Multi-agent Technology as an Enabler of Computer Supported Cooperative Work for the Mobile Workforce

Habin Lee, Patrik Mihailescu, and John Shepherdson

Intelligent Systems Research Centre, BT Group,
B62 MLB1/pp12, Adastral Park, Martlesham, Suffolk, IP5 3RE, UK
{ha.lee,patrik.2.mihailescu,john.shepherdson}@bt.com

Abstract. This paper proposes an agent-based computer supported cooperative work system known as TeamWorker that enables cooperation between mobile workers for the execution of their jobs by providing services via conversational components (C-COMs). C-COMs encapsulate agent interaction protocols, which can be dynamically plugged into participating agents. Furthermore, the intelligence and autonomy of agent technology is applied to overcome the barriers faced in mobile computing environments, such as difficulty in monitoring work progress, device diversity, need for multi-modal human agent interface, etc. Finally, we discuss how TeamWorker has been applied to a real mobile business process in a telecommunications company in the UK.

1 Introduction

There have been many investigations into the IT support required to facilitate cooperation of team members in cooperative information systems [18] area, however, most of these are targeting an office-based workforce. On the other hand, a mobile workforce requires different types of IT support due to the inherent constraints faced within a mobile computing environment such as mobility of workers, device diversity, and unreliable network connections.

Based on the experiences from two field trials in which a multi-agent platform was applied to support a mobile workforce in the UK and Germany [2], this paper addresses three main issues that need to be challenged by information system developers to support the coordination of a mobile workforce. Firstly the reach-ability of each mobile worker should be considered. Typically mobile workers spend the majority of their time either in transit, moving from one location to another, or working in the field. This prevents them from constantly monitoring a device screen (and using the application), as well as preventing the real time tracking of assigned jobs from an administrator's point. Technologies such as text to speech and speech recognition can enable users to interact with their colleagues in a timely manner. Secondly the diversity of mobile computing devices should be considered. Mobile computing devices differ significantly from each other in terms of not only their hardware, and operating system functionality but also their physical appearance, and usability properties. This prevents the use of a fixed GUI client for interaction with

M. Klusch et al. (Eds.): CIA 2004, LNAI 3191, pp. 183–198, 2004.

the user, as a GUI developed for a PDA cannot be used on a mobile phone, which has a smaller screen, and a limited keypad. Thirdly the flexibility to adopt new coordination services should be considered. Typically once an information system is rolled out, additional services may need to be deployed, including enhancements to existing services. Some of these services may only be applicable for specific types of mobile computing devices and users that satisfy certain preconditions. Therefore the provision of services should be on demand, considering both the computing environment at the time of installation, and the user role.

This paper proposes TeamWorker as a Computer Supported Cooperative Work (CSCW) system based on multi-agent technology to support the coordination of mobile workers. In TeamWorker, an intelligent agent supports each mobile worker for the cooperative execution of assigned works. The requirements identified above can best be supported by MAS technology due to the following aspects. Firstly, MAS technology provides well defined interaction protocols (or conversation policies) that are essential for the implementation of location-based coordination of remotely distributed workers. Secondly, the intelligence of a software agent can be applied to capture the progress of a mobile task by monitoring the actions and location changes of a worker. For this purpose, the internal architecture of a software agent can be applied to capture the belief, desire, and intention of a human worker. Thirdly the autonomy of an agent can be applied to increase the usability for mobile workers who have difficulties in making real time responses due to their frequent movements, and working with field equipment. Furthermore, an agent can autonomously configure GUI components or services for a user considering the constraints imposed by a specific mobile device.

This paper is organized as follows. Section 2 reviews related work and section 3 details the main features of TeamWorker. In section 4 TeamWorker is applied to a real mobile business process, and finally, section 5 summarizes and concludes this paper.

2 Related Works

CSCW [12] has been considered as one of the key technologies to implement a virtual workspace that enables distributed team-members to work together, regardless of location and time differences.

Reviewing the current CSCW systems, they can be characterised by the use of network infrastructure to facilitate the communication, collaboration, and coordination of team members. Examples of CSCW include an electronic meeting system [16], a group decision support system [21], a workflow management system [11] and so on. Most of these technologies are based on the assumption of a reliable network connection for the transmission of high volumes of multi-media data, which limits their application to in-office work.

Handheld CSCW [20] or mobile CSCW [22] extends the application area from in-office work to mobile work. Domingos et. al. [6] propose a mobile CSCW system to support mobile cooperative work from the workflow management system's perspective. However, they are considering the situation where each mobile worker carries a mobile device which is disconnected from a central workflow server most of the time (which is not the case in many modern mobile work configurations, wherein mobile workers are connected with a workflow server all the time via GPRS). One of main advantages of using mobile CSCW is facilitating knowledge sharing among mobile workers. Fagrell et. al. [8] identified four aspects of mobile knowledge management as it took place in the mobile work setting: sharing, i.e., several parties exchange knowledge; indexing, i.e., one party explains to another which knowledge to retrieve; diagnosing, i.e., two parties make sense of how to interpret a situation, and; foreseeing, i.e., one party (or more) uses knowledge to project the future. RoamWare [22] has been developed as a mobile CSCW to support physical/virtual meetings of mobile workers and evaluated in Umea University and Telia Nara. From the evaluation, they argue that the meetings among mobile actors are crucial aspects of the knowledge-sharing, coordination and decision-making processes. Fagrell et. al. and RoamWare focus on the knowledge management perspective of mobile cooperative work, whereas in this paper we propose the provision of knowledge management services in a workflow context.

Recently, the role of MAS technology to implement CSCW has been highlighted and research that applies it to the coordination of distributed workers is sometimes categorised as Agent Supported Cooperative Work (ASCW) [24]. For example, Ellis et al. [7] apply agent technology to improve the performance of an online meeting system. One of the main areas in ASCW is a workflow management system wherein agent technology is used to increase the flexibility of the system [1][4][13]. However, such research is more focussed on workflow task scheduling and does not address the coordination mechanism among process actors. Furthermore, it does not address the issues specific to the mobile computing environment, such as device diversity, frequent disconnection, the nomadicity of workers, etc.

3 TeamWorker: Agent Support for Cooperative Mobile Work

TeamWorker is an agent-based CSCW system that has been implemented using the JADE-LEAP [1] multi-agent platform. This platform is Java based, and is a version of the JADE multi-agent platform that has been designed to run on a variety of mobile computing devices. At present, two types of device configurations are supported by JADE-LEAP: i) PDA, and ii) Mobile phone/smart phone. The main component of TeamWorker is a smart agent, called a Personal Agent, which resides on a mobile computing device, and plays a personal assistant role for a mobile worker. This section shows how a personal agent supports coordination between mobile workers by overcoming the constraints identified in section 1. The following sub sections introduce firstly the internal architecture of a personal agent (3.1), followed by a discussion on how the personal agent can adapt the application based on the underlying mobile computing device (3.2). In sub section (3.3) a discussion is given on how TeamWorker provides a flexible provision of services, and in sub section (3.4) we conclude with a discussion on ‘always on’ awareness of a personal agent.

3.1 Personal Agent Architecture

Fig. 1 shows the internal architecture of a personal agent that has been designed to support the coordination and activities of mobile workers. The architecture consists of three layers: i) Human-Agent Interface layer, ii) Agent Internal Layer, and iii) Coordination layer. The Human-Agent Interface Layer handles the management of interactions between the personal agent and the user. The main functionality of this layer is to determine which GUI components should be used depending upon the properties of the underlying mobile computing device, and to determine the medium (text or speech) to use during certain interaction requests from/to the user.

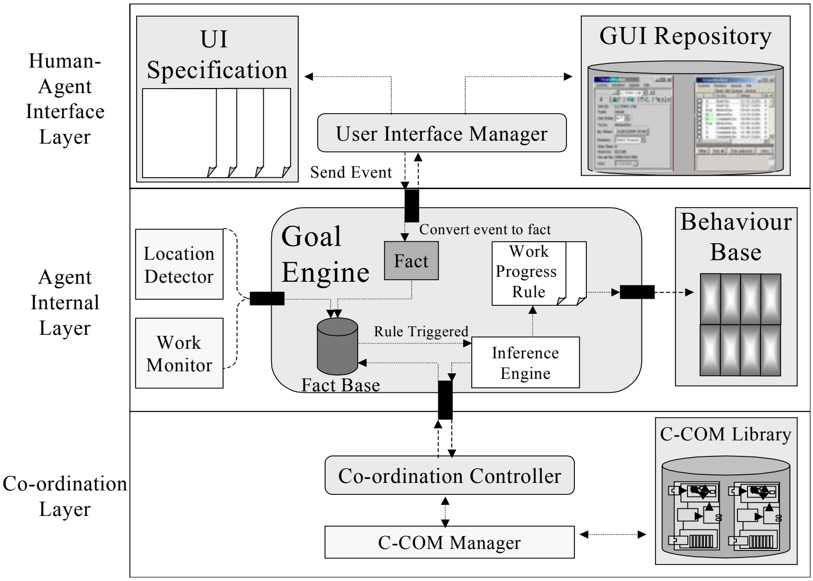


Fig. 1. Personal agent internal architecture

The Agent Internal Layer determines the optimal actions to undertake to achieve the best provision of services required for the execution of cooperative work. The main features of this layer are the goal engine that is comprised of the following: i) a fact base, ii) a set of rules, and iii) an inference engine. The fact base contains all the acquired knowledge of the personal agent. The set of rules comprise both an initial set of pre-defined (static) rules as well as dynamic rules that can be added during the execution of the personal agent. Finally the inference engine determines, depending upon the facts contained within the fact base, if a rule has been triggered. The other parts within this layer are the behaviour base that contains a set of agent behaviours, a location detector that tracks the current physical location of the user, and the work monitor that monitors the current progress of the user's work stack. When an event is received within this layer, it is initially converted into a fact and then stored within the fact base. Depending upon the type of fact, a rule maybe triggered and a set of actions may then be performed. Finally, the Coordination Layer consists of a coordination controller, and a C-COM manager that manage the service components installed within the personal agent, and also interact with other mobile workers (represented by their personal agents), to execute coordination based services.

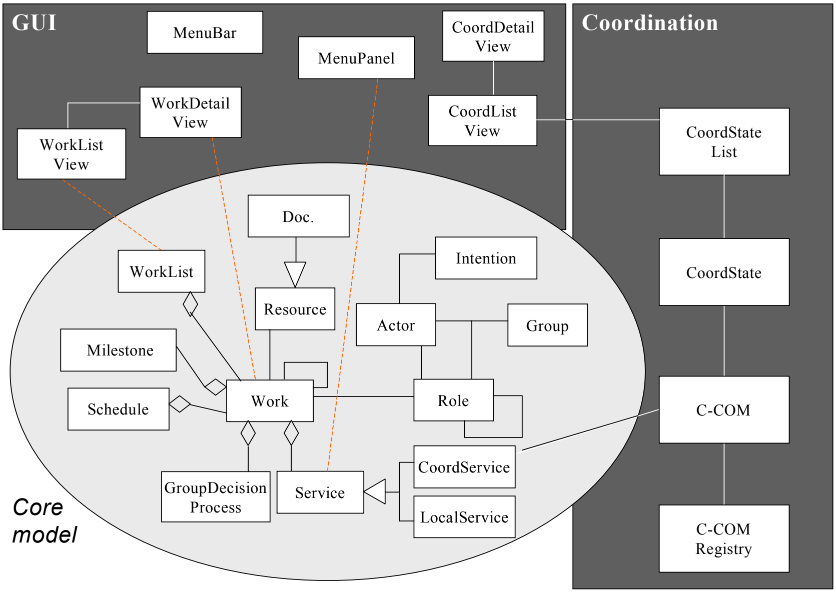


Fig. 2. The meta-model of cooperative work context in TeamWorker

3.2 Adapting to Application and Device Specificity

One of the main features of a personal agent is the capability to configure GUI components and services according to given specifications on a target application or a device. The meta-model described in section 3.2.1 shows the contextual information wherein a personal agent supports the execution of mobile work. The application developers use the meta-model to provide specifications on their application specific features such as additional attributes for some ontology items, application specific GUI components and services. A personal agent uses the specifications to configure corresponding application systems on specific devices.

3.2.1 Metamodel for Cooperative Work

A general set of ontology items have been defined that outlines the scope of the cooperative work context within TeamWorker as shown in Fig. 2. Using the knowledge contained within the work context, a personal agent is able to guide the user during the execution of jobs by selecting the most appropriate services, and GUI components based on the current work context. For example, the work context may contain details such as the current location of the user and their proximity to colleagues, and the current state of their work schedule. The central ontology item within TeamWorker is ‘Work’ that represents a unit of a job that should be completed by a worker or a group of workers. Work is assigned to a user (as represented by an Actor) via a Role concept that determines their responsibilities and privileges. During the execution of Work one or more Resources may be consumed such as a Document or a Service, and a Group Decision Process may also be required to solve any problems that arise during its execution. One example of a Group Decision Process is the co-ordination of multiple team members to a single job site.

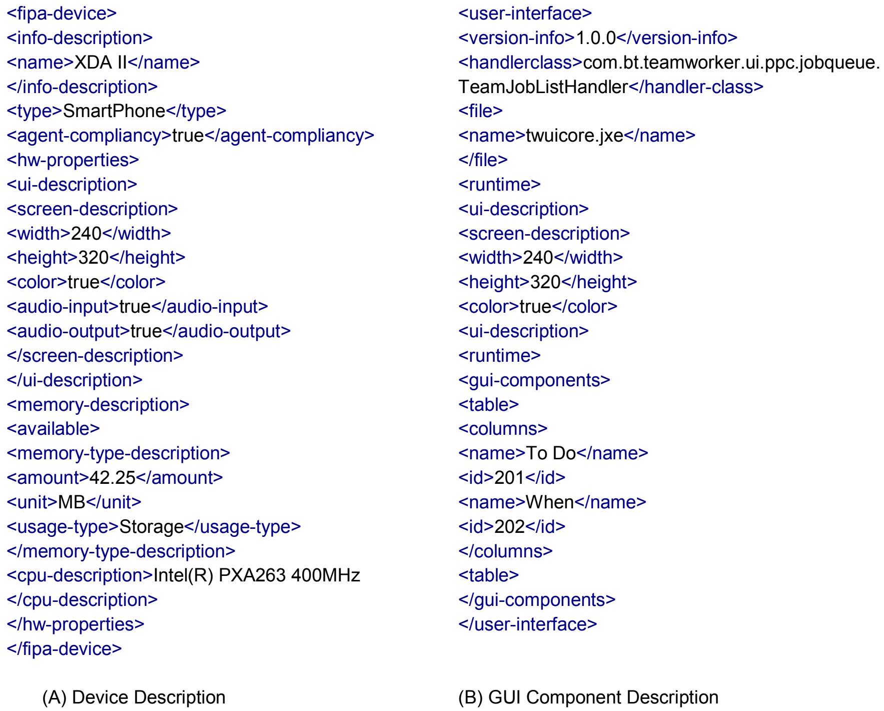
Work can be part of a Schedule that represents the chronological order of the execution of each Work item assigned to a worker. Each Work item has one or more Milestones that represent the execution progress of a Work item. A Service is classified either as a Coordination Service or a Local Service. A Coordination Service is delivered by an executable software component known as a C-COM [15], which captures all information related with the coordination such as work progress, involved workers, location, etc, within a Coordination State item. Work can either be viewed individually in more detail, or as a group as defined respectively in the WorkDetail View and WorkList View GUI ontology items. Other GUI ontology items such as MenuPanel define how services can be accessed, and the CoordList View and CoordDetail View define how a user can view the current status of coordination-based services. These ontology items can be extended upon or new ontology items created according to an application’s requirements.

3.2.2 GUI Configuration

At the first launch time, each personal agent refers to a device description, which contains details on both the hardware and usability properties of the underlying host device. The device description is based on the FIPA device ontology specification [9], and a shortened example is shown in Fig. 3. Using this information, a personal agent is able to determine the best configuration of the GUI by only selecting those GUI components that meet the device constraints, and then adapting them to meet the user/application requirements. Each GUI component contains a description of the runtime properties required for its use (e.g. screen size, library support), and a list of customisations that can be made. For example, a table widget contained within a GUI component may allow both its column and data source to be customised. An example of a GUI component description is shown in Fig. 3.

3.2.3 Service Composition

A personal agent composes and provides services to aid the execution of jobs for the user. For this purpose, a personal agent maintains a service hierarchy that is represented as a tree for a specific CSCW application. Fig. 4 shows a sample service tree and its XML-based description. A service tree shows the order of services that



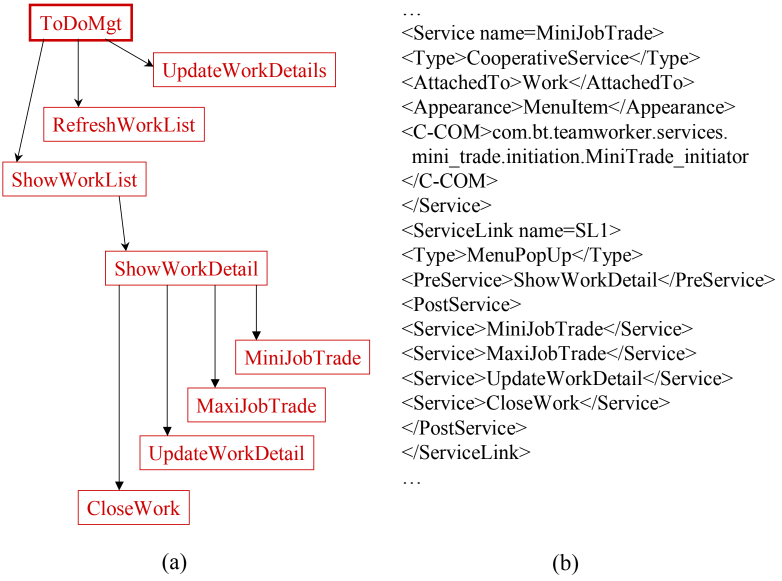
```
<fipa-device>
<info-description>
<name>XDA II</name>
</info-description>
<type>SmartPhone</type>
<agent-compliancy>true</agent-compliancy>
<hw-properties>
<ui-description>
<screen-description>
<width>240</width>
<height>320</height>
<color>true</color>
<audio-input>true</audio-input>
<audio-output>true</audio-output>
</screen-description>
</ui-description>
<memory-description>
<available>
<memory-type-description>
<amount>42.25</amount>
<unit>MB</unit>
<usage-type>Storage</usage-type>
</memory-type-description>
<cpu-description>Intel(R) PXA263 400MHz
</cpu-description>
</hw-properties>
</fipa-device>
```

(A) Device Description

```
<user-interface>
<version-info>1.0.0</version-info>
<handlerclass>com.bt.teamworker.ui.ppc.jobqueue.
TeamJobListHandler</handler-class>
<file>
<name>twuicore.jxe</name>
</file>
<runtime>
<ui-description>
<screen-description>
<width>240</width>
<height>320</height>
<color>true</color>
<ui-description>
<runtime>
<gui-components>
<table>
<columns>
<name>To Do</name>
<id>201</id>
<name>When</name>
<id>202</id>
</columns>
<table>
</gui-components>
</user-interface>
```

(B) GUI Component Description

Fig. 3. Sample description of device and GUI component profiles



(a)

```
…
<Service name=MiniJobTrade>
<Type>CooperativeService</Type>
<AttachedTo>Work</AttachedTo>
<Appearance>MenuItem</Appearance>
<C-COM>com.bt.teamworker.services.
  mini_trade.initiation.MiniTrade_initiator
</C-COM>
</Service>
<ServiceLink name=SL1>
<Type>MenuPopUp</Type>
<PreService>ShowWorkDetail</PreService>
<PostService>
<Service>MiniJobTrade</Service>
<Service>MaxiJobTrade</Service>
<Service>UpdateWorkDetail</Service>
<Service>CloseWork</Service>
</PostService>
</ServiceLink>
…
```

(b)

Fig. 4. A part of a service tree (a) and its specification (b)

can be provided to a worker according to a human user's use of GUI components. A service description contains the name, type, appearance, target ontology (<AttachedTo> tag), and C-COM reference to implement a cooperative service. A Service Link describes the transition path from one service to another service. For example, in Fig. 4, a service link 'SL1' links a service 'ShowWorkDetail', with four succeeding services. A service link provides a personal agent with the information representing the context wherein a service is provided.

A personal agent configures the services for a mobile device according to the service tree and service specifications provided by a human user. A service is either generic or application specific depending on the generality of the service. Generic services can be provided to multiple applications while application specific services are applicable for a specific application. As mentioned in section 3.2.1, a service can also be classified as either a 'local' or 'coordination' service. A coordination service is delivered or created as a result of coordination with other worker's personal agents.

3.3 Flexible Management of Coordination Services

TeamWorker adopts CCoMaa [15], a conversational component based MAS architecture, as a means for the delivery of coordination services. In CCoMaa the interaction among agents for service delivery is performed via C-COM [15]. A C-COM is a conversational component that executes a service via asynchronous message based communication between two or more role components. A role component abstracts all the behaviour of an agent in the communication to achieve the service. The use of C-COM enables a personal agent to communicate with any service providing agents that use ad-hoc conversation policies for the provision of their service. The coordination logic required to facilitate the cooperative execution of a specific work item can be easily implemented using a conversation policy.

Fig. 5 shows the process for the dynamic registration and installation of a C-COM in CCoMaa. First, new services are implemented as a C-COM and their corresponding role components are registered to a service mediator agent by a service provider agent (or an administrative agent). In addition a service description (that contains the authorised role of target workers and related business rules), and a component description that specifies the required runtime environment (e.g. a JVM supporting the CDC specification) and required computing resources (e.g. 20k storage space, 160x160 screen resolution, etc) is also registered. Later, a service consumer agent (a personal agent on a mobile device) contacts the service mediator agent to find any available role components for new services by providing its device profile (as mentioned in section 3.2.2) which is used by the mediator agent to match appropriate role components. The matched role components are downloaded and installed on the device of the agent and can be used by a worker on demand. In this approach, a personal agent also contacts a mediator agent to find the latest version of C-COMs installed on the device. This version matching process enables efficient management of coordination services where the coordination logic changes frequently due to organisational changes or any other reasons.

3.4 Always on Awareness

In the following two sub-sections, we firstly discuss how the personal agent can keep the user informed when events that affect them occur, while the user is unable to interact with the system. Secondly, we discuss how the personal agent can take over responsibility (from the user) of maintaining real time tracking of the user's work stack.

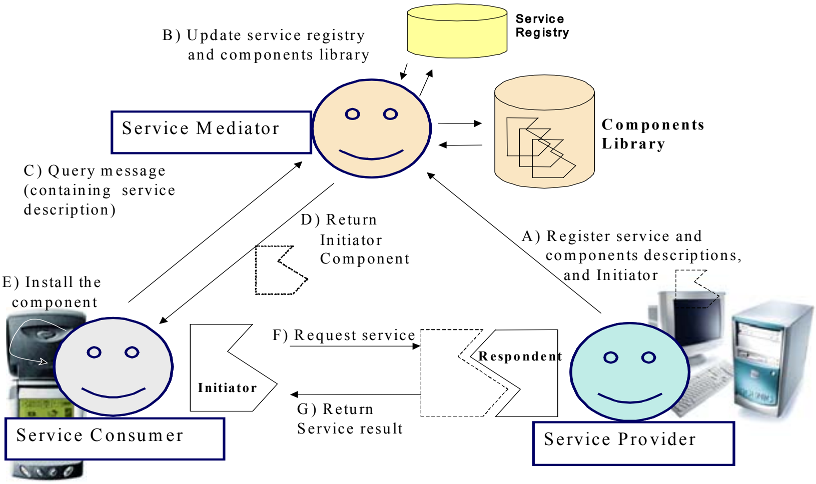

B) Update service registry and components library
Service Registry
Service Mediator
Components Library
C) Query message (containing service description)
D) Return Initiator Component
A) Register service and components descriptions, and Initiator
E) Install the component
F) Request service
Respondent
Initiator
G) Return Service result
Service Consumer
Service Provider

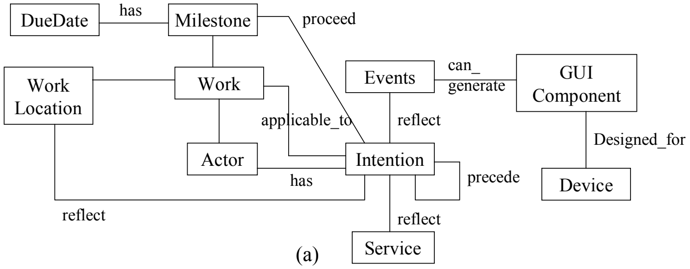
Fig. 5. Dynamic registration and download of coordination services in CCoMaa (cited and modified from [15])

3.4.1 Context Aware Multi-modal Human Agent Interface

Where appropriate, the interaction between a user and a personal agent can be based on both speech synthesis and voice recognition. TeamWorker utilises an internally developed speech engine [17] that contains both a text-to-speech (TTS) component and a voice recognition component. The TTS component is used to keep the user informed of incoming messages, and system alerts, when they are no longer visibly interacting with the application. Users are able to configure which messages/events they wish to be informed of. The TTS component provides a voice queue in which multiple entries can be placed by the personal agent, as well as a number of configuration options to manipulate the produced voice. The voice recognition component is used as a means of enabling user interaction without requiring the user to look or touch the device. Due to the limited processing capabilities of mobile computing devices, the personal agent only recognizes specific command words based on current contextual information in order to increase the matching accuracy. Expected commands are restricted to those related to both the GUI components currently showing, and any data constraints. For example, if the GUI is showing details of an activated job, the most likely recognized command words are 'update job information', 'request assistance', and 'close job', etc. Coupled with the 'always on' nature of TeamWorker, text and voice integration dramatically increases its usability for mobile workers by keeping them in touch with their colleagues in ‘hands free’ situations, such as driving, climbing, and using tools and equipment.

3.4.2 Autonomous Work Milestone Monitoring

Another feature of the personal agent with regards to improving reach-ability of mobile workers is its ability to assess the user’s work progress, by monitoring both their usage of the system, and their physical movements. This progress information can be used in various ways to improve the management of the user’s work stack. For example, a personal agent can transparently download relevant information for a job such as test results, and health and safety tips when it detects that the user is in transit to a job. To support this feature, each work item is described with one or more milestones. Each milestone is linked with some type of contextual information that is used by the personal agent when determining if the milestone has been reached. For example, contextual information includes the job state, location of the user, time, etc. A personal agent collects this contextual information several ways; by monitoring the use of services run during the execution of jobs, the location changes, the schedule of assigned jobs, etc. Fig. 6 (a) shows the ontology items and their relationships for this purpose. Fig. 6 (b), (c), and (d) show the sample rules used for an application specific milestone management. These rules can be classified as three types: rules for dispatching each event into a related user intention fact (b), rules for updating the progress of a milestone based on the new user intention if appropriate (c), and rules for determining appropriate actions to undertake when a desired milestone has been reached (d).



```
(defrule dispatch-event
(Event (Type
GUI_Tr)(Source
WorkDetailView)(WorkId
?wid))
=>
  (assert  (Intention (Type
WorkProgress)(WorkId
?wid)))
```

(b)

```
(defrule progress-milestone
(Intention (Type WorkProgress)
(WorkId ?wid))
(Schedule (PreWork
?wid)(PostWork ?pwid))
=>
(assert  (Milestone (Name
Closed)(WorkId ?wid)(State
Reached))
(exec SuggestRara ?pwid))
```

(c)

```
(defrule alarm-user
(Milestone (Name
Closed)(WorkId ?wid)(State
Reached))
(Schedule (PreWork
?wid)(PostWork ?pwid))
(UserConfirm Yes)
=>
(exec PerformRARA ?pwid)
```

(d)

Fig. 6. (a) Ontologies and their relationships for work milestone management (b)(c)(d) simplified sample rules for work milestone management

4 Application to a Real Mobile Business Process

TeamWorker has been applied to support a mobile business process within British Telecommunications plc. (BT). This section shows how TeamWorker improves the working practices of a BT Field Engineer.

4.1 Telecommunications Service Provision and Maintenance

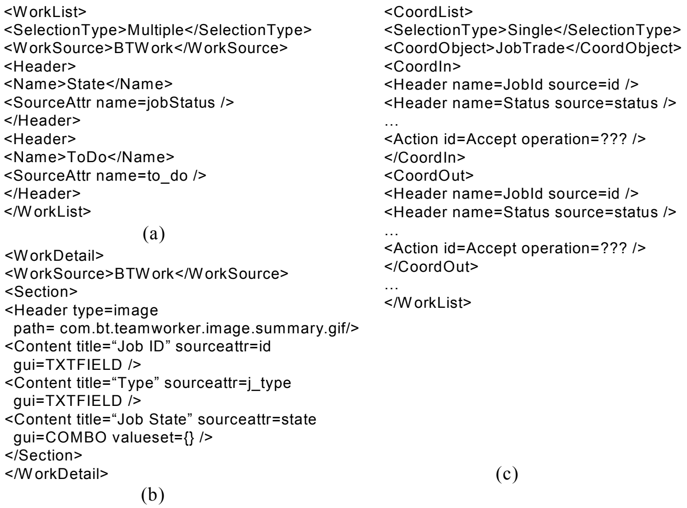
BT operates a large mobile workforce to maintain its network and provide telecommunications services to residential and business customers across the UK. The process starts when a customer or a BT Field Engineer (a mobile worker) requests provision of a new service or repair of an existing one. The execution of each job requires one or more BT Field Engineers to visit a customer site and/or BT Equipment sites. In a typical working day, a BT Field Engineer picks up jobs from a centralised workflow management system, called Work Manager [8]. Then, the engineer moves to a customer site to execute a job. BT is operating a customer satisfaction initiative, called RARA (Ring Ahead Ring After) which mandates that a BT Field Engineer must ring the next customer before they arrive to inform them that they are on their way, as well as after completion of a job to receive customer feedback. In the case of a repair job, the engineer may visit a place where network-equipment is located before visiting the customer to perform one or more tests to identify the cause of the problem.

4.2 TeamWorker for BT Field Engineers

TeamWorker has been customised to support BT Field Engineers by providing coordination services on the fly. Fig. 8 contains screenshots showing lists of BT jobs and their details based on the following specifications shown in Fig. 7. Fig. 7 shows specification examples for the configuration of GUI components for BT Field Engineers. Fig. 7 (a) is a part of the GUI specification for the configuration of the WorkList view that presents the list of assigned jobs to a BT Field Engineer. TeamWorker provides a tabular view for presentation of the job list. The <SelectionType> tag indicates that a BT Field Engineer can select multiple jobs when executing an operation (such as update job details). The <Header> tag is used to define each column of a job list table in the GUI to show WorkList information. Fig. 8 (a) and (b) show the WorkList View and the WorkDetail View configured by a personal agent based on the specification in Fig. 7 (a) and (b) respectively. The WorkList View shows the minimal information with regard to each job assigned to an Engineer while the WorkDetail View shows the details of the job. All the jobs assigned to a BT Field Engineer are delivered via a message-based interaction between a personal agent (running on the user's device) and a workflow agent (which operates inside of the BT Corporate Intranet) via C-COMs.

The capability of TeamWorker to adapt to heterogeneous devices is expected to be very useful to BT Field Engineers who normally will carry three types of devices: A company provided a Laptop Computer, a PDA, and their own mobile phone. The

Laptop Computer is expected to be equipped within a Van while an Engineer will carry a PDA and a mobile phone. The Laptop Computer will be used for accessing heavy weight services which requires a larger screen resolution and intensive keyboard inputs while a PDA and a mobile phone will be used for accessing light weight services such as colleague awareness, job list and details, and message or voice based communications based services. If a PDA runs out of battery power, an Engineer is able to continue working with the application, by switching to a mobile phone wherein the personal agent will reconfigure the GUI and service components dynamically by sacrificing some GUI components and services that are subject to performance and resolution etc.



```
<WorkList>
<SelectionType>Multiple</SelectionType>
<WorkSource>BTWork</WorkSource>
<Header>
<Name>State</Name>
<SourceAttr name=jobStatus />
</Header>
<Header>
<Name>ToDo</Name>
<SourceAttr name=to_do />
</Header>
</WorkList>
```

(a)

```
<WorkDetail>
<WorkSource>BTWork</WorkSource>
<Section>
<Header type=image
  path= com.bt.teamworker.image.summary.gif/>
<Content title="Job ID" sourceattr=id
  gui=TXTFIELD />
<Content title="Type" sourceattr=j_type
  gui=TXTFIELD />
<Content title="Job State" sourceattr=state
  gui=COMBO valueset={} />
</Section>
</WorkDetail>
```

(b)

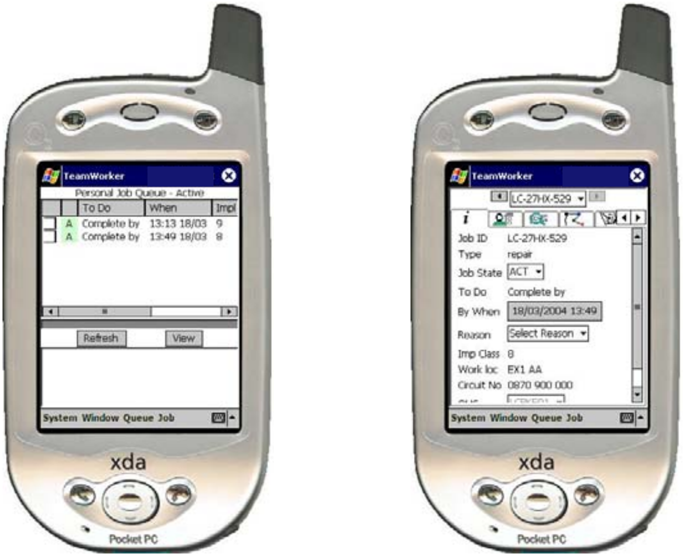
```
<CoordList>
<SelectionType>Single</SelectionType>
<CoordObject>JobTrade</CoordObject>
<CoordIn>
<Header name=JobId source=id />
<Header name=Status source=status />
…
<Action id=Accept operation=??? />
</CoordIn>
<CoordOut>
<Header name=JobId source=id />
<Header name=Status source=status />
…
<Action id=Accept operation=??? />
</CoordOut>
…
</WorkList>
```

(c)

Fig. 7. Sample specifications for the configuration of GUI components for BT Field Engineers

BT Field Engineers are classified into three categories according to the line of business they belong to: BT Retail, BT Wholesale, and BT Global Service. The cooperation requirements of the three groups are slightly different due to the diverse types of equipment for which the Field Engineers are responsible. The plug and play feature of coordination services via C-COM makes TeamWorker applicable to all three organisations. Each worker is able to configure a set of cooperation services depending on the organisation s/he belongs to and her/his role. With this functionality, BT can manage all the services in a centralised way and a cooperation service for an organisation can be customised for another organisation and deployed dynamically.

However, so far three types of common coordination services have been identified that are useful for BT Field Engineers across all the three organisations. Firstly, two types of job trading services, mini trade and maxi trade, have been implemented to enable each Engineer to re-assign their jobs via peer-to-peer negotiations with other Engineers. Mini trade is an agent-based job trading service wherein a personal agent contacts other users' personal agents to determine the most appropriate candidates for the reassignment of a job. For this, a contract net protocol has been used to implement the coordination logic. Each respondent personal agent creates a Bid, which contains the distance between its current location and the job location, the skill set of the mobile worker the personal agent is assisting. The initiator personal agent compares the Bids from the responder personal agents and proposes the job to the highest-ranking respondents in order.



TeamWorker

Personal Job Queue - Active

		To Do	When	Impl
□	A	Complete by	13:13 18/03	9
□	A	Complete by	13:49 18/03	8

TeamWorker

LC-27HX-529

Job ID LC-27HX-529

Type repair

Job State ACT

To Do Complete by

By When 18/03/2004 13:49

Reason Select Reason

Imp Class 8

Work loc EX1 AA

Circuit No 0870 900 000

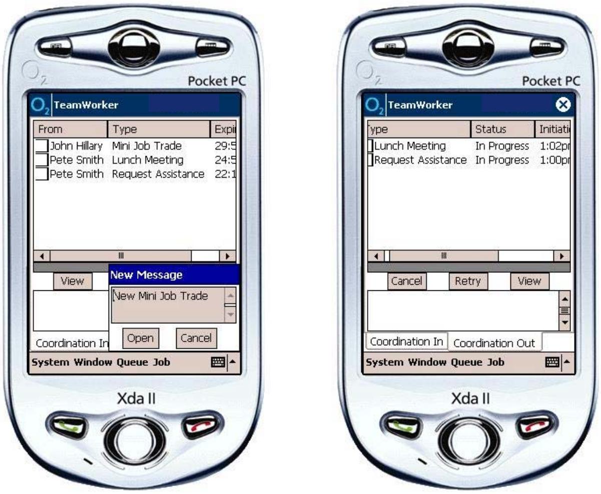
(a) (b)

Fig. 8. TeamWorker screenshots for Work list (a) and the detail of a BT work (b)

In a Maxi job trade service, a personal agent contacts the personal agent of a team leader to ask for reassignment of a job. Secondly, a request assistance service is provided to each Engineer who wants to form a temporary group to execute a collaborative job. In this service, an initiator personal agent contacts one or more respondent personal agents to find candidates, based on their user's schedules, locations, and skills. Thirdly, an informal coordination service, 'organise lunch meeting', is provided to each Engineer. In this service, an Engineer can propose a lunch meeting by specifying candidate restaurants, times, and buddies. Each respondent can specify a preference for the venue and time, which is monitored by other colleagues. The informal meeting service is held to be an important element in improving a team's performance, because Field Engineers can share their knowledge during informal social interaction [23]. Fig. 9 contains screenshots that show the incoming (a) and outgoing (b) coordination services of a Field Engineer. Especially, Fig. 9 (a) shows a message box indicating an arrival of new coordination service from another Engineer. The message can be read by a personal agent via the TTS engine.

The speech based interaction capability along with the work milestone management capability of a personal agent has been applied to implement a BT policy, RARA. With the support from a personal agent on each device, the ratio of failure of RARA is expected dramatically reduced, which will increase the customer satisfaction.

Finally, the "always on" awareness between team members is expected to improve the effectiveness of team working by increased intra-team communications, which will facilitate the knowledge sharing between a novice Engineer and expert Engineers.



From	Type	Expi
John Hillary	Mini Job Trade	29:5
Pete Smith	Lunch Meeting	24:5
Pete Smith	Request Assistance	22:1

New Mini Job Trade

ype	Status	Initiati
Lunch Meeting	In Progress	1:02pı
Request Assistance	In Progress	1:00pı

(a) (b)

Fig. 9. Screens showing outgoing (a) and incoming (b) coordination services

5 Conclusion

This paper proposed TeamWorker as a CSCW system for mobile workforce. The Agent technology plays a central role in TeamWorker to assist a mobile worker by overcoming the constraints faced within a mobile computing environment. The main features of TeamWorker include the flexible composition of services for execution of work via a component-based approach, the ability to adapt to diverse devices, multi-modal interaction between agents and users, and the work milestone management capability of a personal agent. The benefits of TeamWorker are improved cooperation among mobile workers, increasing the reachability of remote workers by supporting multi-modal interactions between them and their personal agents, and better workforce management via transparent work progress monitoring.

During the development of TeamWorker, multi-agent technology showed the following advantages for the implementation of mobile CSCW. First, from the group awareness perspective, the autonomy of an agent is a useful feature to free a mobile worker from continuously reporting his location and job execution progress (information which is vital for collaboration with colleagues). Second, the message-based interaction protocols for multiple agents are useful for implementing asynchronous coordination of mobile workers. The use of the C-COM service architecture enables workers to use a new service on demand, as an agent can dynamically locate and install the corresponding C-COM for the service, with no a priori knowledge of any novel protocols that may be employed by that service. Third, agent 'intelligence' is a useful feature which, when used to understand the current job execution context, can support an enhanced the user experience via pro-active services such as finding the nearest qualified colleagues for the collaborative execution of the next job, switching to the appropriate interface mode (speech or text based) for human computer interaction, and so on.

TeamWorker will be trialled in the near future with two teams of circa twenty mobile workers. The metrics for the evaluation of the field trial as are:

- Re-work Ratio: Re-work normally occurs when engineers cannot complete their jobs due to lack of required knowledge or parts. The coordination support from TeamWorker is expected to enable engineers to instantly contact the person best able to help them with their problem.
- RARA Outage Ratio: The intelligence and autonomy of the personal agents in TeamWorker is expected to reduce the RARA outage ratio dramatically by giving reminders to engineers when appropriate, or by directly calling the relevant customer.
- Customer Promise Hit Ratio: The real-time job trading functionality of TeamWorker is expected to increase this ratio by allowing engineers to reassign their jobs in exceptional cases.

References

1. Adorni, G., Bergenti, F., Poggi, A., and Rimassa, G.: Enabling FIPA agents on small devices . Lecture Notes in Artificial Intelligence 2182 (2001), 248–257
2. Berger, M., Buckland, B., Bouzid, M., Lee, H., Lhuillier, N., Olpp, D., Picault, J., and Shepherdson, J., An Approach to Agent-based Service Composition and it's Application to Mobile Business Processes. IEEE Transactions on Mobile Computing, 2(3) (2003) 197-206
3. Bose, R.: Intelligent Agent Framework for development knowledge-based decision support system for collaborative organisational processes. Expert Systems with Applications 11(3) (1996) 247-261
4. Chang, J.W. and Scott, C.T.: Agent-based workflow: TRP support environment (TSE). Computer Networks and ISDN Systems 28 (1996) 1501-1511
5. Cao, J. and Zhang, S.: An Integrated Multi-agent CSCW System for Concurrent Product Development. Int. Jr. of Information Technology & Decision Making 1(3) (2002), 423-440
6. Domingos, H., Martins, J.L., Preguiça, N., and Duarte, S.M.: A Workflow Architecture to Manage Mobile Collaborative Work. in Proc.s of EPCM'99 (1999)
7. Ellis, C., Wainer, J., Barthelmess, P.: Agent-Augmented Meetings In: Agent supported cooperative work. Ye, Y. and Churchill E. (ed.). Kluwer Academic Publishers (2003)

8. Fagrell, H., Ljungberg, F., and Bergquist, M.: Exploring Support for Knowledge Management in Mobile Work. In Proceedings of the European Conference on CSCW (1999)
9. FIPA Device Ontology Specification, http://www.fipa.org/specs/fipa00091/ (2000)
10. Garwood, G.J.: Work Manager. BT Technology Journal 14 (3) (1996)
11. Georgakopoulos, D., Hornick, M.F., and Sheth, A.P.: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases 3 (2) (1995) 119-153
12. Greenberg, S. (ed.): Computer-supported Cooperative Work and Groupware. Computers and People Series, Academic Press Ltd. (1991)
13. Jennings, N. R., Norman, T. J., Faratin, P., O'Brien P., and Odgers, B.: Autonomous agents for business process management. Int. Jour. of Applied AI 14(2) (2000) 145—189
14. Khoshafian, S. and Bucklewicz, M.: Introduction to Groupware, Workflow, and Workgroup computing. John Wiley & Sons Inc. (1995)
15. Lee, H., Mihailescu, P. and Shepherdson, J.W.: Conversational Component-based Open Multi-agent Architecture for Flexible Information Trade. Lecture Notes in Artificial Intelligence (LNAI) 2782 (2003) 109-116
16. Nunamaker, J.F., Dennis, A.R., Valacich, J.S., Vogel, D.R., George, J.F.: Electronic Meeting Systems to Support Group Work. Communications of the ACM 34(7) (1991) 40-61
17. Page, J.H., and Breen, A. P.: The Laureate Text-to-Speech system: Architecture and Applications. BT Technology Journal 14(1) (1996) 84-99
18. Papazoglou, M.P. and Schlageter, G.: Cooperative Information Systems – Trends and Directions. Academic Press, ISBN 0125449100 (1997)
19. Schmidt, A., Beigl, M. and Gellersen, H. W.: There is more to Context than Location. In Computers & Graphics Journal. 23(6) (1999) 893-902
20. Schmidt, A., Lauff, M., and Beigl, M.,: Handheld CSCW, In Proceeding of Workshop on Handheld CSCW at CSCW '98, Seattle (http://www.teco.edu/hcscw/papers.html) (1998)
21. Sheetz, S.D., Tegarden, D.P., Kozar, K.A., and Zigurs, I.: Group Support Systems Approach to Cognitive Mapping. Jr. of Management Information Systems 11(1) (1994) 31-58
22. Wiberg, M.: Knowledge Management in Mobile CSCW: Evaluation Results of a Mobile Physical/Virtual Meeting Support System. In Proceeding of 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 4, Maui Hawaii (2001)
23. Wittaker, S., Frolich, D. and Daly-Jones, W.: Informal workplace communication: What is it Like and How Might we Support It?, in Proceedings of Human Factors in Computing Systems, CHI '94, ACM Press: New York (1994)
24. Ye, Y. and Churchill, E.: Agent Supported Cooperative Work. Kluwer Publishers, Boston Hardbound ISBN 1-4020-7404-2 (2003)