means that the engine must react on context changes and act upon them. This component must be flexible in order to enable realization of workflow modifications.

Distribution Manager: Distribution Manager is a component that manages collaboration between participants and dissemination of data and content. It functions as a task allocation manager and scheduler, therefore, it makes decisions what content and data need to be sent and when to send it.

Identity Management: In general, certain tasks within the process can be performed only by collaborators with a specified role. A list of collaborators is available on each device. This component interprets roles/collaborators part of workflow and provides access management to information about workflow participants.

Communication Manager: Communication manager cooperates with the distribution manager and communication middleware. Its role is to define a preferred communication method between mobile devices and decide how content or data should be sent.

The manager contains a mechanism to serialise and deserialise data. A mechanism for serializing structured data has been created by using Protocol Buffers - Google's data interchange format. By using Protocol Buffers, structured data are encoded in an efficient, language-neutral and platform-neutral format. The data structure for the workflow management system has been identified in the .proto file as shown in Listing 1.

Listing 1: Protocol buffer data structure described by using .proto file syntax

```
message TaskData {

  enum State {
    NEW = 0;
    RUNNING = 1;
    END = 2;
  }

 enum DataType {
    CONTEXT = 0;
    CONTROL = 1;
    VARIABLE = 2;
  }

  required string processInstanceID = 1;
  required string processID = 2;
  required string senderPhoneID = 3;
  optional State type = 4;
  optional string contentName = 5;
  optional string contentType = 6;

  message Data {
   required string dataName = 1;
   required string dataValue = 2;
optional DataType dataType = 3;
  }

  repeated Data workflowData = 7;
}

message TaskDataCollection {
  repeated TaskData taskData = 1;
}
```

The data structure includes the following fields:

enum State - identifies the workflow state: NEW - informing about a new process instance; RUNNING - indicating that the message is related to an existing instance; END - informing about termination of an instance;

enum DataType - specifies a type of data attached to the message: CONTEXT - context data such as location or time; CONTROL - data that drives the correct execution of decentralized workflow; VARIABLE - inputs or outputs of the tasks;

processInstanceID and processID - unique identifiers of process and process instance;

senderPhoneID - identifies the sender of the message;

contentName and contentType - are optional. If a piece of content is shared between devices, these fields are filled with corresponding information about content name and its type like picture, video etc.;

workflowData - data associated with the message; the 'repeated' word indicates that more than one workflowData can be added;

message Data - determines the structure of workflowData;

message TaskDataCollection - allows packing of more than one TaskData within one message.

Communication middleware is used to exchange messages between mobile devices. It needs to support various communication protocols for information passing and content sharing over Wi-Fi, Bluetooth or by using SMS/MMS. However, communication middleware is not part of the workflow management system. The mobile workflow management systemuses only services provided by communication middleware.

8 Conclusion

This paper presents a conceptual reference framework for definition, management and execution of mobile context-aware content-centric collaborative workflows. The workflow definition has been specified in a form of metamodel. The logical architecture of the workflow management system has been described.

The development of the framework is in progress. As a proof of concept that the theory is functional, a realistic artifact is being developed on the Android platform. Its run-time architecture will be described in our future work.

The integration of content lifecycles with a workflow process presented in this work has been only an initial step towards more advanced content integration. In our future work, our aim is to extend the workflow process in the following way. The workflow process should be capable of reacting on particular changes in content lifecycles in two ways. Firstly, querying of a content lifecycle should be supported at specific places in the workflow process. Based on the obtained current content state, decisions should be made. Secondly, certain activities in the workflow process should be able to register their interest and act only when content reaches a required content state.

References

1. Fling, B.: Mobile Design and Development. O'Reilly Media, Inc., Sebastopol (2009)
2. Erickson, J., Rhodes, M., Spence, S., Banks, D., Rutherford, J., Simpson, E., Belrose, G., Perry, R.: Content-Centered Collaboration Spaces in the Cloud. IEEE Internet Computing, pp. 34–42, September/October (2009)
3. Dey, A.K., Abowd, G.D., Salber, D.: A Conceptual Framework and A Toolkit For Supporting the Rapid Prototyping of Context-Aware Applications. Hum.-Comput. Interact. 16 (2), pp. 97–166 (2001)
4. Stephanos, A.T., Diomidis, S.: A survey of peer-to-peer content distribution technologies. ACM Comput. Surv. 36 (4), pp. 335–371 (2004)
5. Schoder, D., Fischbach, K.: Peer-to-peer prospects. Comm. ACM 46 (2), pp. 27–29 (2003)
6. Wieland, M., Kopp, O., Nicklas, D., Leymann, F.: Towards Context-aware Workflows. In CAISE 07 Proceedings of the workshops and doctoral consortium (2007)
7. Henricksen, K., Indulska, J.: Developing context-aware pervasive computing applications: Models and approach. Pervasive and Mobile Computing, 2(1), pp. 37–64 (2006)
8. Reichle, R., Wagner, M., Khan, M., Geihs, K., Lorenzo, J., Valla, M., Fra, C., Paspallis, N., Papaddopoulos, G.: A comprehensive context modeling framework for pervasive computing systems. In Distributed applications and interoperable systems. pp. 281–295 (2008)
9. Bardram, J.: The Java Context Awareness Framework (JCAF)a service infrastructure and programming framework for context-aware applications. Pervasive Computing, pp. 98–115 (2005)
10. Grossmann, M. et al.: Efficiently managing context information for large-scale scenarios. In Pervasive Computing and Communications, Third IEEE International Conference on. pp. 331–340 (2005)
11. Vara, J.L., Ali, R., Dalpiaz, F., Sanchez, J., Giorgini, P.: COMPRO: A Methodological Approach for Business Process Contextualisation. In R. Meersman, T. Dillon, and P. Herrero, eds. On the Move to Meaningful Internet Systems: OTM 2010. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 132–149 (2010)
12. Saidani, O., Nurcan, S.: Context-awareness for adequate business process modelling. In Third International Conference on Research Challenges in Information Science, RCIS 2009, IEEE, pp. 177–186 (2009)
13. Schonenberg, H., Mans, R., Russell, N., Mulyar, N., Aalst, W.: Process Flexibility: A Survey of Contemporary Approaches. In J. L. G. Dietz, A. Albani, and J. Barjis, eds. Advances in Enterprise Engineering I. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 16–30 (2008)
14. Smanchat, S., Ling, S., Indrawan, M.: A survey on context-aware workflow adaptations. In Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia, pp. 414–417 (2008)
15. Nurcan, S.: A survey on the flexibility requirements related to business processes and modeling artifacts. In Hawaii International Conference on System Sciences, Proceedings of the 41st Annual (2008)
16. Redding, G., Dumas, M., Hofstede, A. H. M. , Iordachescu, A.: A flexible, object-centric approach for business process modelling. Service Oriented Computing and Applications 4 (2010)
17. El Ghayam, Y., Erradi, M.: Distributed Context Management in Collaborative Environment. In 11th Annual International Conference on New Technologies of Distributed Systems (NOTERE), IEEE, pp. 1–8 (2011)

18. Lei, W., Ming, L., Xiaoming, W., Haigang, G.: Dynamic distribution-aware data dissemination for Vehicular Ad Hoc Networks. In 2010 2nd International Conference on Future Computer and Communication (ICFCC), IEEE, pp. 353360 (2010)
19. MejIa, D.A., Favela, J., Moran, A.L.: Understanding and Supporting Lightweight Communication in Hospital Work. Information Technology in Biomedicine, IEEE Transactions on, 14(1), pp.140–146 (2010)
20. Hull, R.: Artifact-centric business process models: Brief survey of research results and challenges. On the Move to Meaningful Internet Systems: OTM 2008, pp.1152–1163 (2008)
21. Künzle, V., Reichert, M.: PHILharmonicFlows: towards a framework for object-aware process management. In Journal of Software Maintenance and Evolution: Research and Practice (2011)
22. Vanderfeesten, I., Reijers, H. and van der Aalst, W.: Product based workflow support: dynamic workflow execution. In Advanced Information Systems Engineering, pp.571–574 (2008)
23. Kumaran, S., Liu, R., Wu, F.: On the duality of information-centric and activity-centric models of business processes. In Advanced Information Systems Engineering. pp. 32–47 (2008)
24. Bhattacharya, K., Hull, R., Su, J.: A data-centric design methodology for business processes. In Handbook of Research on Business Process Modeling, chapter 23 (2009)
25. Aalst, W., Hee, K.: Workflow Management: Models, Methods, and Systems. MIT Press, Massachusetts (2004)
26. Bierig, R.: Event and map content personalisation in a mobile and context-aware-environment. In ACM SIGIR Forum, Thesis (2010)
27. Bardram, J.E., Hansen, T.R.: The AWARE Architecture: Supporting Context-mediated Social Awareness in Mobile Cooperation. In Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, Chicago, Illinois, USA, pp. 192–201 (2004)
28. Kramer, D., Kocurova, A., Oussena, S., Clark, T., Komisarczuk, P.: An Extensible, Self Contained, Layered Approach to Context Acquisition. In 3rd International Workshop on Middleware for Pervasive Mobile and Embedded Computing at Middleware 2011. Lisbon, Portugal (2011).
29. Hollinsworth, D.: The Workflow Reference Model. Workflow Management Coalition, TC00–1003 (1994)
30. Havey, M.: Essential business process modeling. Sebastopol, CA: OReilly Media, Inc. (2005)
31. Sliver - A SOAP and BPEL execution engine for mobile devices, http://mobilab.cse.wustl.edu/projects/sliver/