Goal-Directed Automated Negotiation for Supporting Mobile User Coordination

Iyad Rahwan[1,⋆], Fernando Koch[2], Connor Graham[3], Anton Kattan[3], and Liz Sonenberg[3]

[1] Institute of Informatics, The British University in Dubai,P.O.Box 502216, Dubai, UAE
iyad.rahwan@buid.ac.ae

[2] Institute of Information and Computing Sciences,
Utrecht University, Utrecht, The Netherlands
fkoch@acm.org

[3] Department of Information Systems, University of Melbourne,Parkville, VIC 3010, Australia
{cgraham, kattan, l.sonenberg}@unimelb.edu.au

Abstract. While interacting with other users in dynamic use contexts, one often aims at coordinating activities as events unfold. Such coordination can often be *unplanned* or *impromptu*. There are opportunities for supporting impromptu coordination among mobile individuals by representing and processing contextual information. In this paper we present a novel technique, based on goal-oriented automated negotiation, to enable computational agents acting on behalf of users to automatically negotiate opportunities for coordination. Our focus is on the technology 'under the skin' that can represent, analyse, and integrate information to support the user's tasks in a timely and appropriate way. An implemented prototype is demonstrated via a scenario, which is based on a workday narrative.

1 Introduction

The use of mobile and handheld computing devices in everyday life is increasing largely due to the advancement of enabling technologies [20] and increasing efforts to improve usability [21]. In order to cope with the dynamism resulting from user mobility during task execution, providing *context-sensitive* support becomes crucial. Dey [6] describes *context* as "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 the application themselves." For the purpose at hand, context-sensitive support includes performing background computation to identify the situation, in order to take action on behalf of the user, and to represent appropriately to the user the state of any such external interaction.

In this paper, we are concerned with forms of interaction in which multiple users, with differing agendas and interests, may realise opportunities for useful coordination of

⋆ This work was initiated while Iyad Rahwan was at the University of Melbourne. Iyad was supported by a University of Melbourne Research Scholarship. We also thank Hewlett Packard's Philanthropy Division for donating equipment.

A. Dey et al. (Eds.): CONTEXT 2005, LNAI 3554, pp. 382–395, 2005.

their activities. This interaction occurs in a dynamic environment and hence automated support for this interaction has to be context-sensitive, taking into account contextual variables such as location, time, and the task(s) at hand. As a result, mobile coordination is often *unplanned*, and therefore takes place *impromptu* as users happen to be in a context where coordination opportunities arise. We are interested in supporting this type of interaction and, therefore, not concerned with cooperation in a classic "groupware" sense. The proposed technology discussed here is not targeted at supporting "groups of people engaged in a common task (or goal)" [7–page 40] or providing "an interface in a shared environment" (ibid). We are concerned with communication among multiple entities and coordination, in a non-team-driven sense. We use the term *coordination* to capture the kinds of activity we have in our focus, adopting Malone and Crowston's definition of coordination as "managing dependencies between activities" [13].

We explore scenarios where the user is engaged in goal-oriented task execution where they could exploit impromptu coordination with others with overlapping task structures. Hence, we consider the user's *task structure* as part of the user's context. We present a novel framework (which we sketched earlier in [17]) for reconciling task structures of different users using *interest-based negotiation* (IBN) [16], a form of automated negotiation that exploits an explicit representation of task structures. We also present a prototype system based on the framework.

The paper advances the state of the art in two ways. To our knowledge, it is the first attempt at using automated negotiation to support non-routine coordination of mobile users. Moreover, the paper introduces a novel coordination architecture, that integrates context-aware networked devices, agent-based reasoning, and automated negotiation. This approach may be used for building a variety of mobile coordination-support systems that suit domains beyond that of the simple narrative used here for illustration.

We begin by discussing the characteristics of mobile coordination in the next section. This helps clarify the features required by potential supporting technology. In Section 3 we present a framework for automated coordination support that provides some of these features, followed by a description of the implemented prototype in Section 3.4. We discuss related work in Section 4 and conclude in Section 5.

2 Problem: Impromptu Coordination While Mobile

We begin with some observations regarding the role of technology in facilitating interactions through which multiple users, with differing agendas and interests, may realise their opportunities for useful coordination of their activities. To better understand the opportunities for technology intervention, we "examine activities in which people engage with others when they are 'mobile' and how various tools and artifacts feature in those activities" [12]. To this end, we analyse a 'Day-in-the-life' scenario [19] to distill essential characteristics of mobile use.

Scenario: Supporting Impromptu Coordination

In the settings of interest, the user is mobile, connected, and engaged in complex interactions. This creates an opportunity for technology to support the user. A basic approach would be to provide connectivity, e.g. using mobile telephones. In this case, users would need to keep track of all changes to their context "in their heads," manage the complexity of identifying opportunities as events unfold, deal with multiple interaction partners, and so on. This places great cognitive load on mobile users, and it is precisely for this reason that support software such as calendar applications are appropriate tools.

When a mobile phone is endowed with a calendar functionality, the user can *outsource* the storage of large amounts of information about activities (meetings, special occasions etc.) to his/her device. This representation of *individual* activities can then be used to help a user coordinate with others. Applications allowing for group task representation go a step further by providing users with global activity representations.

One could envisage device support not only through *representation* of individual and group activities, but also *automation* to support the cognitive processes that exploit and manipulate those representations. Such automatic processes would use the available information about the user's context as well as information available about other users in order to automatically negotiate agreements over collaboration and coordination of activities. We now turn to look in more detail at types of situations where such support could be advantageous - with a view to illustrating the settings in which negotiation can effectively facilitate impromptu coordination. "Mobility" [15] naturally creates such opportunities as we shall see. The analysis below emerged from discussions in a multidisciplinary focus group and from a narrative based on a diary of an actual PhD student renamed Fred, generated over a period of three days. The narrative approach has been used in order to understand individual mobile activities in other projects, such as ActiveCampus [8]. An approach grounded in broader and more systematic data collection would be desirable in the future, c.f. [10].

> I realized I had not set up a lift home so I called my wife. I couldn't get through, so I left her a message and asked her to call me when she was close. While waiting for her to reply, I continued work. Then Jack gave me a call to discuss our Wednesday meeting. Jack asked if I could get him a book from the university library, which he needs for an assignment. I declined because I needed more time to finish my work. But Jack happened to be planning to head home to study at the same time I wanted to leave the University. I managed to get myself a lift home by offering to help him out with his assignment, in which case he no longer needed the book.

From the analysis of this scenario, four distinctive characteristics of mobile work emerged. We argue that "negotiation" has the potential to address these characteristics.

Fluidity. Kakihara and Sorenson [11] describe how interaction experienced by mobile individuals is "fluid". They describe how "human interaction is becoming ambiguous and transitory. The patterns of social interaction are dynamically reshaped and renegotiated through our everyday activities significantly freed from spatial, temporal and contextual constraints" [ibid]. Fluidity in mobility suggests that interaction can be occasional in mobile use situations, since the context in which these portable devices operate changes more frequently than with stationary computers. Thus, well-established, long-term relationships, in which task structures are well-defined and agreed upon, are less likely (e.g. due to the dynamism of resource availability). Negotiation is one way to reach temporary agreement in such dynamic settings, as Fred did in the above scenario.

Impromptu Coordination. For mobile users, opportunities for collaboration arise more frequently than with static users due to the more diverse forms of context change, such as change in the user's location or the proximity of multiple users. Such opportunities usually cannot be anticipated a priori. In the narrative above, Fred being connected to Jack was critical to him being able to capitalise on the opportunity presented by Jack's proximity. The phone did not allow him to predict the possible chances of the success of this opportunistic interaction through a representation of Jack's goals or tasks. Negotiation is a way of dynamically realising and taking advantage of such opportunities.

Heterogeneity. When the modelling of context is to take into account varying location, time, user profiles, tasks, interaction history etc., we are confronted with a much greater variety of agent (and user) types. Each individual agent may achieve tasks in a different way. It is unlikely that information about this heterogeneity will be available a priori. Negotiation is a natural way to exchange information and reach useful agreement or compromise with collaborators (or in collaboration settings) not known before. In the above narrative, Fred's coordination could have been made easier by accessing a representation of Jack's activities or, at the very least, his availability.

Privacy and Connectivity. Mobile users are constantly confronted with different interaction partners that want to obtain information about them. Users may be unwilling to disclose all the information required to run a centralised algorithm for coordinating joint activity. They may be willing to do so only when interacting with particular partners, or when the they realise the potential benefit of exchanging such information. Negotiation is a natural way to reconcile one's own wish to protect private information with the potential benefit of interacting with others.

3 Solution: Negotiating Impromptu Coordination

In the previous section, we argued that impromptu mobile coordination requires the ability to represent information about the tasks of different users, and the ability to interactively process this information. We now present a framework, based on goal-directed automated negotiation, that addresses some of these requirements.

3.1 Conceptual Model

The conceptual framework for mobile user coordination through automated negotiation is illustrated in Figure 1. An agent running on a user's mobile device acts as an intermediary between the user and other potential collaborators. The agent gathers contextual information from the environment (e.g., lecture times, location of user and colleagues) and from the user (e.g., availability, goals). The agent then uses this information, as well as domain-specific knowledge (e.g., procedures for borrowing books from the library) in order to negotiate with agents representing other users. Negotiations are motivated by the user's goals and aim at achieving "deals" with other users. If negotiation results in potentially useful deals (e.g., appointment, lunch, lift home), these are proposed to the respective users, who might accept, reject, or modify these deals as they see suitable.

Note that agents may have incomplete information about each others' plans and desires, and about the environment and the appropriate planning procedures within it. In

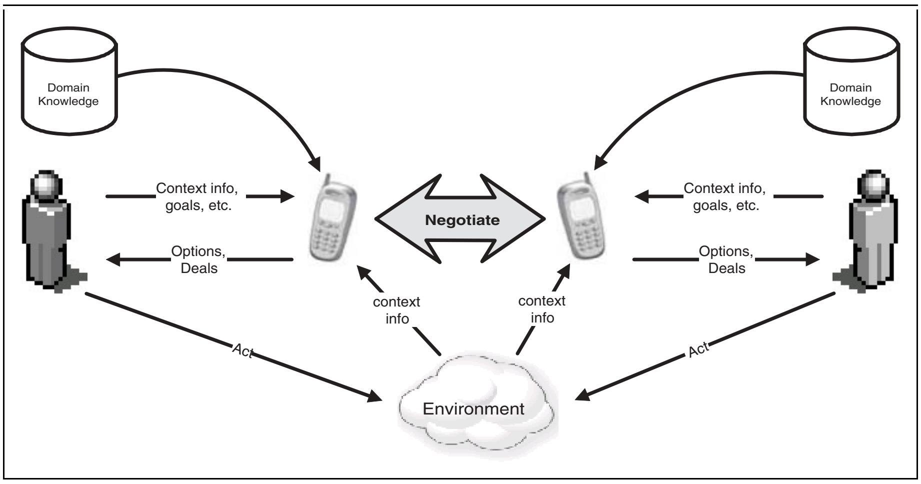


Fig. 1. Conceptual framework for automatically negotiated mobile coordination

order to address this issue, the automated negotiation framework must enable agents to exchange information about these underlying notions. Since the negotiation mechanism required must exploit representations of users' tasks and goals, interest-based negotiation seems an appropriate choice.

3.2 The Architecture

At it's core, our solution is based on an extended implementation of *3APL* [9]. 3APL is a logic-based agent programming language which provides constructs for implementing agents' beliefs, goals and capabilities as explicit run-time entities. It uses practical reasoning rules in order to generate plans for achieving agents' goals and for updating or revising these plans. Each 3APL program is executed by means of an interpreter that deliberates on the cognitive attitudes of that agent. Due to space limitation, we shall not discuss 3APL in detail, and suffice by introducing the general idea behind it.

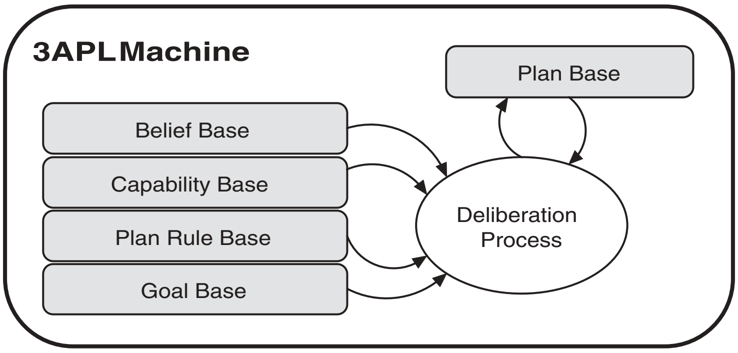


Fig. 2. 3APL Architecture

Figure 2 describes the abstract architecture of 3APL. Each computational agent has explicit representations of its goals in the *goal base*, which contains goals in the form of ground predicates. For example, the goal to finish an assignment may be represented with the predicate $\mathit{finish}(\mathit{assignment})$. In order to achieve its goals, the agent decomposes these into less abstract *sub-goals* using planning rules from the *plan rule base*, which may themselves be decomposable into other sub-sub-goals, until concrete *actions* are reached (i.e., physical actions agents may execute directly in the world). This results in a hierarchical structure in which the top-level root nodes represent desires, intermediate nodes represent abstract goals, and leaf nodes represent concrete actions to be executed. During plan generation, the agent takes into account its *belief base*. Beliefs take the form of predicates and represent contextual information the agent may use in its reasoning. For example, the predicate $\mathit{location}(\mathit{jack}, \mathit{bldgA})$ could denote that the agent believes Jack is currently located in building A. In a mobile coordination scenario, beliefs may be acquired by sensing the environment (e.g. using a location system) or directly through user input. The *capability base* describes possible actions by the agent and user. A planning rule takes the form $\mathit{head} \leftarrow \mathit{guard}|\mathit{body}$, and means that if the agent has goal g that unifies to the head of the plan *head* and the condition declared in *guard* is satisfied (i.e., it unifies to the contents of the belief base), then goal g can be achieved by executing the sequence of actions (or set of sub-goals) listed in *body*. Following is an example planning rule which states that if one wants to finish an assignment, and is currently outside, then one needs to collect the relevant book and go home to study:

$$\mathit{finish}(\mathit{assignment}) \leftarrow \mathit{location}(\mathit{me, out}) \mid \mathit{collect}(\mathit{book}), \mathit{go}(home)$$

where $\mathit{go}(home)$ may have to be decomposed to further sub-goals, such as catching a taxi or taking the train.

In the initial 3APL version presented by Hindriks et al [9], the deliberation cycle is fixed and agents generate their plans by choosing the first applicable rule that matches a particular goal/desire. This means that an agent generates only one plan for each goal, and only generates other plans if the initial plan fails. This deliberation cycle is depicted in Figure 3.

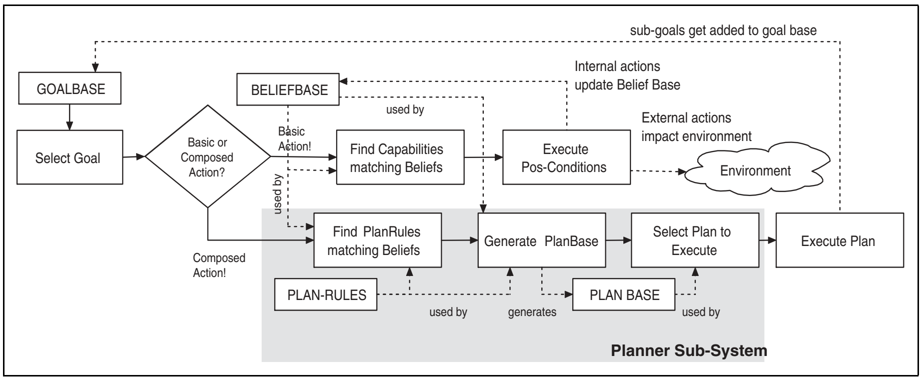


Fig. 3. Basic Deliberation Cycle

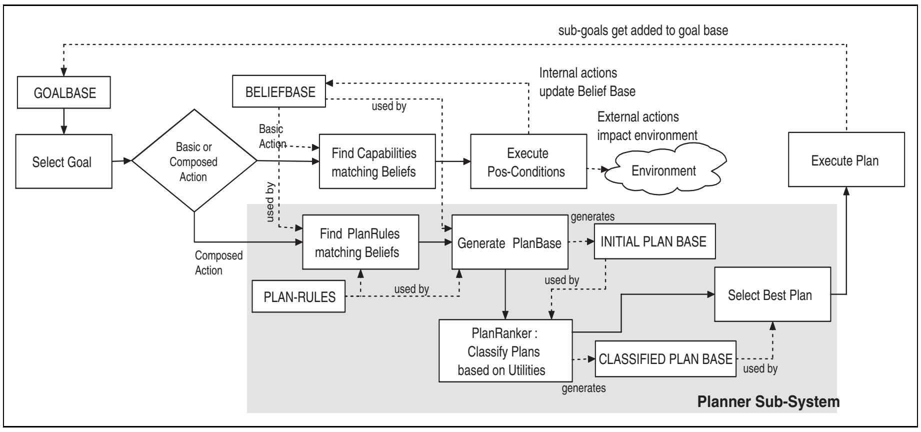


Fig. 4. Extended Deliberation Cycle

We want our agents to generate all possible plans and select the best plan using some appropriate criteria (perhaps based on user preferences), which is performed by the *plan ranker*. Currently, the plan ranker works by calculating the *utility* of different plans based on the *costs* of executing the plan and the *worth* of the goals achieved. This process results the *plan base*, a list of plans generated by the deliberation process. This list is available during the processing phase and can be manipulated by an external logic (supplied by the developer). This allows the creation of external Plan Analysers, that can be use to improve the deliberation cycle (for example, classify plans per utility). This *Deliberative 3APL* implementation is based on an abstract modified reasoning cycle presented by Dastani et al [5]. The deliberation cycle is described in Figure 4.

Each user in our framework has a 3APL agent running on his/her device. The agent has a representation of the user's goals, a set of predefined capabilities, and a set of beliefs that represent the current context. We assume this knowledge is captured through a separate mechanism (see Section 5 for more discussion). The agent must be capable of receiving information from the user and environment, and taking action externally (e.g. activating a reminder alarm or requesting information from the user). This is catered for by adding sensor and actuator interfaces that allow the integration of the 3APL machinery to the external world. In addition, agents need a mechanism by which they can communicate with one another and negotiate opportunities for coordinated action. For that purpose, we use our recently proposed *interest-based negotiation* (IBN) framework [16]. Negotiating agents exchange *proposals*, which are suggested "exchanges of favours" or actions. If a proposal made by one agent is accepted by another agent, the proposal becomes a *deal*.

The idea behind IBN is that agents can also exchange information about their underlying goal structures and use this information in order to discover better deals. This communication takes place according to a protocol, which is discussed in detail in [16]. This extended architecture is described in figure 5. Basically, the IBN protocol provides a way for agents to exchange information about their own goal structures, and to exchange new planning rules and beliefs. This enables agents to discover opportunities

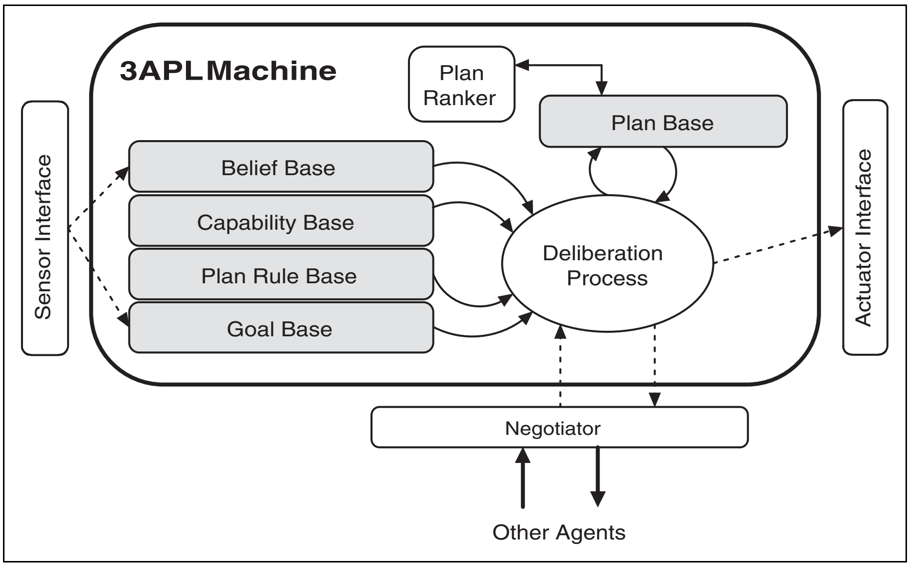


Fig. 5. 3APL-M and IBN

for reconciling their individual plans for mutual benefit. The way this works will be clarified in the following subsection.

3.3 Illustrative Example

Let us revisit the narrative from section 2. Recall the situation where Fred fails to get in contact with his wife to secure a lift home. Suppose Fred has a permanent desire to save money whenever possible, and this is stored in his device in the form of a desire generation rule for which the rule body is always true. One way of saving money is by getting a lift home rather than taking the train or a taxi. This is represented on the device as a planning rule. The two rules are encoded as follows, where $\mathcal{G}^i$ denotes the goals of user i and $\mathcal{P}^i$ denotes the planning rules i's agent is aware of.

$$\mathcal{G}^{Fred} = \{save(fred))\}$$
$$\mathcal{P}^{Fred} = \{save(Y) \leftarrow \mathsf{true} \mid GetLift(X,Y)\}$$

Recall also that Jack would like to finish his assignment, and in order to achieve that, he believe he needs to collect a book and go home.[1]

$$\mathcal{G}^{Jack} = \{finish(assignment)\}$$
$$\mathcal{P}^{Jack} = \{finish(assignment) \leftarrow \mathsf{true} \mid Collect(book), Go(home)\}$$

[1] Note that we do not have an explicit representation of time and temporal constraints. This is mainly to simplify implementation. A realistic implementation would of course need to account for temporal aspects, integrate tasks in a proper calendar application etc.

Now, suppose Jack is unable to collect the book himself (say because he has other tasks to perform) or that he simply prefers someone else to do it for him. Jack's device, which is equipped with interest-based negotiation ability, could automatically attempt to find alternative ways to get a lift home by searching for nearby friends and checking (with their devices) for potential collaboration. When Fred's and Jack's devices detect one another, they initiate a negotiation process.

As soon as Jack's device detects that Fred is in a nearby area, it checks whether Fred is willing to collect the book for Jack. Upon inspection of the request, Fred's device discovers there is not enough time to go to the library to pick the book, say because he has to do some work, as part of another task; i.e., that there is some form of conflict between the two actions. Therefore, Fred's device would reject Jack's request. However, Fred's device could attempt to find out the reason behind the request, with the objective of exploring other potential solutions. After finding that Jack wants the book in order to finish the assignment, Fred's agent could propose an alternative way to achieve the objective, by offering Fred's help. This is not for free, though, as it is in exchange for getting a lift home.

The following is the dialogue sequence just described, encoded using the IBN protocol presented in [16], between Fred's and Jack's negotiation-enabled mobile devices. Locutions PROPOSE$(.)$, ACCEPT$(.)$ and REJECT$(.)$ allows agents to propose, accept and reject deals, respectively. The locution ASSERT$(.)$ allows an agent to make assertions about its own beliefs, goals, sub-goals or planning rules. Finally, locution REQ-PURPOSE$(.)$ allows an agent to ask another for the higher-level purpose of a particular request (e.g. "why do you need the book?"). The special predicate prule$(.)$ is used to exchange planning rules among agents. Another modality instr(X,Y) is used by one agent to indicate that goals X are adopted because they are *instrumental* towards achieving higher-level goals Y. Part (a) in Figure 6 shows a sketch of (parts of) the plan structures for Fred and Jack from the narrative. Part (b) shows Jack's modified plan, which can achieve his goals while also helping Fred.

JACK: PROPOSE$(\textit{jack}, \textit{fred}, \mathbf{do}(\textit{fred}, \textit{Collect}(\textit{book})))$
FRED: REJECT$(\textit{fred}, \textit{jack}, \mathbf{do}(\textit{fred}, \textit{Collect}(\textit{book})))$

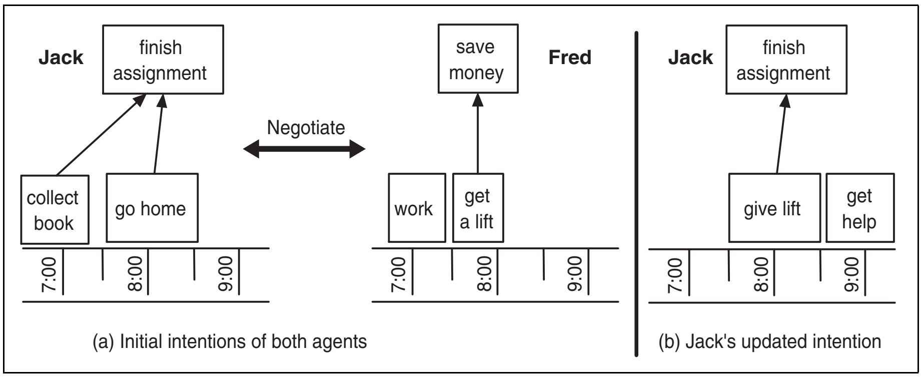


Fig. 6. An abstract view of negotiation

JACK: $\mathsf{PASS}(\mathit{jack})$
FRED: $\mathsf{REQ\text{-}PURPOSE}(\mathit{fred}, \mathit{jack}, \mathit{Collect}(\mathit{book}))$
JACK: $\mathsf{ASSERT}(\mathit{jack}, \mathit{fred}, \mathsf{instr}(\mathit{Collect}(\mathit{book}), \mathit{finish}(\mathit{assignment})))$
FRED: $\mathsf{ASSERT}(\mathit{fred}, \mathit{jack}, \mathsf{prule}(\mathit{finish}(\mathit{assignment}) \leftarrow$
$\mathsf{true} \mid \mathit{Help}(\mathit{assignment}), \mathit{GiveLift}(\mathit{jack}, \mathit{fred}))$
JACK: $\mathsf{PASS}(\mathit{jack})$
FRED: $\mathsf{PROPOSE}(\mathit{fred}, \mathit{jack}, \mathbf{do}(\mathit{jack}, \mathit{GiveLift}(\mathit{jack}, \mathit{fred})),$
$\mathbf{do}(\mathit{fred}, \mathit{Help}(\mathit{assignment})))$
JACK: $\mathsf{ACCEPT}(\mathit{jack}, \mathit{fred}, \mathbf{do}(\mathit{jack}, \mathit{GiveLift}(\mathit{jack}, \mathit{fred})),$
$\mathbf{do}(\mathit{fred}, \mathit{Help}(\mathit{assignment})))$
FRED: $\mathsf{OK}(\mathit{fred}, \mathit{jack}, \mathbf{do}(\mathit{jack}, \mathit{GiveLift}(\mathit{jack}, \mathit{fred})),$
$\mathbf{do}(\mathit{fred}, \mathit{Help}(\mathit{assignment})))$

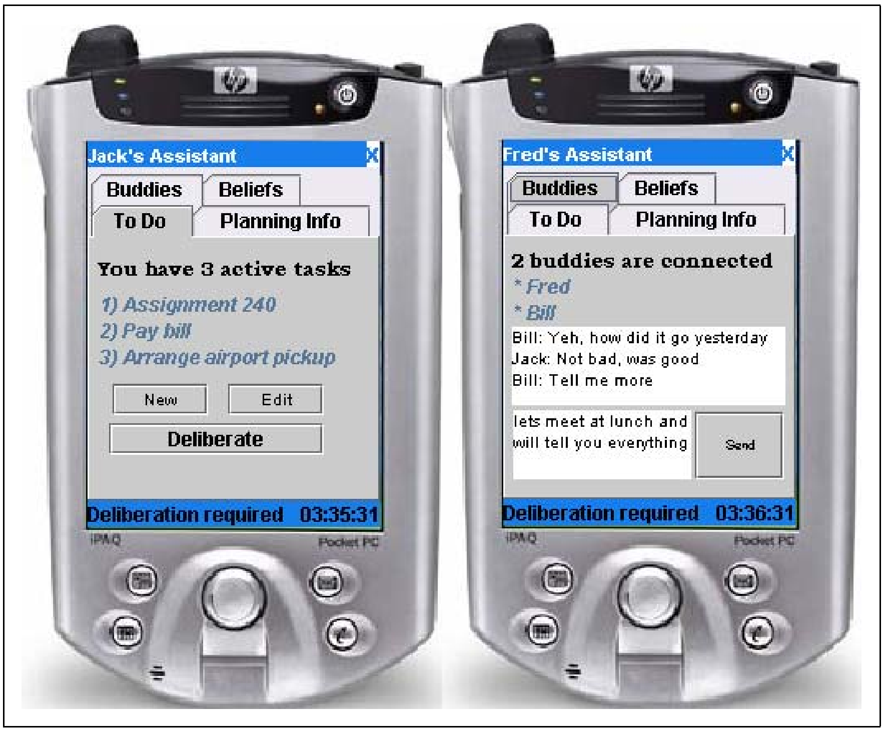
There are other types of arguments that Fred and Jack could exchange. For example, Jack could ask Fred why he needs the lift, and after finding out that is is purely for saving, he could offer to take him out for dinner instead, in exchange for collecting the book. Not all such arguments may be realistic. A sensible approach might be to encode a set of typical "attitudes" towards cooperation (represented as negotiation strategies), from which the user could choose for his/her agent, or which could be automatically learned by observing the user's response over a period of time.

3.4 Results

We have implemented a prototype system as a proof-of-concept for our framework. The prototype uses our implementation of *3APL-M* [1] using Java Micro Edition. The deliberation part of 3APL-M makes use of *mProlog*, a scaled-down Prolog engine. It is a sub-product of this project and packaged with the 3APL-M application library. Sensors and Actuators are developed in Java and attached to the 3APL-M machinery through special methods in the API. Figure 7 shows sample screen shots of the prototype. On the left is the interface of Jack's mobile device, showing the list of active tasks. The user can add new tasks or edit existing tasks. The right hand side of the figure shows Fred's mobile device, with an instant messaging dialogue with another user named "Bill."

The "Deliberate" button on Jack's device triggers the deliberation process, which is described in figure 8. In this figure, the normal arrows denote messages that immediately follow an action by the user. For example, after Fred clicks "Yes" on the top-right screen, a $\mathsf{REJECT}(.)$ message is sent. Dotted lines, on the other hand, denote messages exchanged by the agents without an explicit user action. For example, the $\mathsf{REQ\text{-}PURPOSE}(.)$ message from Fred's agent to Jack's agent did not require a confirmation or explicit request by the user, but is rather triggered automatically as part of Fred's agent's built-in coordination strategy.

In this particular implementation, Jack's agent selects Jack's task to finish the assignment, and by applying the planning rule mentioned above, creates a plan that involves collecting a book and then going home. After searching for nearby friends, Jack's agent discovers Fred's presence and suggests to Jack to ask Fred to collect the book on his behalf. If Jack clicks on "Yes" (part (a) in figure 8), his agent initiates a negotiation dialogue with Fred's agent by proposing that Fred pick the book. Fred's device



You have 3 active tasks

1) Assignment 240
2) Pay bill
3) Arrange airport pickup

Deliberation required 03:35:31

2 buddies are connected

* Fred
* Bill

Bill: Yeh, how did it go yesterday
Jack: Not bad, was good
Bill: Tell me more

lets meet at lunch and will tell you everything

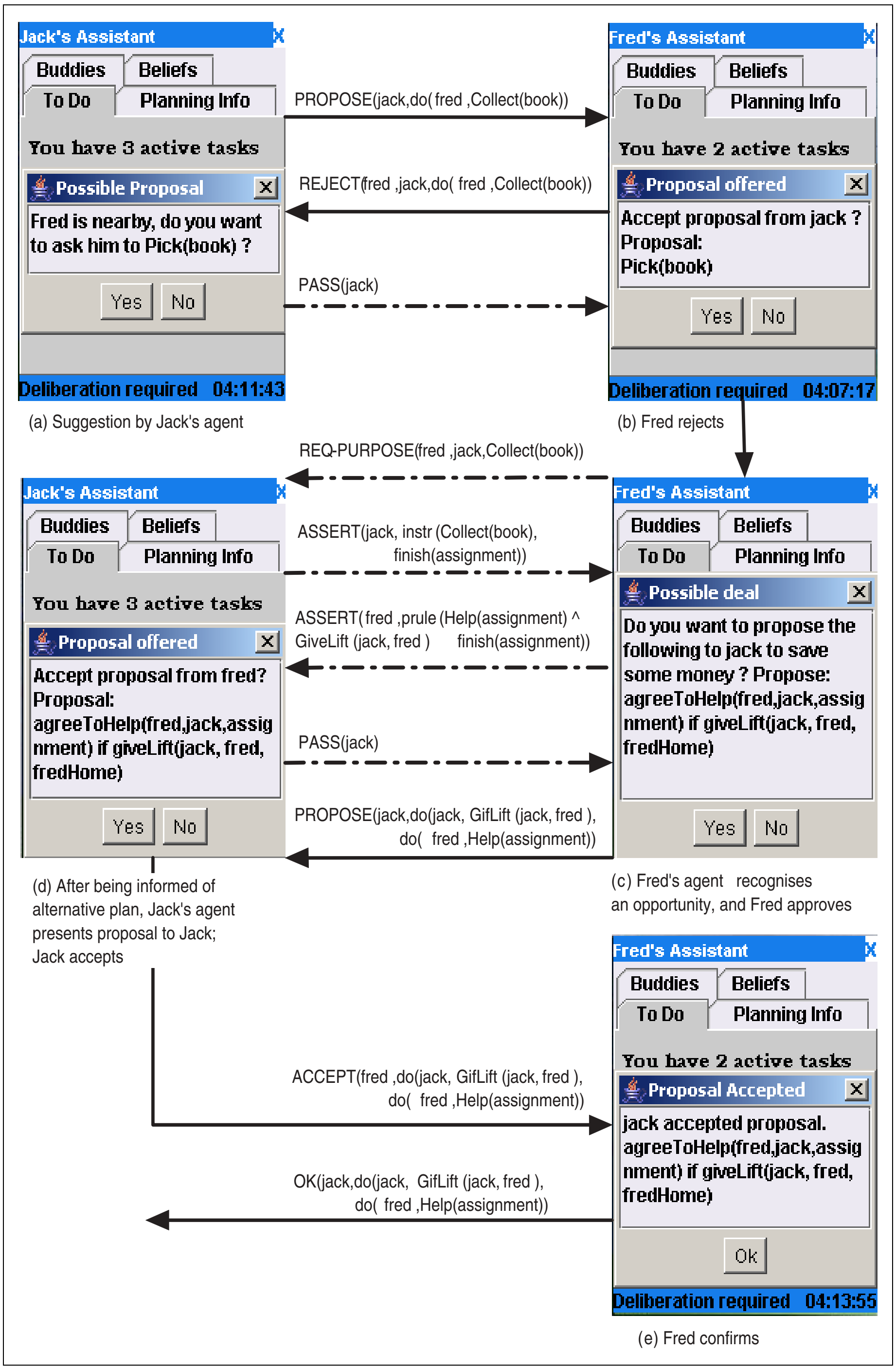
Deliberation required 03:36:31

Fig. 7. Screen shot of the prototype implementation

presents this proposal, which Fred rejects by clicking "No" (part (b)). However, Fred's agent attempts to discover the reason behind Jack's request, and based on that information, presents an alternative. Then Fred's agent asks Fred whether he would be interested in getting a lift home (part (c)). Fred clicks "Yes," after which Fred's agent sends a proposal to Jack's agent. Upon receipt, Jack's agent presents the proposal to Jack, who accepts the proposal by clicking "Yes" (part (d)). This approval is sent to Fred's agent, which informs Fred of the deal and sends a confirmation back to Jack (part (e)).

4 Related Work

A growing body of research looks at the importance of unintended interactions in task-oriented (typically workplace) settings, and on systems that partially support discovery of opportunity for such interactions [8, 10]. For many activities involving coordination between mobile users, geographic co-location is a key indicator of opportunity, and this frequently applies also to the problems we are interested in. Increasingly, location aware devices can provide robust access to such data. The ActiveCampus project [8] is one example of such an approach – deployed in a university setting and mediating serendipitous learning opportunities.



You have 3 active tasks

Possible Proposal

Fred is nearby, do you want to ask him to Pick(book) ?

Deliberation required 04:11:43

(a) Suggestion by Jack's agent

PROPOSE(jack,do(fred ,Collect(book))

REJECT(fred ,jack,do(fred ,Collect(book))

PASS(jack)

You have 2 active tasks

Proposal offered

Accept proposal from jack ?
Proposal:
Pick(book)

Deliberation required 04:07:17

(b) Fred rejects

REQ-PURPOSE(fred ,jack,Collect(book))

You have 3 active tasks

Proposal offered

Accept proposal from fred?
Proposal:
agreeToHelp(fred,jack,assignment) if giveLift(jack, fred, fredHome)

(d) After being informed of alternative plan, Jack's agent presents proposal to Jack; Jack accepts

ASSERT(jack, instr (Collect(book), finish(assignment))

ASSERT(fred ,prule (Help(assignment) ^ GiveLift (jack, fred) finish(assignment))

PASS(jack)

PROPOSE(jack,do(jack, GifLift (jack, fred), do(fred ,Help(assignment))

Possible deal

Do you want to propose the following to jack to save some money ? Propose:
agreeToHelp(fred,jack,assignment) if giveLift(jack, fred, fredHome)

(c) Fred's agent recognises an opportunity, and Fred approves

ACCEPT(fred ,do(jack, GifLift (jack, fred), do(fred ,Help(assignment))

OK(jack,do(jack, GifLift (jack, fred), do(fred ,Help(assignment))

You have 2 active tasks

Proposal Accepted

jack accepted proposal.
agreeToHelp(fred,jack,assignment) if giveLift(jack, fred, fredHome)

Deliberation required 04:13:55

(e) Fred confirms

Fig. 8. Screen shots of the scenario run

Our work uses techniques from automated negotiation from the field of Multi-Agent Systems (MAS) [22]. Other attempts to use agent-based techniques to support mobile users exist. However, most existing work on agents for mobile devices focuses on supporting single users [18] or collaborative teams executing routine tasks [3]. Our work is complementary to these efforts since we focus on unplanned, opportunistic collaboration. Paurobally and Jennings [14] uses automated negotiation for negotiating paid mobile services as opposed to coordination of activities.

5 Conclusions

We argued that automated negotiation technologies, from the multi-agent systems literature, could support coordination of activities in dynamic, mobile use contexts. We grounded our discussion in current views of mobility in the literature, and, through a narrative, identified key issues of mobile coordination and showed how they may be addressed using negotiation technologies. We presented a framework for supporting unplanned coordination based on our goal-directed automated negotiation framework. We presented a pilot application and illustrated its use through a scenario.

Our framework supports the features required to deal with the characteristics of impromptu mobile coordination discussed in section 2. The framework caters for the fluidity encountered in mobile use contexts, since coordination does not assume predetermined and pre-negotiated task structures. Moreover, the focus on tasks and their underlying goals also enables impromptu realisation of opportunities for coordinating activities. By expressing the resources and objectives explicitly, it becomes possible to build technology that processes this information in order to "allow more effective planning land flexible allocation of resources" [15].

A very important issue worth mentioning is "knowledge elicitation." The scenario implemented here was based on a set of hard-coded rules, based on a relatively naive representation of user tasks. In a realistic implementation, it would be unreasonable to expect users to pre-program their task structures into their devices. Moreover, even if users did manage to encode their task representation, an 'ontology' problem arises, since different users may represent the same tasks differently, and adequate translation between these description is likely to be nontrivial. However, in relatively well-structured domains, such as truck delivery, pre-programming typical task representations may be practical. Knowledge elicitation to initialise automated negotiating agents has begun to receive attention recently [2]. In less structured domains, it may be possible to use knowledge acquisition techniques based on machine learning [4] to enable agents to automatically build task structures by observing user behaviour. Once these structures are available, IBN may be used to automate interaction that exploits them.

Future work can be carried out in two directions: the technical and the user-oriented. Technical challenges include integrating our system with existing calendar and organiser applications that run on common mobile devices. Another technical challenge is finding a way to automatically capture the task knowledge in order to address the knowledge acquisition problem. From a user-oriented perspective, it is essential to study the design issues surrounding the interaction between the device and the user, as well as those relating to social interaction.

References

1. 3APL-M, 2004. http://www.cs.uu.nl/3APL-M.
2. J. J. Castro-Schez, N. R. Jennings, X. Luo, and N. Shadbolt. Acquiring domain knowledge for negotiating agents: a case study. *Human-Computer Studies*, 61(1):3–31, 2004.
3. H. Chalupsky, Y. Gil, C. A. Knoblock, K. Lerman, J. Oh, D. V. Pynadath, T. A. Russ, and M. Tambe. Electric elves: Applying agent technology to support human organizations. In H. Hirsh and S. Chien, editors, *Proc. IAAI-2001*. AAAI Press, 2001.
4. S. F. Chipman and A. L. Meyrowitz. *Foundations of Knowledge Acquisition: Machine Learning*. Kluwer Academic Publishers, 1993.
5. M. Dastani, F. de Boer, F. Dignum, and J.-J. Meyer. Programming agent deliberation: an approach illustrated using the 3APL language. In *Proc. AAMAS*. ACM Press, 2003.
6. A. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1), 2001.
7. C. A. Ellis, S. J. G. G. L, and Rein. Groupware: Some issues and experiences. *Communications of ACM*, 34(1):38–58, 1991.
8. W. G. Griswold, R. Boyer, S. W. Brown, T. M. Truong, E. Bhasker, G. R. Jay, and R. B. Shapiro. Using mobile technology to create opportunitistic interactions on a university campus. In *W.shop on Supporting Spontaneous Interaction in Ubiquitous Comp. Settings*, 2002.
9. K. V. Hindriks, F. S. de Boer, W. van der Hoek, and J.-J. Meyer. Agent programming in 3apl. *Autonomous Agents and Multi-Agent Systems*, 2(4):357–401, 1999.
10. E. A. Isaacs, J. C. Tang, and T. Morris. Piazza: a desktop environment supporting impromptu and planned interactions. In *Proc. ACM conf. on CSCW*, pages 315–324. ACM Press, 1996.
11. M. Kakihara and C. Sørensen. Mobility: An extended perspective. In R. S. Jr, editor, *Proc. 35th Hawaii Int. Conf. on Systems Sciences*. IEEE Press, 2002.
12. P. Luff and C. Heath. Mobility in collaboration. In *Proceedings of the 1998 ACM conference on Computer Supported Cooperative Work*, pages 305–314. ACM Press, 1998.
13. T. W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, 1994.
14. S. Paurobally, P. J. Turner, and N. R. Jennings. Automating negotiation for m-services. *IEEE Trans. on Systems, Man and Cybernetics (Part A: Systems and Humans)*, 33(6), 2003.
15. M. Perry, K. O'Hara, A. Sellen, B. Brown, and R. Harper. Dealing with mobility: understanding access anytime, anywhere. *ACM Trans. on Comp. Human Interaction*, 8(4), 2001.
16. I. Rahwan. *Interest-based Negotiation in Multi-Agent Systems*. PhD thesis, Department of Information Systems, University of Melbourne, Melbourne, Australia, 2004.
17. I. Rahwan, C. Graham, and L. Sonenberg. Supporting impromptu coordination using automated negotiation. In *Proc. Pacific Rim Int. Workshop on MultiAgent Systems*, 2004.
18. T. Rahwan, T. Rahwan, I. Rahwan, and R. Ashri. Agent-based support for mobile users using AgentSpeak(L). In P. Giorgini, B. Hederson-Sellers, and M. Winikoff, editors, *Agent Oriented Information Systems*, volume 3030 of *LNAI*, pages 47–62. Springer Verlag, 2004.
19. R. Reimann and E. Bacon. A scenario-based approach to creating interaction frameworks. In *Proc. Workshop on Tools, Conceptual Frameworks, and Empirical Studies for Early Stages of Design*, 2001.
20. N. M. Sadeh. *M–Commerce Technologies, Service, and Business Models*. Wiley, 2003.
21. S. Weiss. *Handheld Usability*. Wiley, Hoboken NJ, USA, 2002.
22. M. J. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, 2002.