SUPPORT OF SMART WORK PROCESSES IN CONTEXT RICH ENVIRONMENTS

Carl-Fredrik Sørensen, Alf Inge Wang, and Reidar Conradi
Dept. of Computer and Information Science, Norwegian University of Science and Technology (NTNU), N-7491 Trondheim, Norway.

Abstract: The evolution of mobile and ubiquitous technologies gives promises for computational services and resources to support and influence work processes planned or performed in physical work environments. Such support should optimally provide the workers with a safer work environment for both the environment itself and the workers. The extended support can give more economic and optimal work processes through proactive and situated planning and execution. We introduce the concept of a *smart work process* to capture adaptive and context-aware process support. This combination of ubiquitous computing and workflow defines a new research direction to be investigated. This paper elaborates on research challenges related to how smart work processes can be supported. We present and discuss general cases where context information or change in context information influences mobile work activities. Finally, we propose a framework for modelling smart work processes, and present a high-level architecture to support smart work processes.

Keywords: Mobile work, Mobile and ubiquitous environments, Smart work processes

1. INTRODUCTION

The future mobile and ubiquitous computing environments hold promises for computational services and resources to become invisible but important parts of the supporting computing environment for all kinds of user activities. It can be used to improve the accessibility to information and computational services. The information sources can be very diverse, from simple sensors sampling environmental properties, to complete information

systems with the ability to roam e.g. the Internet for information and services. Mobility imposes a dynamic and unstable environment that challenges how to create mobile work support environments. New opportunities to be utilised by the mobile workforce include management of ad hoc activities and cooperation with other people, as well as exchange of information and services with the surrounding environment.

Today, mobile computing and communication support is mostly directed to provide availability of services while being mobile through wireless networks and mobile computing devices. We consider this support to be an extension of existing distributed and wired services to make it possible to work distributed at different places using mobile computers.

The demanded computational support will differ from user to user, and from activity to activity. The needs are based on, e.g., what process or context information available or required by the user, and under which conditions an activity is to be performed. The conditions are influenced by temporal changes of context information as well as the behaviour of humans or other actors in the environment. Together, they constitute the state of the mobile working environment.

In this paper, we discuss and differentiate mobile work from nomadic work, and then identify issues related to how different types of context information can be supported and utilised in systems to support mobile work processes. We therefore introduce the notion of the *smart work process*. We further propose a process framework and architecture to support smart work processes.

2.   MOTIVATION

Many work places have a complex structure of participants, activities, and artefacts. This makes even simple work processes hard to plan and execute. Such environments are dynamic and unpredictable. Resources can be scarce and different participants can therefore compete to make necessary adaptation in the environment to fit their own goals. Often in such environments, safety is a very important issue for the employers and other actors in the society.

Safety for single actors can be ensured by establishing safe working conditions to perform activities within, or to re-establish safe conditions by performing situated or planned activities. Actors can influence the working environment in different ways that can create safety-breaks for other actors as well as "stealing" resources from others. By providing context information to the individual and to supervisory process enactment services, it is possible to initiate coordination activities to establish safer and more

economic working conditions. **Smart work processes** can be used as a means to coordinate multiple actors. Reactiveness to environmental changes is thus important to ensure an optimal safety and production rate. In environments that are self-aware, i.e., equipped with augmented, "intelligent" artefacts (Strohbach, 2004), activities can be initiated by the environment to ensure certain environmental goals. In addition to provide activities, coordination of actors and sequencing of activities can be performed to maintain or enhance productivity, safety, or other goals defined in the environment, by the actors or by their processes.

3.   MOBILE WORK ENVIRONMENT

Mobile work has often been looked upon as an extension of distributed work in terms of technological solutions to support such processes. Support of distributed work includes systems to manage configuration of shared artefacts, and to manage coordination and collaboration among the participants. Mobile work support systems may have need for these properties as well, but mobile work is clearly distinct from distributed work by the motivation to be *mobile* and how the dynamic change of physical environment influences how to perform work. The supporting infrastructure suffers from unpredictability and availability to the people that work in a mobile setting. In addition, mobile workers also have to face dynamic and partly unpredictable changes in the physical environment. To establish a clear definition of mobile work, *mobility* must get more emphasis to distinguish mobile work processes from distributed, co-located or individual work processes. Mobility must be a property necessary to perform the actual work, irrespective of technology. This means that work that is independent of mobility, cannot be characterised as mobile work, even if the work is performed when mobile, i.e., change of location is not a pre-condition for performing the work. We can therefore split work processes performed in a mobile setting into two different categories based on context-sensitivity:

- **Work in a mobile environment**: Work processes performed in a mobile environment *independent* of context information extracted from the physical environment. That is, mobility is *not necessary* to accomplish the process goals.
- **Mobile work**: Work processes performed in a mobile environment *dependent* of context information extracted from the physical environment. That is, mobility is *necessary* to accomplish the process goals.

A simple public transportation scenario can illustrate the difference: A bus is transporting people from one place to another. The driver is

performing mobile work because he has to adapt to changes in the physical environment. A bus passenger working with a laptop connected to a wireless network is not performing mobile work but work in a mobile environment. Figure 1 illustrates the differences between nomadic work, service work, and inherent mobile work. **Nomadic work** refers to **anywhere, anytime computing**, (often called *nomadic computing*). The goal is to provide users with access to popular desktop applications, applications specially suited for mobile users, and basic communication services in a mobile, sometimes wireless environment (La Porta, 1996). Such work is not regarded as *context-dependent*. **Service work** refers to workers that need to travel to a specific location to perform work. The work is thus context-dependent, but the context can be regarded as quite stable. **Inherent mobile work** requires continuously change of location and thus a dynamic environment. All work performed when in a mobile situation may require similar technological and computational support.
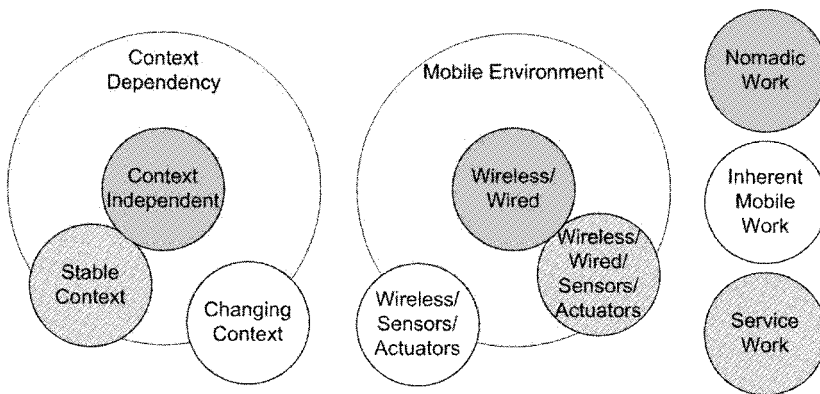

Figure 1. Mobile Work vs. Working when Mobile

To support mobile work, the evolution in mobile computing technology enables access to distributed workflow systems. However, the mobile environment is dynamic, and processes depending on changes in properties of the local environment are not particularly supported by such systems. Thus, for mobile work, it is necessary to have access and support related to the local environmental context. Traditional distributed workflow system with enabling mobile technology is therefore not sufficient to support mobile work processes.

Mobile work and the local working environment can mutually influence each other in different ways:
- *Mobile work can change the state of its environment.* This can be done directly by initiating and performing activities with the purpose of

changing the environmental state, e.g., by introducing actuators or other equipment to control or change the environment. The environmental state can also be changed indirectly through activities performed by one or more actors (workers) that together introduce state changes.

- *The state or change of state in the environment* can influence many different aspects of how, when, by whom, and what to be done in an environment. The environmental state is thus directly or indirectly used in the planning and enactment of the work process. The state or state change can:
  - *Initiate activities that must be performed.*
  - *Decide start, duration, delay, stop, and termination of activities.*
  - *Decide which activities to be performed.*
  - *Change the content or goal of an activity.*
  - *Initiate exceptions in the current activities.*

Activities and change of state in the environment together provide dynamics to reach process goals by adjusting each other. This point relates to how an environment can be regarded as an organism with a certain amount of order (state) infused with chaos through activities, where the order and chaos must balance to successfully reach a pre-defined or situated process goal.

We will in next section go into deeper details how to support mobile work in a mobile working environment.

4.   SMART WORK PROCESSES

In this section, we unify mobile work with environmental context-awareness to propose a framework for process models to be used to support mobile work. To be able to sense the environment, we need methods and tools to capture relevant context information that affects the mobile work process. This information will be necessary to drive the work support application; we therefore denote all externally sensed information as *context*.

To provide the required process support to deal with work processes that influence or are influenced by the mobile environment, we introduce the notion of a **smart work process**:

*A work process that is **sentient** (i.e., sense the environment which it performs in), **adapts** to relevant context or context changes through **context-based reasoning** to reach process goals, and **actuates** by providing situated activities, or by changing/refining the current planned activities.*

Smart work processes are thus especially tailored to make use of context information to monitor and coordinate activities within a context-rich environment. Importantly, coordination is the glue for how to manage

singular activities within an environment. Coordination needs to be performed between specific work processes and the work environment, between multiple actors performing possibly cooperative or competing work processes, and with respect to some stated paramount requirements like safety, time, and economy. The contextual relationships are not directly related to the normal relationships that exist between activities within ordinary, office-like work processes or plans. There is therefore a need for extracting and specifying how work support applications can be adjusted or adapted to also cover smart work processes.

4.1   Context-Aware Activities

An activity can be defined in a process model by providing goals, pre- and post-conditions, invariants, and the use or production of artefacts and resources. The process model elements may be affected by context information in various ways. Context or context changes can in such a model be used as pre-conditions whether to start, stop, or terminate an activity. The activity is thus directly affected by and dependent on the contextual state and hence need to include specifications and rules for how the context types are related to the activity. Context can also provide rules for how and what to do in an activity. In this case, the context is used to specify activity content and thus refining and concretising the abstract definition of an activity. Further, context can provide alternative process paths as described in Section 3. The process paths can either be pre-specified or need in situ specification when encountering new states not covered by the model. Context can also be used to trigger or create new activities not previously defined in a process model. This case is related to situated actions (Suchman, 1987) and can possibly be handled through situated planning as described in (Bardram, 1997). This means that the process model needs to be extendable in runtime to cover in situ specification and enactment.

Physical work or an actuation resulting from an activity can directly or indirectly change the physical environment (post-condition). Such a change can either be predictable and enforced through the activity and thus can be specified, or be a side-effect. Side-effects must either be included into the process model in cases where such effects are preferable, or be met by reciprocal activities to counter the side-effect. In any case, side-effects must be handled by the process enactment service. In some cases the environment is in such a state that it is necessary to perform activities to provide pre-conditions to perform new or (pre- or situated) planned activities. A related case is activities to prepare the environment for a certain planned activity. This requires a process model to be extendable based on how activities are dependent on certain environmental states to be executed.

The cases above identify how a process enactment service must take into account the surroundings to effectively support the enactment of smart work processes. The process model is in addition to the changes in the process plan and goals, also affected by state changes of elements in both the physical and the computational environments. The elements in the environment may be affected by the process, but also by other factors that might be out of control from the enactment service point of view. The environment may contain other "competing" actors, artefacts, resources and other elements that cause coordination problems between actors, the physical and computational environment itself, and the process goals of the different actors.

4.2   Situated actions and planning

The state of the physical work environment will often be fluctuating and dynamic, and it is thus very hard to provide realistic or detailed plans beforehand for which activities to be performed. Definition or refining of activities based on the perceived state of the environment will therefore be necessary to create executable process models. Some activities can also be defined in situ by the environment itself. This is possible if the environment is self-aware and augmented with applications and technology making the environment able to change its own state as well as ask for concrete actions by external or passing (human) actors not statically bound to the environment. The state of the environment provides means to create activities to change the environmental state to a level where a planned goal or intention may be accomplished.

Situated planning and thereby actions will therefore be a natural part of a support environment for smart work processes. Context information is thus used for both definition and adaptation of work processes in situ.

4.3   Challenges in context-aware process support

The support for context-aware or smart work processes is to a small degree covered in the literature and is thus a new area of research within pervasive computing and workflow/process support. Since the domain has little coverage in the research community, many challenges arise in the cross-section of mobile/wireless computing, ubiquitous/pervasive computing, and workflow:

- Specification of *contextual pre/post-conditions* related to some process goal.
- Specification of *environmental behaviour* related to an activity (adaptation).

- Specification of a ***uniform representation of sensors and actuators*** from a process enactment perspective to make it possible to reason about and change the context state of the environment.
- ***Planning, specification, and execution*** of activities in concert with the current environmental context. These challenges relate to how the dynamics are handled by the workflow enactment services, both on client and on server.
- ***Managing process changes to ensure a consistent state*** of the process.
- ***Managing the dynamics of ad hoc activities and process changes*** locally and in a central enactment service.

5.   A PROCESS FRAMEWORK FOR SMART WORK PROCESSES

Process models are used to define how work should be performed and can be used for visualising processes, guiding the user through the process, automating steps of the process, educating users about new processes, analysing or simulating the process etc. To define a process model, a process modelling language (PML) must be used.

The Workflow Management Coalition (Hollingsworth, 1995) provides a reference model for how to build a workflow management system and how to make interfaces to other applications and workflow systems. To support smart work processes, some additional or changed requirements are necessary to provide extended support of context-aware work processes. We therefore specify some of the requirements to a process support system supporting smart work processes:

The process model must be able to ***adapt*** during the process enactment. This is partly covered through the use of exceptions in the reference model.

The process model must be ***refinable/extendable*** by situated planning. Working in a dynamic environment with coarse plans is quite normal. Such plans are refined or extended based on the properties of the working environment itself. Activities resulting from situated planning are not possible to plan beforehand, but systems can use encountered processes to "learn" about specific activities performed during specific situated work situations.

A ***separate environmental or world model*** must be built up to support specific processes by identifying relevant context sources that may affect or support the process.

Rules must be specified or developed for how the process model can adapt to the relevant context information. The rule model defines how the process and the environment may or should interact during the process

enactment. The rules may be pre-defined, but must also be derived from the need for situated actions, the sensed state of the environment, and the process itself. In addition, the environment can provide rules based on artefacts augmented with smart sensors and some representation of "intelligence" (Strohbach, 2004).

The working environment must have "services" that can be given values of certain required properties or conditions to be satisfied during the process or activity execution, e.g., the environment is supposed to keep a certain temperature during a process. A break in such conditions can contradict the process goal and in the worst case endanger the environment itself or the people in the environment.

A *coordination service* is necessary to coordinate the different actors with cooperating or competing process goals or plans. The actors might be loosely coupled through the environment and not directly aware of each other presence in the environment.

Specific environments can have pre-defined activities or plans that can be put into enactment in certain pre-defined environmental states, e.g., defined safety procedures when the safety requirements are not satisfied.
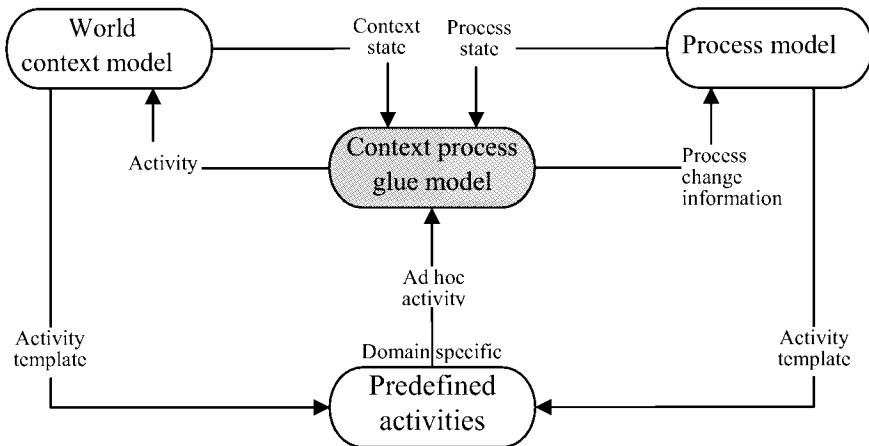

Figure 2. Smart work process glue model

Figure 2 shows a proposal of model needed to provide support for smart work processes and how these models relate. A central part of modelling smart work processes is the representation of the context relevant for the work process.

The *world context model* represents objects in a ubiquitous computing environment, often embedded with sensors and actuators. Such a world model should consist of all relevant static objects in an environment that somehow can influence the work processes. In addition, mobile objects that can provide relevant context information must also be integrated/included in

such a model. Also non-tangible properties like e.g. weather, sounds, and light conditions may be relevant and therefore need to be included in the model. The construction of a relevant world model poses challenges since this means that the world model itself must be able to change when objects are approaching or leaving the environment. The static world model is, however, easier to comprehend and therefore more easily represented in a model. Other challenges are how to specify and select the relevant objects in an environment to be included in the specification and enactment of context-aware work processes. Such a selection is necessary both when pre-planning the work process and when using situated planning. The world context model must represent the relevant objects in a way that make it possible to catch the state of the physical or execution environment. The state is needed to reason about and take according actions in the work process support system. A challenge here is to have a context model that is able to provide a unified representation of objects/sensors that can represent various physical data that can span from boolean values to complex, composite data types representing abstract properties in the environment. A *process model* is also needed that model the activities of the work process. Such a model must allow late binding and it must be possible to reconfigure and change the model during the process enactment.

A key in modelling and supporting smart work processes is the *context process glue model*. This model is a loose coupling between the world context model and the process model that defines rules for how the context should affect the process model and vice versa. The glue model is responsible to connect pre-planned processes with situated planning, and thus process enactment. For pre-planned processes that are context-sensitive, we proposed to use templates that can be refined during context-aware, situated planning. To deal with unexpected conditions in specific work environments, we propose to have a set of *predefined activities* used to handle such conditions. The predefined activities can dynamically be added to the process model and activated based on the new environmental state. A library of predefined activities will grow over time based on gained experiences on solving unexpected events. The predefined activities are influenced by the process model as well as the world context model.

6.   AN INFRASTRUCTURE TO SUPPORT SMART WORK PROCESSES

The future computing environment is predicted to realise the vision of the invisible computer (Weiser, 2001). Computing devices are in this scenario embedded in almost every kind of physical object. Sensors are

spread like "dust" in the environment or appear as "brilliant rocks" (Satyanarayanan, 2002). This vision makes the computing environment extremely distributed and gives indications of an enormous amount of possible context sources and services to autonomously support people in their whereabouts. Augmented, intelligent artefacts can provide self-aware computing elements in the environment that can cooperate to provide rules and actions used by the process support system. In addition, actuators can provide services that can change the state of the environment. Such actuators can be robot-like and thus have the possibility to perform a limited number of activities (partly autonomous, mobile units with communication capabilities).
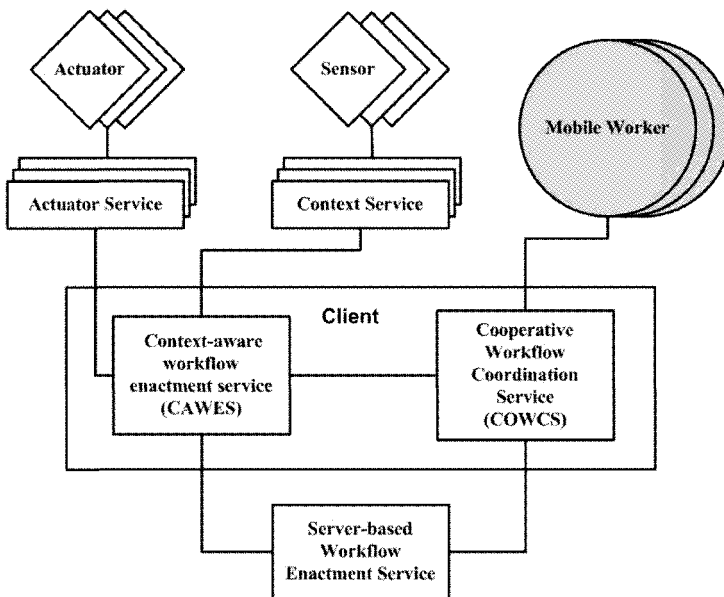

Figure 3. An Architecture for smart work processes

We have developed a high-level architecture as shown in Figure 3. The architecture consists of seven main parts which will be described in more detail below.

Actuator: An actuator is an entity that is able to change the environment based on digital input from some digitial equipment. In our architecture, an actuator is responsible for acting according to specifications instructed by the actuator service.
Sensor: A sensor is responsible for measuring, and transmitting sensor readings according to specifications by the context service. Sensors might vary from very simple, to autonomous, augmented entities.

Actuator Service: An actuator service is responsible to initiate actuations in an environment. The service should be able to receive instructions from the workflow enactment service and translate these to the actual actuator. The actuators can in this service be represented as abtract entities with published properties. Actuator services can hide how to communicate with the different actuators and provide a well-defined interface to be used by the service clients. The actuator service should be able to communicate with actuators through various physical communication implementations like IR, WLAN, LAN, BlueTooth, etc.

Context Service: A context service is responsible for identifying sensors, initiate sensor readings, check sensors, setup/terminate sensor subscriptions, etc., i.e., provide the communication with all actual sensors. The context service can accept subscriptions from different clients and is then responsible for setting up the sensor subscription properties. The clients can specify which properties they prefer to receive. The context service is responsible for handling client sensor subscription information including receiving rules, preferences, and facts related to sensor properties; collect/aggregate basic readings into abstract context (e.g. weather conditions). Based on the properties received from the various clients, the context service should send context messages to the client subscribers related to subscriber preferences. A context service may be extended to also represent e.g. augmented artefacts. It should then be able to communicate process fragments based on templates initiated by certain contextual states/conditions either contained within the context service, or received from smart, autonomous sensors. Possible context services that can be provided are e.g. weather conditions, engine and application states, etc. (including the state og the client itself), location, and the mobility of the user and other objects/entities within the environment.

Client-Based, Context-Aware Workflow Enactment Service (CAWES): CAWES is responsible for monitoring and executing delegated activity (ies), through surveillence of the current process/activity state, and contextual events received either from the context service, the COWCS (see below), or the Server-based Workflow Enactment Service. CAWES can also receive process/activity fragments from the context service, the COWCS, and the Server-based Workflow Enactment Service, can send its own process/activity state as contextual events to the COWCS, and the Server-based Workflow Enactment Service. If it is a need for actuations, CAWES initiates actuations using the Actuator service. Further, it can display/inform about environmental state wrt. the process itself, safety state, etc. It can also display/inform about information related to the activity performed. Such information can be based on the activity itself (manuals, checklists, or multimedia information), location-based information, safety information, etc. In additions, CAWES should also provide a traditional workflow user interface for filling in information, create/update artefacts related to the

activity, e.g. interactive checklists (can also be filled automatically using the context service), manage/integrate audio/video/textual information collected through the device or an actuator service), etc. An extension of CAWES is to provide virtual/augmented, location-based information (manuals, multimedia) related to the working environment, process type, available tools and sensors.

Client-Based Cooperative Workflow Coordination Service (COWCS): COWCS is responsible for managing and coordinating activities in a multi-actor environment. This includes a coordination service that based on policies can send messages to all involved parties about coordination needs, competitive resources, etc. COWCS receives process goals, and currently performed activities from the other actors. Similarly, it can send its current state of executing activity and possibly prepare other users about new activities to be started within a time limit, and coordination messages to other users including needs for artefacts, services, actuators, and a preferred environmental state. Internally on the client, COWCS sends contextual events to the CAWES based on coordination reasoning. Such events may include information about the need for artefacts, services, actuators and an environmental state with priorities, time, demand, etc. It can also send new process fragments derived from the coordination service or received from other clients based on demand/need from other users. The fragments are sent both to the CAWES, and to the Server-based Workflow Enactment Service.

Server-based Workflow Enactment Service: The Server-based Workflow Enactment Service is responsible for traditional workflow management including sending activities to the users, based on delegation/plans, and managing the complete, high-level process for the involved participants. The architecture has not been completely validated, but we have developed some prototypes to start the process of validating the architecture and the concepts behind it. Unfortunately, few work environments have been sufficiently instrumented to try out the proposed technology in practice. This situation, we hope will be improved in a few years.

7.   RELATED WORK

(Wiberg, 1999) state that "anytime, anywhere" does not necessarily mean "everytime, everywhere". The ideal mobile situation is not to work continually without any stops. Further, true mobility goes beyond mobile support for "here and now". There is also a need to support the place to go, and the place left behind as well as to make plans for the future or backtracking earlier events. Mobile work is in many cases a kind of stationary work because the worker has to stop to perform any real work when work is physically oriented.

Situated actions (Suchman, 1987) and situated planning (Bardram, 1997) are terms that relate to our approach. It is hard to make predictions of which actions to perform in dynamic environments, i.e. the environment itself is influencing the actions based on the state of the environment. This motivates for a broader use of contextual information to support both the execution f actions, and also the planning of them. Situated planning can be used to refine coarse-grained plans based on the contextual properties of the environment and the worker itself. Situated actions can be used to create immediate plans to be used in situ.

The CORTEX project proposes the sentient object (SO) programming model (Fitzpatrick, 2002) for pervasive and ad hoc computing applications. Smart work processes can be looked upon as sentient objects. A smart work process can be decomposed into smart activities or process fragments that also can be regarded as sentient objects.

All the work presented above is important to enable sound architectures and designs for smart work processes. Thus, both conceptual and technical issues can be addressed by using similar design frameworks to create a workable infrastructure to support mobile work where context information plays an important role for a successful enactment of smart work processes.

8.   DISCUSSION

We have in this paper not discussed all challenges that may arise when working in mobile, ubiquitous computing environments. We have, however, identified a few issues that may be hard to address to ensure a safe and dependable support of smart work processes:

Supporting process state transitions and process enactment when disconnected from a central work process support server: When disconnected we need to establish local enactment services in addition to a central enactment and coordination service. It must also be possible to coordinate the clients decentralised in an ad hoc fashion.

Ad hoc activities based on environmental context: The definition of ad hoc activities can be done in at least three ways: Such activities can be automatically defined by inferring activities through context-sensing and reasoning, they must be specified semi-automatically (using partly pre-planned templates), or be specified manually. This requires knowledge about the relationship between the context state and activities to be performed to reach an either pre-planned, or a situated goal. The activity must also be directed to the correct worker and then be coordinated with other activities in the environment. The state initiating the ad hoc activity can at the same time also influence the other activities in the environment creating conflicting

goals or incompatible activities. This requires coordination activities to solve inconsistencies as well as optimising the activity throughput in the environment.

Ad hoc collaboration between environment and users, user to user: Collaboration between the different actors in a working environment is often not specified explicitly. In some cases, the collaboration is specified in the PML, but often collaboration can be initiated by the current situation or state in the environment. It is a challenge to create support systems that are able to create such process ad hoc relationships.

Human considerations: Context-aware systems (Bellotti, 2001) must consider properties like intelligibility and accountability to give dependable systems providing added value to the user. We envisage the need to provide intermediate evidence of the applicability and dependability of the solutions, and that the number of different kind of context sources to be included in the systems must be slowly increased to keep the systems manageable and accountable. The importance of *which* context information to include *when,* will therefore be a paramount issue to evolve applicable rules. Learning processes should therefore be included to evolve the inference engines embedded in the systems.

Technical issues and solutions related to mobility, mobile devices, wireless networks, and the inherent properties of these, have not been covered in this paper. Such issues are important to address to successfully implement support for smart work processes.

9. CONCLUSION

We believe that the concept of a *smart work process* is useful as an abstraction of adaptive, context-aware work processes used to model and support mobile work in future ubiquitous computing environments. We have in this paper clearly distinguished mobile work from nomadic work to better illustrate how future process support should be developed. Current process modelling languages do not support situated processes directly influenced or created by the environment. Such processes are mainly unknown to the work process support systems until the situation has occurred in which the process is to be performed in. Context information captured in the working environment and used in work support applications can in the future be important to keep control of dynamic work environments. Coordination and implicit cooperation between the environment and participating actors can ensure a safer and more economic work environment directing and coordinating work activities using all relevant context information that can be captured using sensors and reasoning techniques. A lot of work remains

before the vision of smart process support systems can be realised. Especially in multi-actor environments and environments with autonomous equipment like robots, the need for supervision and adaptation of work processes is important to ensure safety, and economy through better utilisation of the available resources.

References

Atluri, V. and Chun, S. Handling Dynamic Changes in Decentralized Workflow Execution Environments. In *Lecture Notes in Computer Science*, pages 813–825. Springer-Verlag, 2003. LNCS 2736.

Bardram. J.E. Plans as Situated Action: An Activity Theory Approach to Workflow Systems. In *5th European Conference on Computer Supported Cooperative Work*, Lancaster University, UK, 7-11 September 1997. Kluwer Academic Publishers.

Bellotti, V. and Edwards, K.. Intelligibility and Accountability: Human Considerations in Context-Aware Systems. *Human-Computer Interaction (HCI) Journal. Special Issue: Context-Aware Computing*, 16(2–4):193–212, 2001.

Fitzpatrick, A, Biegel, G., Clarke, S., and Cahill, V.. Towards a Sentient Object Model. In *Workshop on Engineering Context-Aware Object Oriented Systems and Environments (ECOOSE)*, Seattle, WA, USA, November 2002.

Hollingsworth, D.. The Workflow Management Coalition – The Workflow Reference Model. Technical report, Workflow Management Coalition, Lighthouse Point, Fla., January 1995. Technical Report WFMC-TC00-1003, version 1.1.

La Porta, T. F., Sabnani, K. K., and Gitlin, R. D.. Challenges for Nomadic Computing: Mobility Management and Wireless Communications. *Mobile Networks and Applications*, 1(1):3–16, 1996.

Satyanarayanan, M.. Of Smart Dust and Brilliant Rocks. *IEEE Pervasive Computing*, 2(4):2–4, October-December 2003.

Strohbach, M., Gellersen, H., Kortuem, G., and Kray, C. Intelligent Artefacts: An Embedded Systems Approach for Cooperative Assessment of Situations in the World. In *The Sixth International Conference on Ubiquitous Computing (Ubicomp 2004)*, Nottingham, England, September 7-10 2004.

Suchman, L.. Plans and situated actions. The problem of human-machine communication. Cambridge University Press, 1987.

Weiser. M.. The Computer for the 21st Century. *IEEE Pervasive Computing*, 1(1):18–25, January-March 2002. Reprinted from Scientific American, 1991.

Wiberg, M. and Ljungberg. F.. *Exploring the vision of "anytime, anywhere" in the context of mobile work*. Idea Group Publishing, 1999. Ed. Yogesh Malhotra