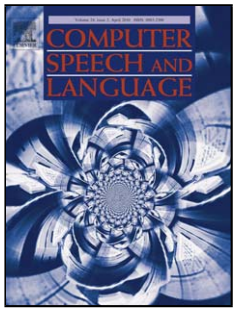
Multimodal and Mobile Conversational Health and Fitness Companions

Markku Turunen, Jaakko Hakulinen, Olov Ståhl, Björn Gambäck, Preben Hansen, Mari C. Rodríguez Gancedo, Raúl Santos de La Cámara, Cameron Smith, Daniel Charlton, Marc Cavazza



Title: Multimodal and Mobile Conversational Health and Fitness Companions

Authors: Markku Turunen, Jaakko Hakulinen, Olov Ståhl, Björn Gambäck, Preben Hansen, Mari C. Rodríguez Gancedo, Raúl Santos de la Cámara, Cameron Smith, Daniel Charlton, Marc Cavazza

PII: S0885-2308(10)00035-5
DOI: doi:10.1016/j.csl.2010.04.004
Reference: YCSLA 451

To appear in:

Received date: 30-4-2009
Revised date: 25-3-2010
Accepted date: 1-4-2010

Please cite this article as: Turunen, M., Hakulinen, J., Ståhl, O., Gambäck, B., Hansen, P., Gancedo, M.C.R., Cámara, R.S., Smith, C., Charlton, D., Cavazza, M., Multimodal and Mobile Conversational Health and Fitness Companions, *Computer Speech & Language* (2008), doi:10.1016/j.csl.2010.04.004

Multimodal and Mobile Conversational Health and Fitness Companions

Markku Turunen[*,a], Jaakko Hakulinen[a], Olov Ståhl[b], Björn Gambäck[b,c], Preben Hansen[b], Mari C. Rodríguez Gancedo[d], Raúl Santos de la Cámara[d], Cameron Smith[e], Daniel Charlton[e], Marc Cavazza[e]

[a]*Department of Computer Sciences, 33014 University of Tampere, Finland*
[b]*SICS, Swedish Institute for Computer Science AB, Box 1263, 164 29 Kista, Sweden*
[c]*Department of Computer and Information Science, Norwegian University of Science and Technology, Sem Sælands vei 7–9, 7491 Trondheim, Norway*
[d]*Telefonica I+D, C/ Emilio Vargas 6, 28043 Madrid, Spain*
[e]*School of Computing, University of Teesside, Middlesbrough TS1 3BA, United Kingdom*

Abstract

Multimodal conversational spoken dialogues using physical and virtual agents provide a potential interface to motivate and support users in the domain of health and fitness. In this paper we present how such multimodal conversational Companions can be implemented to support their owners in various pervasive and mobile settings. We present concrete system architectures, virtual, physical and mobile multimodal interfaces, and interaction management techniques for such companions. In particular, we present how knowledge representation and separation of low-level interaction modelling from high-level reasoning at the domain level makes it possible to implement distributed, but still coherent, interaction with Companions. The distribution is enabled by using a dialogue plan to communicate information from domain level planner to dialogue management and from there to a separate mobile interface. The model enables each part of the system to handle the same information from its own perspective without containing overlapping

[*]Corresponding author
Email addresses: mturunen@cs.uta.fi (Markku Turunen), jh@cs.uta.fi (Jaakko Hakulinen), olovs@sics.se (Olov Ståhl), gamback@sics.se (Björn Gambäck), preben@sics.se (Preben Hansen), mcrg@tid.es (Mari C. Rodríguez Gancedo), e.rsai@tid.es (Raúl Santos de la Cámara), c.g.smith@tees.ac.uk (Cameron Smith), d.charlton@tees.ac.uk (Daniel Charlton), m.o.cavazza@tees.ac.uk (Marc Cavazza)

logic, and makes it possible to separate task-specific and conversational dialogue management from each other. In addition to technical descriptions, we present results from the first evaluations of the Companions interfaces.

Key words: Companions, Embodied Conversational Agents, Conversational Spoken Dialogue systems, Mobile interfaces, Cognitive Modelling, Dialogue management

1. Introduction

Most existing spoken dialogue systems provide a single interface to solve a well-defined task, such as booking tickets or providing timetable information. However, there are emerging areas that differ dramatically from task-oriented systems. In domain-oriented dialogues (Dybkjaer et al., 2004) the interaction with the system, typically modelled as a conversation with a virtual human-like character, can be the main motivation for the interaction. These systems are often multimodal, and may be implemented in pervasive computing environments where various mobile, robotic, and other novel interfaces are used to communicate with the system.

In the EC-funded COMPANIONS project (Wilks, 2007) we have developed conversational Companions that build long-lasting relationships with their users via mobile and physical agent interfaces. Such systems have different motivations for use compared to traditional task-based spoken dialogue systems. Instead of helping with a single, well-defined task, a companion provides long lasting support and companionship on a daily basis.

One of the developed companions, and the topic of this paper, is the conversational Health and Fitness Companion, which helps users to maintain a healthy lifestyle. There are several examples of commercial systems in the domain of health and fitness, in particular in exercise domain. Some of the most well known examples are miCoach from Adidas and NIKE+. [1] More in line with the present work, MOPET (Buttussi and Chittaro, 2008) is a PDA-based personal trainer system supporting outdoor fitness activities. MOPET is similar to a Companion in that it tries to build a relationship with the user, but there is no real dialogue between the user and the system and it does not support speech input or output. Neither does MPTrain/TripleBeat (Oliver and Flores-Mangas, 2006; de Oliveira and Oliver, 2008), a system that runs

[1] `www.micoach.com` and `www.nike.com/nikeplus`

on a mobile phone and aims to help users to more easily achieve their exercise goals. The system selects music indicating the desired pace and different ways to enhance user motivation, but without an agent user interface model. InCA (Kadous and Sammut, 2004) is a spoken language-based distributed personal assistant conversational character with a 3D avatar and facial animation. Similar to the Mobile part of Companion, the architecture is made up of a GUI client running on a PDA and a speech server, but the InCA server runs as a back-end system, while the Companion utilizes a stand-alone speech server.

The difference between the Health and Fitness Companion and most of the existing health related systems is that our Companion prototype aims to be a peer rather than an expert system in health-related issues. The ultimate goal of the Health and Fitness Companion is to support overall lifestyle changes in the user's daily habits rather than giving detailed advice on any specific health related issues. A lifestyle change often requires great motivation and the Health and Fitness Companion aims to support building and maintaining this motivation. The method the Health and Fitness Companion uses to support the user motivation is to set up a long lasting social and emotional relationship with the user. Since people build relationships mostly in face-to-face conversations, a conversational embodied agent is a potential platform to build such a relationship (Dybkjaer et al., 2004).

Both mobile usage and physical and virtual agent interfaces can support the goal of making a spoken dialogue system part of its user's everyday life, and building a meaningful relationship between the system and the user. There are numerous examples of virtual embodied conversational agents, but only a few examples of physical conversational agents. Physical agents, however, can be particularly efficient in applications that try to evolve dialogue systems into being part of people's lives.

While naturalistic human-like physical robots are under development, there is room for a variety of different physical interface agents ranging from completely abstract (e.g., simple devices with lights and sound) to highly sophisticated anthropomorphic apparata. For example, Marti and Schmandt (2005) used several toy animals, such as bunnies and squirrels, as physical embodied agents for a conversational system. Another example is an indoor guidance and receptionist application involving a physical human-like interface agent that combines pointing gestures with conversational speech technology (Kainulainen et al., 2005). Some physical agent technology has also been commercialized. Examples include the Paro Therapeutic Robot

[2], and the wireless Nabaztag/tag rabbit [3], which has resulted active user community.

In this work, we have used the Nabaztag/tag device as a multimodal physical interface to create a conversational Health and Fitness Companion. In addition to the physical interface we have used a mobile device interface for outdoor usage, and a graphical, lifelike virtual agent interface in certain dialogues to provide rich multimodal interface when those features are needed.

In these kinds of applications where multiple user interfaces can be used to access parts of the same functionality, e.g., the same physical exercise information, and the system interacts with a user many times over a long time period, modelling the interaction and domain quickly becomes complex. To address this, the system must include a model capable of reasoning about the domain, and learn from the user and his/her actions to provide meaningful interaction, such as to provide reasonable guidance on the user's health and progress as the user's condition alters over time. Such reasoning should be concentrated on one component, instead of duplicating the logic to keep the system maintainable. Still, the information must be communicated over different interfaces and to the components inside them. Therefore, modularization of the system and appropriate knowledge representation become vital.

On the dialogue management level, a common way to take some complexity away from the dialogue manager and limit its tasks more specifically to dialogue management is to separate domain specific processing, such as database queries, into a back-end component. Many researchers have worked with separating generic dialogue management processes from the domain specific processes. Example solutions include shells (Jönsson, 1991) and object oriented programming methods (Salonen et al., 2004; O'Neill et al., 2003). On the other hand, a simple back-end interface, e.g., SQL queries, can be included as configuration parameters (Pellom et al., 2000). Since dialogue management is usually based on some clearly defined model, such as state transition networks or form filling, keeping domain specific processing in the back-end makes it possible to implement dialogue management purely with the selected model without the unnecessary complexity of domain specific

[2] http://www.parorobots.com/

[3] www.nabaztag.com

information.

The Health and Fitness Companion is based on a model where the domain specific module is more than just a simple backend database and includes active processing of domain information, reasoning, learning, and other complex processes. We call such a component cognitive model. While the task of a dialogue manager is to maintain and update the dialogue state, the cognitive model reasons using domain level knowledge. The separation of the tasks between the different parts is not trivial. For example, managing dialogue level phenomena, such as error handling and basic input processing, are tasks clearly in the area of a dialogue manager. However, cognitive modelling can help in error handling by spotting input that seems suspicious based on domain level information, and in input parsing by providing information on potential discussion topics. The solution we have devised is to have the cognitive model to produce a dialogue plan for the dialogue management in the Home Companion system. The dialogue manager in the Home Companion provides parsed user inputs to the cognitive model and to the Mobile Companion. The Mobile Companion provides similar input back to the Home Companion, which communicates it back to the Cognitive Model.

In the rest of this paper, Section 2 introduces the concept of Health and Fitness Companion including the different types of mobile, physical and virtual agent interfaces that realize the concept. For the Cooking Companion we present the initial prototype interface developed to demonstrate the concept. For the Home and Mobile Companions, which have been fully implemented, we present the technical solutions to implement such Companions. Section 3 details the software architectures used to implement physical and mobile Companions interfaces. Section 4 introduces a novel interaction management solution which separates cognitive modeling from dialogue management and enables their flexible interoperability and distributed but coherent dialogues between different agent interfaces. Together, the solutions presented in these sections result in concrete software components to implement further Companion applications. Some of the solutions, such as the PART system used in the Mobile Companion and the Jaspis architecture and the jNabServer software used for the Home Companion have been released as open source software. Finally, Section 5 presents the key results from the early evaluations of the first Health and Fitness Companion prototypes. Section 6 recapitulates the key parts of the discussion.

2. Companion Interfaces

As discussed previously, the Health and Fitness Companion interacts with its user in three main forms: as a physical Home Companion to provide general support for a healthy lifestyle, as a Mobile Companion to support the user during physical outdoor activities, and as a virtual Cooking Companion to support eating and cooking related lifestyle activities. Together, these interfaces aim to build a long lasting relationship with the user.

2.1. Example Dialogue

The example below show a complete dialogue which a user can have with the three different Health and Fitness Companion interface prototypes. Dialogues 1 and 2 are carried out with the Home Companion interface, Dialogues 3 and 4 are done with the Mobile Companion interface, and Dialogue 5 demonstrates a conversation with the Cooking Companion interface.

(1) **During breakfast**

C1: Do you have anything arranged for today?

U1: I'm playing Squash

C2: Is that before dinner?

U2: Yes

C3: Okay. Are you going to take the train to work?

U3: Yes

C4: Right. You could take in a packed lunch to work?

U4: Yes, that would be fine

(2) **Later on in the evening**

C5: Welcome back! How did your game of Squash go?

U5: The game was cancelled, so I went to the gym

C6: Okay. Did you take the train to work?

U6: Yes, I did

C7: Right. Did you take in a packed lunch?

U7: No. I was in a rush so I went to the cafeteria

(3) **A physical exercise with the Mobile Companion**

C8: Hello, what kind of exercise would you like to do?

U8: I would like to do some running

C9: Ok, running it is. Select start when you are ready.

(4) **After 12 minutes**

C10: You have been running for 12 minutes. The distance is 2.5 kilometres; the current pace is 5 minutes per kilometre.

U9: Okay, please play some relaxing music

(5) **Cooking with the Cooking Companion**

C11: Good, let's start preparing the ingredients. You should remove the stalk and seeds from the red pepper, then chop it up.

U10: And I should chop up the onion now too?

C12: Yes, that's right, peel and chop the onion. You could peel and crush the garlic too afterwards.

2.2. Home Companion Interface

The Home Companion interface is a physical agent that resides in the home of its user, for example, in the kitchen table, where it is natural to have a conversation before going to work or start a physical activity. This setup is illustrated in Figure 1. The physical agent is implemented with the Nabaztag/tag WLAN rabbit that provides audio output and push-to-talk audio input, RFID-based interaction, moves its ears, and operates four coloured lights to signal its status.

In order to provide meaningful advice to the user on his or her daily activities the Home Companion communicates with the user in two main dialogue phases; the planning phase where the system talks about the coming day with the user (as demonstrated in Dialogue 1), and the reporting phase where the users actual activities are assessed with reference to what was agreed on earlier (Dialogue 2). The covered topics include a range of common situations and activities relevant for health and fitness conversations. In overall, the Home Companion is able to discuss the following topics: travelling to work, getting lunch, activities to be performed before dinner, getting dinner, and activities to be performed after dinner. It knows activities such as playing

Figure 1: The Home Companion interface in a typical usage situation (see http://www.youtube.com/watch?v=KQSiigSEYhU for a video presentation)

football, squash, or badminton; going to the gym or shopping; and watching television or reading a book.

2.3. Mobile Companion Interface

The Mobile Companion runs on Windows Mobile devices and can be used during outdoor exercise activities such as walking, jogging or cycling. The Mobile Companion can download the plan of the day which the user has agreed on with the Home Companion. The Mobile Companion will then suggest an exercise, based on the user's current location, time of day and the plans made earlier, or, if there are no suitable exercises, ask the user to define one, as in Dialogue 3. Once an exercise has been agreed upon, the Companion will track the progress (distances travelled, time, pace, and

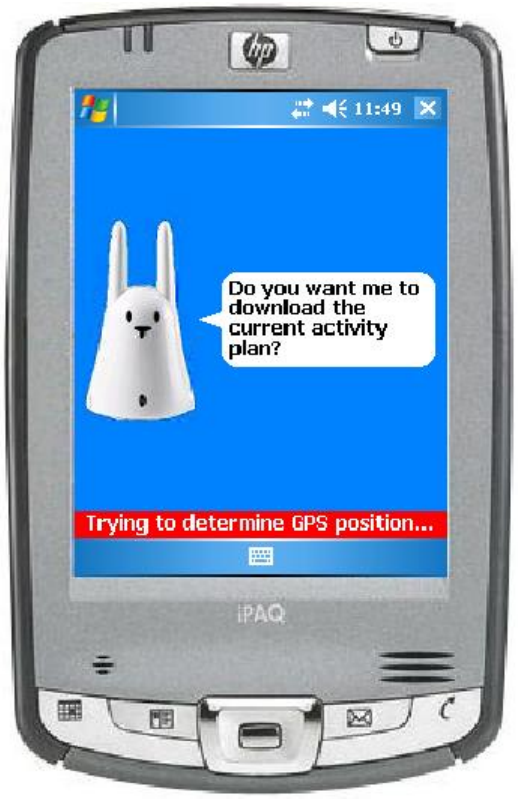


Figure 2: Mobile Companion interface

calories burned) using a GPS receiver. While exercising, the user can ask the Companion to play music or to give reports on how the user is doing. When the exercise is over, the Mobile Companion can upload the result to the home system. The data gathered during an exercise is also stored locally in the device's record store, which allows for access to data from previous exercises (for comparison purposes) even if the mobile internet connection is temporarily unavailable.

The Mobile Companion's graphical user interface, as seen in Figure 2, consists of a single screen showing a static image of the Nabaztag rabbit, along with a speech bubble. The rabbit image is intended to give users a sense of communicating with the same Companion, no matter the interface (Home or Mobile). To further the feeling of persistence and help users associate the two interfaces, the Home and the Mobile Companions use the same

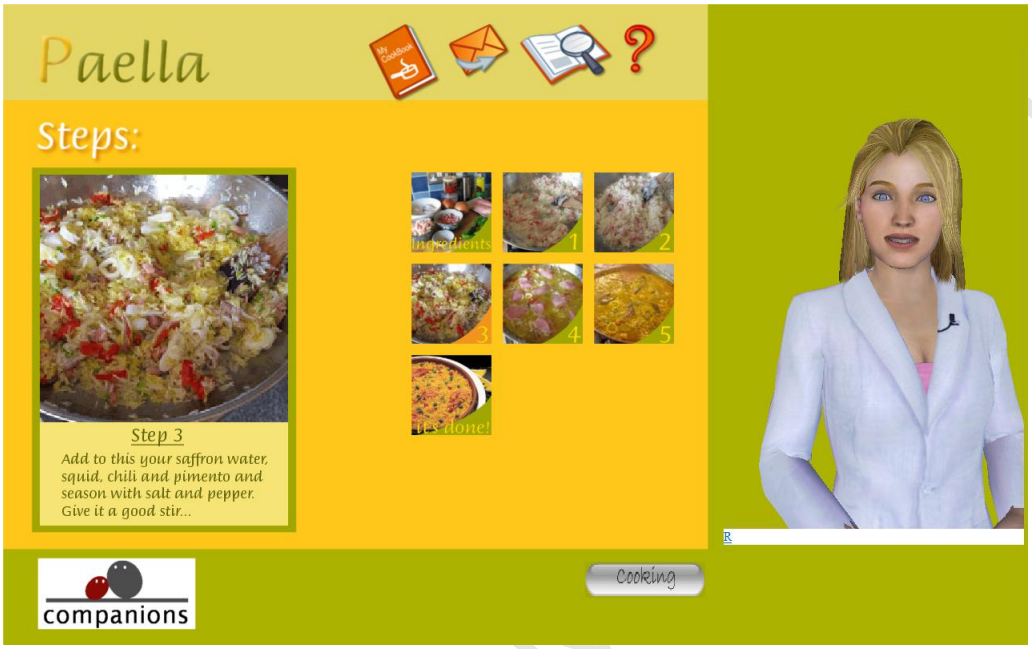


Figure 3: Cooking Companion interface

synthesized voice, while the Cooking Companion uses a different avatar and thus a different text-to-speech engine that produces a voice clearly different to the bunny's, helping the user differentiate between both ECAs.

All spoken messages are also shown as text in the speech bubble. The user can provide input via voice, by pressing hardware buttons on the mobile device, and in some situations, by tapping on a list of selections on the touch screen. Screen-based input is used, for example, when identifying the current exercise route (from a list of route names, defined by the user in previous sessions), but also when ASR errors occur, to perform error correction. In the latter case, the user is presented with a list of input strings that the Companion is able to understand in the current context, and must select one to continue the dialogue.

2.4. Cooking Companion Interface

One of the main aims of the Health and Fitness Companion is to provide tips for a healthier lifestyle, and one of the key aspects is a correct nutrition. The Cooking Companion is a virtual embodiment of a dietary advisor, which:

1. Presents a set of possible recipes to the user, based on availability of ingredients or home delivery food.
2. Informs the user of the appropriateness of each choice, based on available information such as the user's likes and dislikes, authorised medical and nutritional databases, hers/his current physical condition, special dietary requirements (e.g., allergies) and the physical exercise scheduled by the rest of the Health and Fittness Companion.
3. Helps in the preparation of the selected recipe using multimedia and/or dialogue.

As an example, the user wants to eat seafood paella. The system then inspects all of the aforementioned parameters and conclude advising the user to prepare vegetable paella, based on his current exercise schedule and high blood fat levels (see Dialogue 5 for an example). Then the system instructs the user on how to prepare the dish using videos and speech.

In Figure 3, the interface for the Cooking Companion prototype is shown. The system augments plain spoken dialogue with the help of an ECA, powered by a third party engine [4] and a finger-operated touchscreen interface. The interaction is multimodal, intertwining speech utterances and finger pointing. The ECA, modelled as a photorealistic human, includes some advanced gesturing capabilities that have been demonstrated to make the avatar both more engaging and understandable (López Mencía et al., 2006; López et al., 2007, 2008).

The Cooking Companion interface is presented here as an example of how the Companions paradigm can use different agent interfaces to achieve certain aspects of companionship. Currently, a prototype to demonstrate its functionality has been implemented. In the rest of the paper, we focus on the Home and the Mobile Companions, which have been fully implemented as described in the forthcoming sections. Further information on the Cooking Companion and its relation to the overall Health and Fitness Companion is presented in (Hernández et al., 2008) and (Turunen et al., 2008b).

3. Companions Architectures

In order to construct Health and Fitness Companions, we need concrete system architectures and software components that are flexible enough to

[4] Haptek Player, Haptek Inc.: `www.haptek.com`

implement different interfaces running on different hardware and software platforms. At the same time, we need to maintain shared knowledge for the Healt and Fitness Companion. In the following, the Home Companion and the Mobile Companion architectures and their main components are presented. These provide foundations for the interaction management solutions presented in the following sections.

3.1. Home Companion

The Home Companion is implemented on top of Jaspis [5] (Turunen et al., 2005), a generic open source software architecture designed for adaptive spoken dialogue systems. In order to implement the physical Health and Fitness Companion, the Jaspis architecture was extended to support interaction with physical agents, and the Nabaztag/tag device in particular. For this purpose, the jNabServer software [6] was created to handle local communication with Nabaztag/tag. This solution offers full control over Nabaztags, including RFID-reading, and makes it possible to use custom programs and technologies to process inputs and outputs, such as the speech recognition and TTS software used in the Health and Fitness Companion.

Figure 4 illustrates the Home Companion architecture. The top-level structure of the system is based on managers, seven of them in total, which are connected to the central Interaction Manager (a HUB/Facilitator style component) using a star topology structure. In addition, the application has an Information Manager (a database/blackboard style component) that is used by all the other components to store and share information. The Information Manager provides a high-level interface to the Information Storage, which contains all the dialogue management and the cognitive model structures (e.g., confirmation pools and activity models). The communication between the these two models is based on a dialogue plan, which is also in the Information Storage. These are presented in detail in Section 4. The Information Manager and the Information Storage do not modify this content, they just handle the technical management of data in XML format. The details of this layered information management approach are beyond the scope of this paper, and are discussed in detail in (Turunen, 2004).

All components in the system are stateless, meaning they save all their internal data to the central database when they end processing (or go to

[5] `www.cs.uta.fi/hci/spi/Jaspis`

[6] Java-based Nabaztag/tag Server: `www.cs.uta.fi/hci/spi/jnabserver`

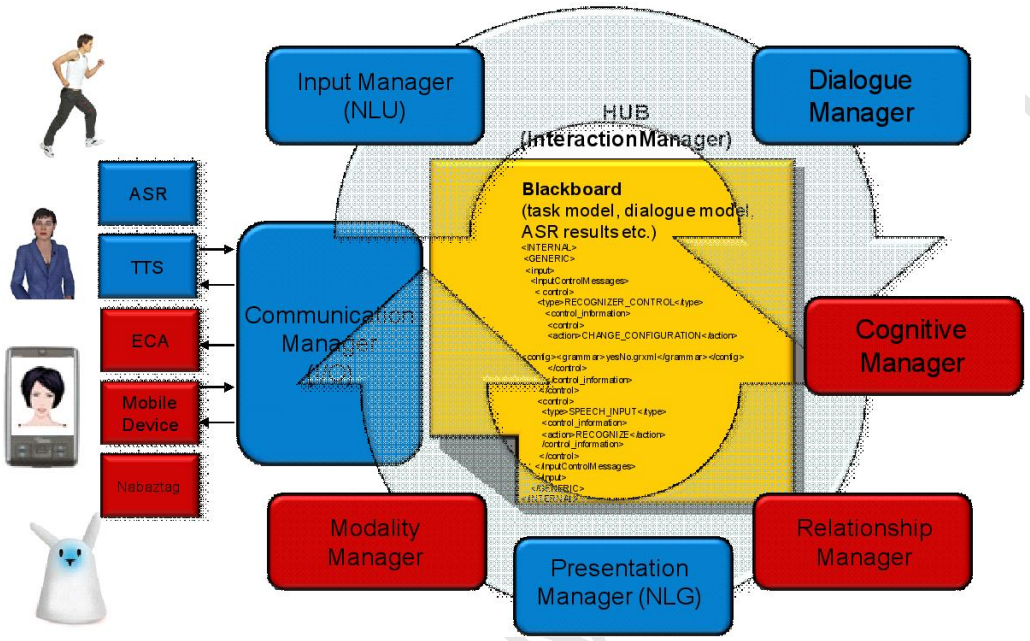


Figure 4: The Home Companion architecture

sleep), and update their internal models when they start processing (wake up) again. This way, the overall system is modular and can be extended easily. We believe this is a requirement needed to construct complex conversational system such as the Health and Fitness Companions, since it separates data from its processing, allowing efficient iterative development, for example, to visualize, debug, and teach complex systems easily (Hakulinen et al., 2007). Most importantly, it makes possible the architecture level modularization and adaption scheme described next.

Modularity is further supported in this architecture by the agents-managers-evaluators paradigm that is used across all system modules. In this approach, all tasks are handled by compact and specialized agents located in modules and coordinated by managers. When one of the agents inside a module, such as the Dialogue Manager, is going to be selected, each evaluator in the module gives a score for every agent in the module. These scores are then multiplied by the local manager, which gives the final score, a suitability factor, for every agent. As an example, the multi-agent architecture of Jaspis (Turunen et al., 2005) is used heavily in dialogue management; in the current

prototype, there are 30 different dialogue agents, some corresponding to the topics found in the dialogue plan, others related to error handling and other generic interaction tasks. These agents are dynamically selected based on the current user inputs and overall dialogue context with using three rule-based reasoning evaluators.

For speech input and output, Loquendo ASR and TTS components (Loquendo, 2008) have been integrated into the Communication Manager. ASR uses grammars in "Speech Recognition Grammar Specification" (W3C) format. In these grammars, semantic tags in "Semantic Interpretation for Speech Recognition (SISR) Version 1.0" (W3C) format are used to provide information for the NLU component (described in detail in the following sections). Domain specific grammars were derived from early informal testing and Wizard-of-Oz sessions conducted during the iterative development of the system. The data collected from these initial studies has been used to build the HFC grammars. There is a set of about 50 grammars and the system dynamically selects these according to the current dialogue state. The grammar size for most grammars is about 1400 words, with a total of about 900 grammar rules. The vocabulary coverage is balanced across the four relevant domain parts of the activity model: transportation, physical activity, leisure and food. Natural language generation is implemented using a combination of canned utterances and Tree Adjoining Grammar-based generation. The grammar-based generation is used mostly in confirmations, while canned utterances with multiple options for each situation are used for most questions about users' day.

3.2. Mobile Companion

The Mobile Companion is realised by two components running on a Windows Mobile device, a Java midlet that handles the graphical user interface and the dialogue with the user, and a speech server that performs ASR and TTS operations on request by the midlet (see Figure 5). The Java midlet is built using the PART library, [7] and uses the Hecl scripting language[8] for GUI and dialogue management. The speech server uses Loquendo ASR (speaker-independent) and TTS libraries for embedded devices (Loquendo, 2008), and SRGS 1.0 grammars. Pre-compiled grammars are loaded dynamically. The

[7] Pervasive Applications RunTime: `part.sourceforge.net`

[8] `www.hecl.org`

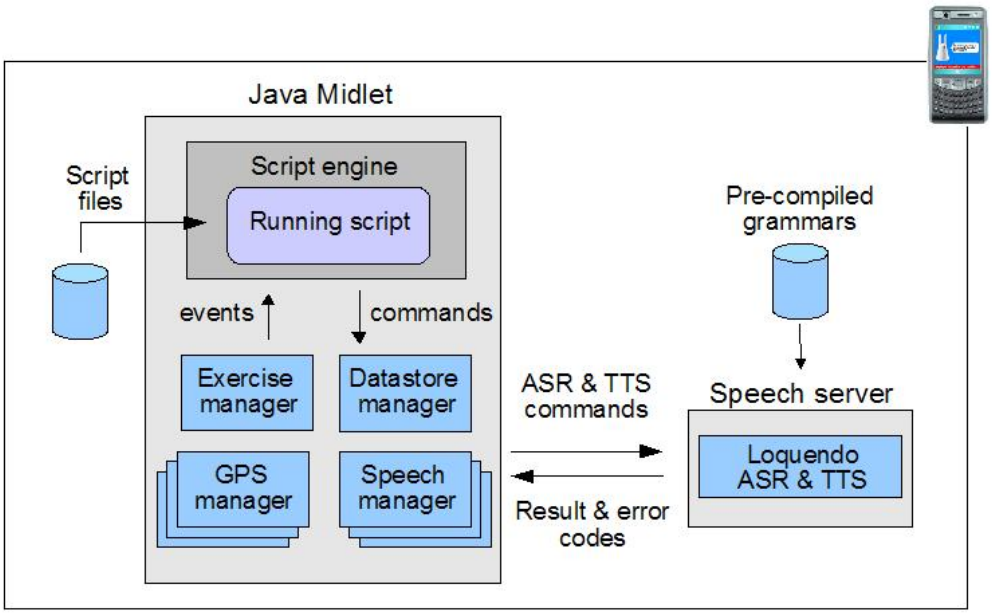


Figure 5: Mobile Companion Architecture

vocabulary and grammar is more restricted on the mobile device than in the Home Companion: its seven different grammars have in total vocabulary size of about 100 words.

While the Mobile Companion itself is running as a stand-alone system, the communication with the Home Companion requires the Mobile Companion to have access to the Internet, for instance via WLAN or 3G/GPRS. The Mobile Companion maintains a persistent data store in device to store user model (name, age, gender, weight, etc.), and the exercise results. Saving the exercise results allows the Companion to compare the progress of an exercise with previous exercises of the same kind. For instance, if the Companion knows that the user is currently cycling from home to work, it can provide feedback on how the user is doing compared to previous sessions. This allows for status messages like "*You are currently 1 minute and 23 seconds behind yesterday's time.*" The Mobile Companion is described in detail by Ståhl et al. (2008, 2009).

4. Dialogue Management and Cognitive Modelling

As mentioned in Section 1 interaction management easily becomes too complex without clear separation of conversational, task, and domain specific information and components. To address these challenges, we introduce a novel solution for dialogue management and cognitive modelling in this domain.

Traditional architectures for dialogue includes three parts: an input module, which receives user input and parses it into a logical form, dialogue management, which maintains and updates dialogue state based on user input and generates output requests, and an output module, which generates natural language output to user based on the requests. In the case of the Health and Fitness Companion, a special component, Cognitive Model (CM) is introduced as separate component from the Dialogue Manager (DM), as seen in Figure 8. The Cognitive Model contains what can be considered the higher-level cognitive processes of the system. The Dialogue Manager takes care of conversational strategies. It presents questions to a user based on the dialogue plan, maintains a dialogue history tree and a dialogue stack, and communicates facts and user preferences to the Cognitive Model. The Dialogue Manager also handles error management, supports user initiative topic shifts, and takes care of top level interaction management, such as starting and finishing of dialogues. The Dialogue Manager and the Cognitive Model communicate using a dialogue plan.

In the rest of this section we present how the DM of the Home Companion, the CM component, and the Mobile Companion interact to crete coherent long-lasting dialogues with the Companion.

4.1. Dialogue Plan

The communication between the Home and the Mobile Companion dialogue managers and the CM is based on a dialogue plan. Various kinds of dialogue plans (Larsson and Traum, 2000; Jullien and Marty, 1989) have been used inside dialogue managers in the past. A plan usually models what the system sees as the optimal route to task completion.

In Health and Fitness Companion, the CM provides a plan on how the current task (planning a day, reporting on a day) could proceed. The plan consists of a list of domain specific predicates. Figure 6 contains two items from the start of a plan for planning the day with the user in the morning. The first plan item (`QUERY-PLANNED-ACTIVITY`) can be realized as the

question "*Anything interesting planned for today?*" by the NLG component.

During the interaction, new relevant information may become available from the user. The information includes statements on the user's condition (*tired*), user commitments to the system (*will walk to work*), user preferences (*does not like cafeterias*) and user reports on past activity (*took a taxi to work*), which can be accomplishments or failures of earlier commitments. The DM provides this information to the CM, piece by piece as it becomes available. As the CM receives the information, it updates the dialogue plan as necessary and builds up the day plan or report. Query type items, whose information has been gathered, disappear from the dialogue plan, while new items may appear when the dialogue progress.

As discussed above, the messages sent by the DM to CM can add new information (predicates) to the CM state. The DM can also remove information from the CM if previously entered information is found to be untrue. At the same time, the information is uploaded to a web server, where the Mobile Companion interface can access it any time.

```
<plan>
 <plan-name>Generate-Task-Model-Questions</plan-name>
 <plan-item>
   <action>QUERY-PLANNED-ACTIVITY</action>
 </plan-item>
 <plan-item>
   <action>SUGGEST-TRAVEL-METHOD</action>
   <param>CYCLING-TRAVEL</param>
   <param>HOME</param>
   <param>WORK</param>
 </plan-item>
```

Figure 6: Start of a plan

The DM in the Home Companion can follow the dialogue plan produced by the CM step by step. Each step usually maps to a single question, but can naturally result in a longer dialogue if the user's answer is ambiguous, or error management is necessary, or if the DM decides to split a single item into multiple questions. For example, the two dialogue turn pairs seen in Figure 7 are the result of a single plan item (`QUERY-PLANNED-ACTIVITY`). Since the first user utterance does not result in a complete, unambiguous

predicate, the DM asks a clarification question. A single user utterance can also result in multiple predicates (e.g., *will not take the bus, has preference for walking*).

When the Mobile Companion interface is activated, it downloads the current day plan from the web server and uses it as a basis for the dialogue it has with the user. The exercise which will then take place can be linked to an item in the day plan, or it can be something new. As the exercise is completed (or aborted), information about this is uploaded to the web server. The DM of the Home Companion can download this information from the server. This information is relevant to the DM when the user is reporting on a day. The DM reports the downloaded information back to the CM when the dialogue plan includes related items. The DM may also provide some feedback to the user based on the information, for example, how the exercise relates to the current plan and the overall situation. It is noteworthy that the CM does not need to differentiate in any way, whether the information about the exercise is originating from the Mobile Companion, or was gathered in a dialogue with the Home Companion.

Similarly, clarifications and confirmations are not directly visible to the CM. The DM can confirm items immediately (for example, when low confidence scores are reported by the ASR component) or it can delay confirmations to generate a single confirmation for multiple items at an appropriate moment.

In this model, the DM has freedom to choose how to operate on the base of the CM information. Most importantly, when presenting questions and suggestions to the user, the DM is free to choose any item in the plan, or even to do something not included in the plan at all. When information from the Mobile Companion is available, it can direct where the dialogue starts from. The DM could also decide to do some small-talk to introduce sensitive topics, which can be useful in managing the user-system relationship (Bickmore and Picard, 2005). The communication that is not related to the domain does not reach the CM at any point. In another words, DM takes care of the conversational dialogue topics and meta-communication, while CM is involved in task-related topics.

In addition to the essential information, such as those presented in Figure 7, the CM can include additional annotation in the plan. First, DM can mark certain information requests as being important. If information is marked important, it is likely, but not certain, that the DM will explicitly confirm it. Second, CM can explicitly request a confirmation by generating

```
(  <plan-item>
<action>QUERY-PLANNED-ACTIVITY</action>
</plan-item>)
```

C1: Good morning. Anything interesting organized for today?

U1: I'm going jogging.

```
(<pred>
<action>PLANNED-ACTIVITY</action>
<param>ACTIVITY-JOGGING</param>
<param>unknownTime</param>
</pred> )
```

C2: Is that jogging exercise before dinner?

U2: No, its after.

```
(  <pred>
<action>PLANNED-ACTIVITY</action>
<param>ACTIVITY-JOGGING</param>
<param>AFTER-DINNER</param>
</pred> )
```

Figure 7: A dialogue fragment and corresponding plan items and predicates

a separate plan item. For example, if the users report on having run much more than they are likely to be capable of in their condition, the CM can generate a confirmation plan item. It is worth noting, that the DM cannot do reasoning on such a high level (as it would be not meaningful to duplicate such information in DM), and therefore the CM must participate in error handling in such cases.

4.2. Cognitive Modelling

In order to make the interaction with the Health and Fitness Companion coherent, we need to provide a shared cognitive model for the Companion. The system architecture provides support for this in the form of the separate

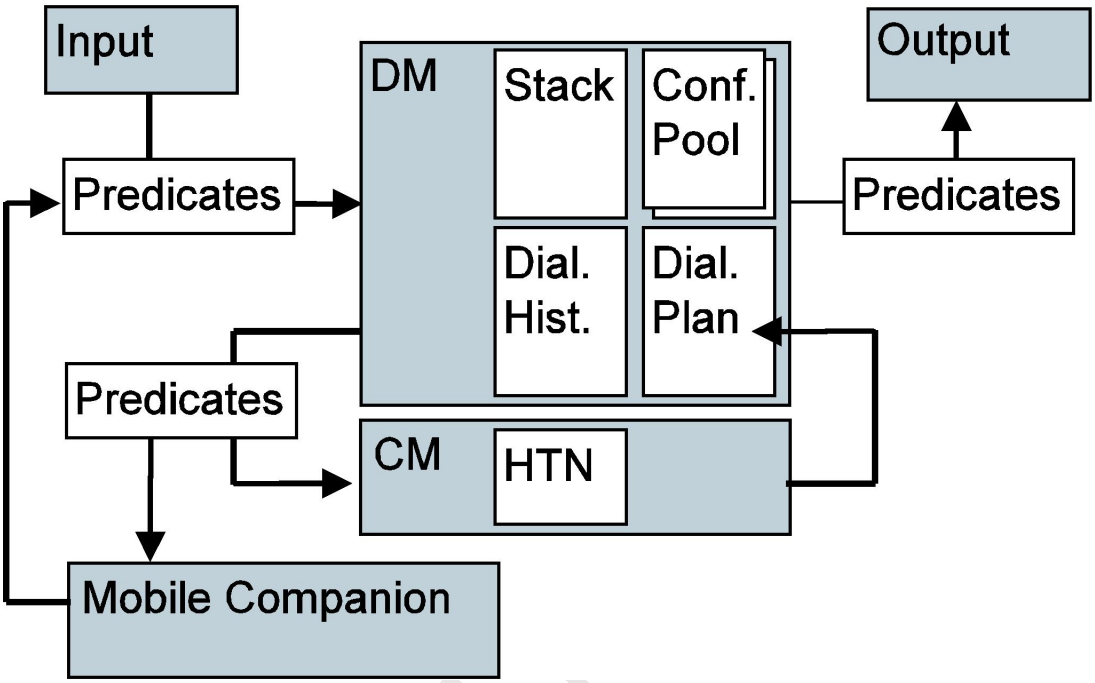


Figure 8: Dialogue management and cognitive modelling components and their information passing

Cognitive Model Manager. The task of the CM is to model the domain, i.e., know what to recommend to the user, what to ask from the user and what kind of feedback to provide.

At the core of the CM is an activity model, which decomposes the day into a series of activities for an office worker during a typical working day. These activities cover transportation to/from work, post-work leisure activities and meals (both in terms of the food consumed and how this food is obtained). A Hierarchical Task Network (HTN) planner is used to generate the activity model in the form of an AND/OR graph.

The planner that implements the activity model includes 16 axioms, 111 methods (enhanced with 42 semantic categories and 113 semantic rules), and 49 operators. The cognitive modelling of the Health and Fitness Companion is presented in detail by Hakulinen et al. (2008); Cavazza et al. (2008); Smith et al. (2008); Turunen et al. (2008a). Here, we focus in its interoperability with the dialogue management.

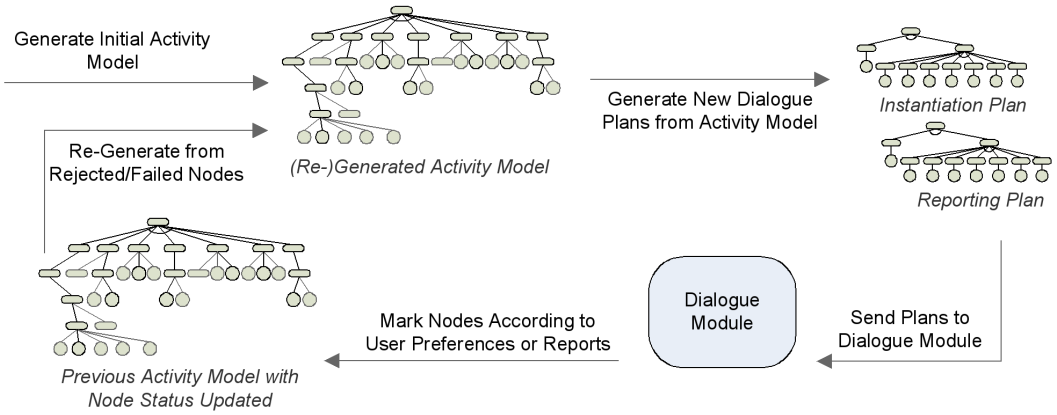


Figure 9: Activity model cycle

The Health and Fitness dialogues have currently two phases. The planning phase involves the system operating through a global interaction cycle, integrating dialogue with planning to construct the user's activity model. This process is illustrated in Figure 9 with a detail of the graph in Figure 10. The first step consists of generating an initial activity model for the user, based on default knowledge, along with any previously captured information on user preferences. Throughout this interaction cycle, each time the Planner generates a candidate activity model, it generates a corresponding dialogue plan, which is then used as a basis for the dialogue to enquire about user preferences and make suggestions in relation to the planned activities.

User responses are used to update the activity model, with the user utterance mapped to the predicates used within the planning domain and semantic categories associated with the domain methods. If a user response validates part of the existing activity model, that part of the model is marked as planned and will not appear in the dialogue plan. If the user response is incompatible with the current activity model, either through explicit rejection or stating of a conflicting preference, the preference is stored so that the relevant items remains unplanned and inactive.

The cycle continues with the activity model being re-generated. Those tasks that have been accepted, that is, marked as planned, are preserved while the unplanned parts are re-planned making use of the latest preferences provided by the user. An updated dialogue plan is then generated and the dialogue with the user continues until the user has agreed on a fully planned

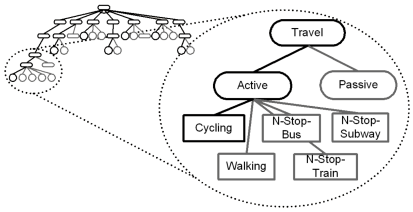


Figure 10: Activity Model Items

activity model.

The reporting phase is accomplished in a similar manner, with a dialogue plan asking the user questions about what he/she did and the user utterances being used to produce an activity model with completed activities.

4.3. Language Understanding and Dialogue Management in the Home Companion

Natural language understanding relies heavily on SISR information (see previous section). The context specific ASR grammars contain tags, which provide logical items to NLU component as part of ASR results. These logical items are either complete predicates, or in most cases, individual parameter values and lists of parameter values. This provides a basis for further input processing, where input is parsed against the current dialogue state to compile full, logical representations compatible with the planning implemented in the Cognitive Model. Input is parsed first against the previous system output. In practise, NLU component tries to unify the logical items into the space of predicates, which match the last system utterance. For example, if the utterance was "Why don't you cycle to work", the possible predicate space for anwers includes acceptance and rejection of the suggestion, specification of an alternative travel method, and an expression of preference for or against the suggested travel method. If user input was "yes, that would be fine", the SISR based item is "ACCEPT-TASK", and the unification results in the complete predicate "ACCEPT-TASK TRAVEL-METHOD CYCLING-TRAVEL HOME WORK".

If the first attempt does not result a full parse, then a new predicate space is built from predicates related to the current topic. In the case of the above example, this would be everything related to the different possibilities when traveling to work. The final parsing option is the parsing against the entire space of known predicates. This set includes generic things like help requests

and enabled topic shifts in this domain. If needed, deeper analysis can be made by parsing against the entire dialogue history and the current plan. This way, the information produced by the CM is used in input parsing. The dialogue plan could also be used in dynamic construction of recognition grammars to support this on ASR grammar level.

In addition to unification of predicates, a reduced set of DAMSL dialogue acts (Core and Allen, 1997) is used to mark functional dialogue acts using rule-based reasoning.

As said, the task of the DM is to maintain and update a dialogue state. In the Home Companion, the dialogue state includes a dialogue history, a stack of active dialogue topics, and current parsed user input, including ASR confidence scores and N-best lists. In addition, two pools of items that need to be confirmed are stored; one for items to be confirmed individually and another for those that can be confirmed together with a single question.

The DM receives user inputs as predicates parsed by the NLU component. The DM is aware of the relations of the predicates on the topics level, i.e., it knows which predicates belong to each topic. This information is used primarily for parsing input in relation to the dialogue context. The DM also has understanding of the semantics of the predicates which relates to interaction. Namely, relations such as question answer pairs (suggestion agreement, confirmation acceptance/rejection, etc.) are modelled by the DM.

If an utterance is successfully parsed and matches the current dialogue plan, the DM does not need to know the the actual meaning of the input. Instead, it takes care of confirmations, when necessary, and provides the information to the CM. Similarly, when generating output requests based on the plan, the DM can be unaware of the specific meaning of the plan items. Overall, the DM does not need to have the deep domain understanding the CM specializes in.

Dialogue management is implemented as a collection of separate compact dialogue agents, each taking care of a specific task. In each dialogue turn one or more agents are selected by the DM, as presented in Section 3.1. In the current Home Companion, there are over 30 dialogue agents. There is a separate agent for each topic that can occur in the plan. In practice, one topic maps to a single planitem. These agents are all instances of a single class with specific configurations. Each agent handles all situations related to its topic; when the topic is the first item of an active plan, the agent produces related output, and when the user provides input matching to the topic it forwards that information back to the Cognitive Model. In addition,

topic specific agents handle explicit topic switch requests from the user (e.g., "*let's talk about lunch*") and also take turns if the topic is found on top of the dialogue topic stack. A topic ends up on the stack if it has not been finished when a new topic is activated. The Dialogue manager processes stack items before returning to dialogue plan items. Currently, we have not encountered any situations where the dialogue stack needs to be flushed. This might be necessary in more complex dialogues.

The generic, non-topic specific agents include one that generates a confirmation if the ASR confidence score is too low, one that repeats the last system utterance when the user requests it ("*eh, come again?*"), and an agent to handle ASR rejection errors. There is no need to model such information in the CM.

4.4. Dialogue Management in the Mobile Companion

In the Mobile Companion, the dialogue with the user is managed by the script code running inside the Java midlet, using a finite state machine model. At any point in time there is one active dialogue state. Each state is represented by four script procedures, enter, leave, input, and error.

A typical use of these procedures in the Mobile Companion is to output a question in the **enter** procedure, and then analyse the user's reply and move on to another state or ask another question in the **input** procedure. The **leave** procedure is used to do state clean-up tasks (if necessary), and the **error** procedure will output an error message and repeat the question.

SISR expression are used in the Mobile Companion as well. The SISR expressions consist of predicates of the form

```
#TASK_SELECTED(walking, home, work)
```

and are used directly by the script code to analyse the input. The user can provide input using voice, by pressing buttons, or tapping on a list on touch screen. However, all user input has the same form, so it does not matter how the input was produced.

Natural language generation is handled by the state procedures directly. There are some canned utterances and some dynamically constructed output, for instance, based on the content of the activity plan downloaded from the Home Companion.

4.5. Benefits of the Model for Conversational Distributed Dialogues

The presented model for interoperability between the Mobile Companion, the DM of the Home Companion, and the CM has provided great flexibility to each component. While the dialogue plan generated by the CM provides a base for dialogue management, which, in most cases, is followed, the DM can deviate from it when feasible. The DM can handle confirmations as it pleases, add small talk, and process the plan items in any order. The model also supports mixed-initiative dialogues; while the DM may follow the plan, the user may discuss any topic.

Most importantly, all this is possible without including domain specific knowledge into DM. All such information is kept exclusive in the CM. Similarly, the CM does not need to know the interaction level properties of the topics, such as recognition grammars and natural language generation details. These are internal to their specific components. The Mobile Companion uses the same knowledge representation as the CM, but the CM does not need to be aware of its existence at all. Similarly, the Mobile Companion can use any part of the information it receives, but is not forced to do anything specific. The DM just feeds all the information to the mobile Companion and lets it decide what to do with the information. When the Mobile Companion provides information back to the Home Companion, the DM handles the access to the information and the CM can ignore completely the fact that different parts of the information it receives were generated using different systems. Similarly, the Mobile Companion does not see any of the internals of the Home Companion.

At the implementation level, the model is independent of the mechanics of either the DM or the CM. The DM can be implemented using state transition networks (a network per plan item), forms (form per item), an agent-based model, like in the case of the Mobile Companion, or any other suitable method. Similarly, the plan does not tie the CM to any specific implementation.

5. Evaluation of Companions

Since Companionship in the sense described in the present paper is a fairly new concept, no agreed-on evaluation strategies of it exist, and one of the goals of the Companions project is to develop such evaluation strategies. In order to gain deep understanding on the Companions paradigm, user studies must be conducted both in laboratory and field settings. Thus an initial set

of user studies of the Health and Fitness Companion prototypes has focused on out-of-the-box functionality to set a baseline for further studies. Two mechanisms have been utilized for the evaluation: qualitative and quantitative. Qualitative surveys are used to acquire subjective opinions from the users; including Likert-based surveys, focus groups and interviews, while the quantitative measures are based on the following set of metrics:

Speech Metrics, such as WER (word error rate) and CER (concept error rate);

Dialogue Metrics, e.g., dialogue duration, number of turns, word per turn dialogue structure;

Task Metrics, e.g., task completion; and

User Metrics, e.g., user satisfaction, requirement elicitation.

The metrics mentioned are quite traditional, and they are included in evaluation frameworks such as PARADISE (Walker et al., 1997). One option would have been to use such a complete evaluation paradigm. However, our aim was a lightweight evaluation at this phase of the project instead of running a comprehensive and laborous evaluation. Furthemore, the existing methods are mainly focusing on task-driven dialogues, and while we use here metrics such as Task Completion rate (see definition below), our tasks are rather loose, so we believe it would not be feasible to try to find similar correlations between objective and subjective metrics as in more more straightforward task-oriented systems.

Two evaluation sessions have been carried out so far. The first evaluation session focused on WER and CER metrics for the Home Companion. The CER is based on correctly identifying the semantic result of a user utterance. For example, correctly identifying whether the user accepted or rejected a proposal from the HFC. If a user utterance contained distinct semantic results, these were then counted as separate concepts when counting the score. For example, if in reply to a proposal to have a rest after dinner the user replies "No, I'll play a game of football." the system would have to identify both a rejection of the HFC's proposal and a counter-proposal of playing football. Since we wanted to find out a baseline for further work, we did not include any clarification dialogues to this evaluation. Furthermore, there

were no issues such as grounding taking place in the dialogues. For these reasons, there was no need to model these when calculating CER.

Another metric calculated is Task Completion rate, which in the case of Health and Fitness companion is specified as the percentage of the activity model correctly instantiated. That is, it provides the percentage similarity between the activity model as instantiated and the instantiation of an ideal activity model with all utterances optimally understood.

The study involved 20 subjects who interacted with the Home Companion in both planning and reporting phases. They were briefly introduced to the concept of the Companion and the scenario, and provided with a set of slides illustrating (via images of activities and food types) what was known to the system. To avoid bias, the subjects were not shown examples of possible utterances nor allowed to witness experiments with other subjects.

For the planning dialogues, the WER was 42% and CER 24%. For the reporting dialogues, the WER was 44% and CER 24%. The fact that no user training or speaker adaptation was carried out, along with the realistic experimental conditions, explains the high Word Error Rate level. The Concept Error Rate is lower, indicating some resilience to misrecognition of portions of the user utterance. Still, Task Completion was high, with, on average, 80% of the Activity Model being correctly instantiated in planning dialogues and 95% being correctly instantiated in reporting dialogues. This is better than the corresponding per utterance CER as the system was able to eliminate some errors over the course of the entire dialogue.

The significant improvement of CER over WER is due to the reasonable robustness in identifying the semantic meaning of the user's utterances. This is partially due to the semantic results being constrained to a relatively low number of possibilities compared to the reasonably large vocabulary user input to the ASR. For example, in response to the HFC asking "How about bringing a packed lunch to work?" the user responds "Yes, I will make a sandwich". When this is recognised as "yes I love that sandwiches" this results in a WER of 50% but the system is still able to identify the concepts (agreement to the proposal) correctly. There is a similar degree of robustness when instantiating the activity model. For example, the HFC can infer from a counter-proposal that the original proposal was rejected despite potentially missing out on an explicit rejection.

Also worth noting is that reporting dialogues tended to involve simpler user utterances, such as basic confirmations, than those in the planning phase, which was reflected by smaller average utterance length and higher

task model completion rate. The initial results show that even with relatively high WER acceptable task completion rates can be obtained in this domain.

In the second evaluation session, the Home Companion and the Mobile Companion were evaluated in more a complex setting. Eight participants completed a study protocol of four distinct tasks: introductory tutorials, using prototypes, on-line surveys, and interviews. Each session began with an introductory tutorial. These presentations (10–16 slides) introduced the prototype, established its intentions, its limitations, what the prototype would say and do, how to use the prototype, and gave the user suggestions in how to respond. Participants then used the Home and Mobile Companion for about 10–15 minutes each, completing an on-line questionnaire after each session. Researchers were sitting in the background while the participants interacted with the prototypes, and the participants were video-taped during their interaction with the systems. At the end of the session, each participant was interviewed. In this study, the Home Companion language model was more complex than in the previous session evaluation. The Word Error Rate ranged between 51% and 79% (however, it must be noted that in one case the error rate was over 100% because of massive amounts of rejection errors). While the Word Error Rate was extremely high, the Concept Error Rate (calculated by ignoring the order of recognized concepts) was somewhat lower, ranging between 33% and 65%. These numbers are still high, but most of the time the Concept Error Rate was reasonably good, with the majority of the errors being insertions of several concepts in some specific cases. The average length of user utterances varied between participants from 3.0 to 8.3 words for the Home Companion, while the average system utterance length was 12 words. Even though the data set is too small for statistical testing, we could see differences in how verbose different people were. The user vocabulary size was surprisingly small, and only varied between 18 and 116 words, with an average of 55 words.

Future plans for the evaluation efforts include to target areas of performance required in long-term collaborative conversational agents. These are discussed in more detail in (Benyon et al., 2008).

6. Summary

In this paper, we have introduced the concept of multimodal Companions that build long-lasting relationships with their owners to support their everyday health and fitness related activities. While traditional spoken dialogue

systems have been task-based, the Health and Fitness Companion is designed to be part of the user's life for a long time, months, or even years. This requires that they are part of the life physically, i.e., that interactions can take place in mobile settings and in home environments outside of traditional, task-based computing devices. With the physical presence of the interface agents, and spoken, conversational dialogue we aim at building social, emotional relationships between the user and the Companion. Such relationships should help us in motivating the users towards a healthier lifestyle. The mobility of the interface integrates the system into the physical activities it aims at supporting users in.

We have shown what kind of architectures such distributed agent interfaces need, and how they can be realized with proper dialogue and cognitive management techniques. When dialogue systems move beyond limited task-based domains and implement multimodal interfaces in pervasive computing environments, complexity increases rapidly. Dialogue management, which in most cases is handled with well-understood methods such as form filling or state transition networks, tends to grow more complex. Therefore, a model to modularize dialogue management and domain reasoning is needed. At the same time, distributed systems require various kinds of information to be communicated between components and systems.

We have presented a novel interaction management model, which separates cognitive modeling from dialogue management and enables flexible interoperability between these two. This model also enables sharing the gathered knowledge to the mobile part of the system and back. This division, while similar to separation of a back-end from dialogue management, draws the line deeper into the area of interaction management. The presented model has been implemented in the Health and Fitness Companion prototype, and it has enabled the Cognitive Model, the Dialogue Manager, and the Mobile Companion interface to be developed in parallel by different groups using various programming languages.

On software architecture level, the solution has enabled the Dialogue Manager to focus only on interaction level phenomena, such as initiative and error management, and other meta-communication while the congitive model takes care of the domain level processing. The Dialogue manager can also include input from a mobile interface of the system without making this explicit to the cognitive model. One example of flexibility is error management; while the actual error correction is the task of the Dialogue Manager, domain level knowledge can reveal errors. Using the dialogue plan, the cog-

nitive model can provide such information to the Dialogue Manager without knowledge of details of error management. The model also enables user initiative topic shifts, management of user-system relationship, and other such issues relevant in domain-oriented dialogue systems.

In overall, the model presented has enabled a clear division and interoperability of the different components handling separate parts of the interaction. We can recommend the model for other applications, where complex domain modeling is used in a situation where error correction and other dialogue managements task must be handled as well. Using the model, the different component can each focus on their own tasks.

In addition to the interaction model presented, we have produced concrete software components to implement further Companion applications. The PART system used to implement the Mobile Companion and the Jaspis architecture (including the jNabServer software) used for the Homa Companion have been released as open source software. Together, they can be used to construct similar distributed multimodal applications with virtual, physical, and mobile conversational agents.

Finally, we have introduced the first evaluation results of the Companions paradigm. These initial results show that even with relatively low speech recognition accuracy meaningful conversations can be carried out in this domain.

Acknowledgements

This work was funded by the COMPANIONS project [9] sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-034434.

References

Benyon, D., Hansen, P., Webb, N., Oct. 2008. Evaluating human-computer conversation in Companions. In: Proceedings of the 4th International Workshop on Human-Computer Conversation. Bellagio, Italy, pp. 1–5.

Bickmore, T. W., Picard, R. W., Jun. 2005. Establishing and maintaining long-term human-computer relationships. ACM Transactions on Computer-Human Interaction 12 (2), 293–327.

[9] www.companions-project.org

Buttussi, F., Chittaro, L., 2008. MOPET: A context-aware and user-adaptive wearable system for fitness training. Artificial Intelligence in Medicine 42 (2), 153–163.

Cavazza, M., Smith, C., Charlton, D., Zhang, L., Turunen, M., Hakulinen, J., May 2008. A 'Companion' ECA with planning and activity modelling. In: Padgham et al. (2008), pp. 1281–1284.

Core, M. G., Allen, J. F., Nov. 1997. Coding dialogs with the DAMSL annotation scheme. In: AAAI Fall Symposium on Communicative Action in Humans and Machines. Cambridge, Massachusetts, pp. 28–35.

de Oliveira, R., Oliver, N., Sep. 2008. TripleBeat: Enhancing exercise performance with persuasion. In: Proceedings of 10th International Conference on Mobile Human-Computer Interaction. ACM, Amsterdam, the Netherlands, pp. 255–264.

Dybkjaer, L., Bernsen, N. O., Minker, W., 2004. Evaluation and usability of multimodal spoken language dialogue systems. Speech Communication 43 (1-2), 33–54.

Hakulinen, J., Turunen, M., Salonen, E.-P., 2007. Visualization of Spoken Dialogue Systems for Demonstration, Debugging and Tutoring. In: Proceedings of Interspeech 2005, pp. 853-856.

Hakulinen, J., Turunen, M., Smith, C., Cavazza, M., Charlton, D., Oct. 2008. A model for flexible interoperability between dialogue management and domain reasoning for conversational spoken dialogue systems. In: Proceedings of the 4th International Workshop on Human-Computer Conversation. Bellagio, Italy, pp. 29–34.

Hernández, A., López, B., Pardo, D., Santos, R., Hernández, L., Relaño, J., Rodríguez, M. C., May 2008. Modular definition of multimodal ECA communication acts to improve dialogue robustness and depth of intention. In: Padgham et al. (2008), pp. 10–17, 1st Functional Markup Language Workshop.

Jönsson, A., 1991. A natural language shell and tools for customizing the dialogue in natural language interfaces. Research Report LiTH-IDA-R-91-10, Dept. of Computer and Information Science, Linköping University, Linköping, Sweden.

Jullien, C., Marty, J.-C., Apr. 1989. Plan revision in person-machine dialogue. In: Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics. ACL, University of Manchester Institute of Science and Technology, Manchester, England, pp. 153–160.

Kadous, M. W., Sammut, C., 2004. InCa: A mobile conversational agent. In: Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence. Auckland, New Zealand, pp. 644–653.

Kainulainen, A., Turunen, M., Hakulinen, J., Salonen, E.-P., Prusi, P., Helin, L., 2005. A speech-based and auditory ubiquitous office environment. In: Proceedings of 10th International Conference on Speech and Computer. Patras, Greece, pp. 231–234.

Larsson, S., Traum, D., 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. Natural Language Engineering Special Issue on Best Practice in Spoken Language Dialogue Systems Engineering, 323–340.

López, B., Hernández, A., Díaz, D., Fernández, R., Hernández, L., Torre, D., Jun. 2007. Design and validation of ECA gestures to improve dialogue system robustness. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. ACL, Prague, Czech Republic, pp. 67–74, workshop on Embodied Language Processing.

López, B., Hernández, A., Pardo, D., Santos, R., del Carmen Rodríguez, M., Apr. 2008. ECA gesture strategies for robust SLDS. In: AISB Annual Convention: Communication, Interaction and Social Intelligence. Vol. 10. SSAISB, Aberdeen, Scotland, pp. 69–76, Symposium on Multimodal Output Generation.

López Mencía, B., Hernández Trapote, A., Díaz Pardo de Vera, D., Torre Toledano, D., Hernández Gómez, L. A., López Gonzalo, E., Nov. 2006. A good gesture: Exploring nonverbal communication for robust SLDSs. In: IV Jornadas en Tecnología del Habla. Zaragoza, Spain, pp. 39–44.

Loquendo, 2008. Loquendo embedded technologies: Text to speech and automatic speech recognition.
URL `www.loquendo.com/en/brochure/Embedded.pdf`

Marti, S., Schmandt, C., Oct. 2005. Physical embodiments for mobile communication agents. In: Proceedings of the 8th Annual Symposium on User Interface Software and Technology. ACM, Seattle, Washington, pp. 231–240.

Oliver, N., Flores-Mangas, F., Sep. 2006. MPTrain: A mobile, music and physiology-based personal trainer. In: Proceedings of 8th International Conference, on Mobile Human-Computer Interaction. ACM, Espoo, Finland, pp. 21–28.

O'Neill, I., Hanna, P., Liu, X., McTear, M., Sep. 2003. The Queen's Communicator: An object-oriented dialogue manager. In: Proceedings of the 8th European Conference on Speech Communication and Technology. ISCA, Geneva, Switzerland, pp. 593–596.

Padgham, Parkes, Müller, Parsons (Eds.), May 2008. Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems. IFAAMAS, Estoril, Portugal.

Pellom, B., Ward, W., Pradhan, S., Oct. 2000. The CU Communicator: An architecture for dialogue systems. In: Proceedings of the 6th International Conference on Spoken Language Processing. Vol. 2. Beijing, China, pp. 723–726.

Salonen, E.-P., Hartikainen, M., Turunen, M., Hakulinen, J., Funk, A. J., Oct. 2004. Flexible dialogue management using distributed and dynamic dialogue control. In: Proceedings of the 8th International Conference on Spoken Language Processing. Jeju, Korea, pp. 197–200.

Smith, C., Cavazza, M., Charlton, D., Zhang, L., Turunen, M., Hakulinen, J., 2008. Integrating planning and dialogue in a lifestyle agent. In: Intelligent Virtual Agents. Springer, Berlin, Germany, pp. 146–153.

Ståhl, O., Gambäck, B., Hansen, P., Turunen, M., Hakulinen, J., Oct. 2008. A mobile fitness Companion. In: Proceedings of the 4th International Workshop on Human-Computer Conversation. Bellagio, Italy, pp. 43–48.

Ståhl, O., Gambäck, B., Turunen, M., Hakulinen, J., Mar. 2009. A mobile health and fitness Companion demonstrator. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. ACL, Athens, Greece, pp. 65–68, demonstrations session.

Turunen, M. Jaspis - A Spoken Dialogue Architecture and its Applications. PhD dissertation, University of Tampere, Department of Computer Sciences A-2004-2, February 2004.

Turunen, M., Hakulinen, J., Räihä, K.-J., Salonen, E.-P., Kainulainen, A., Prusi, P., 2005. An architecture and applications for speech-based accessibility systems. IBM Systems Journal 44 (3), 485–504.

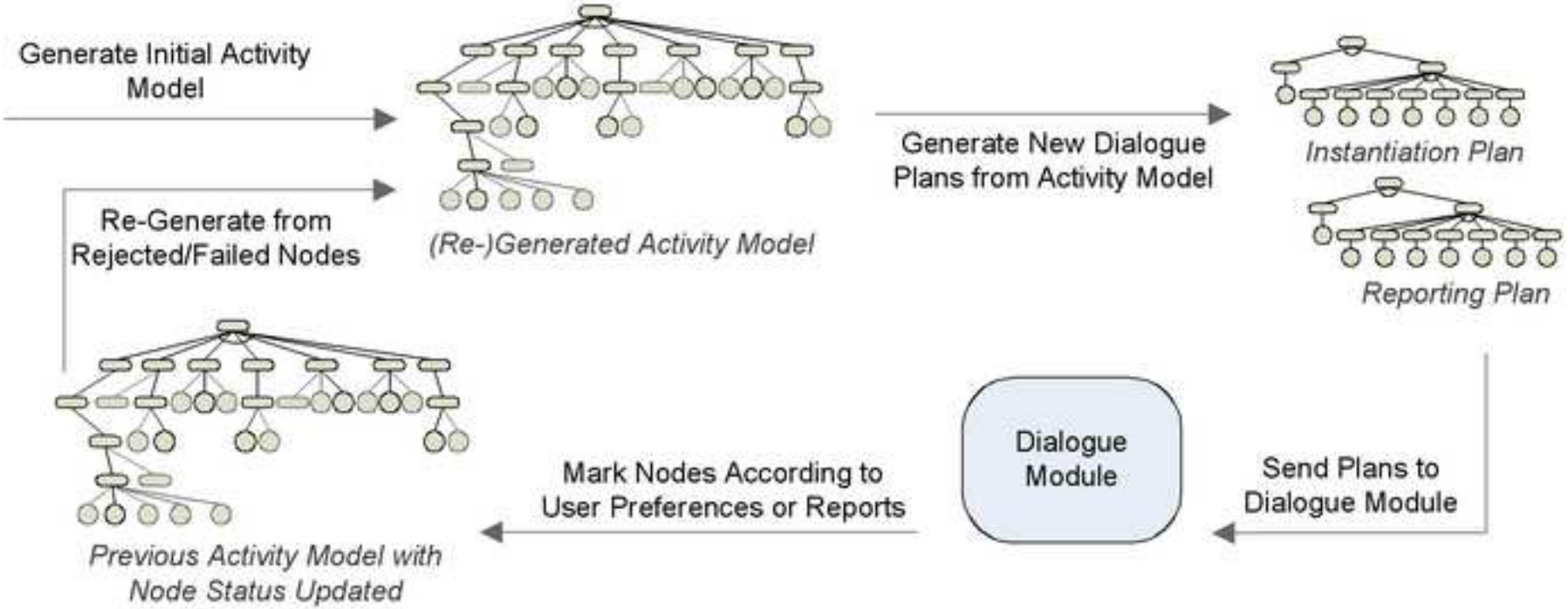
Turunen, M., Hakulinen, J., Smith, C., Charlton, D., Zhang, L., Cavazza, M., Sep. 2008a. Physically embodied conversational agents as health and fitness Companions. In: Proceedings of the 9th Annual INTERSPEECH Conference. ISCA, Brisbane, Australia, pp. 2466–2469.

Turunen, M., Hakulinen, J., Ståhl, O., Gambäck, B., Hansen, P., Gancedo, M. C. R., de la Camara, R. S., Smith, C., Charlton, D., Cavazza, M., Oct. 2008b. Multimodal agent interfaces and system architectures for health and fitness Companions. In: Proceedings of the 4th International Workshop on Human-Computer Conversation. Bellagio, Italy, pp. 49–54.

Walker, M., Litman, D., Kamm, C., Abella, A., 1997. PARADISE: a framework for evaluating spoken dialogue agents. In: Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics, pp. 271 - 280.

Wilks, Y., Nov. 2007. Is there progress on talking sensibly to machines? Science 318 (9), 927–928.

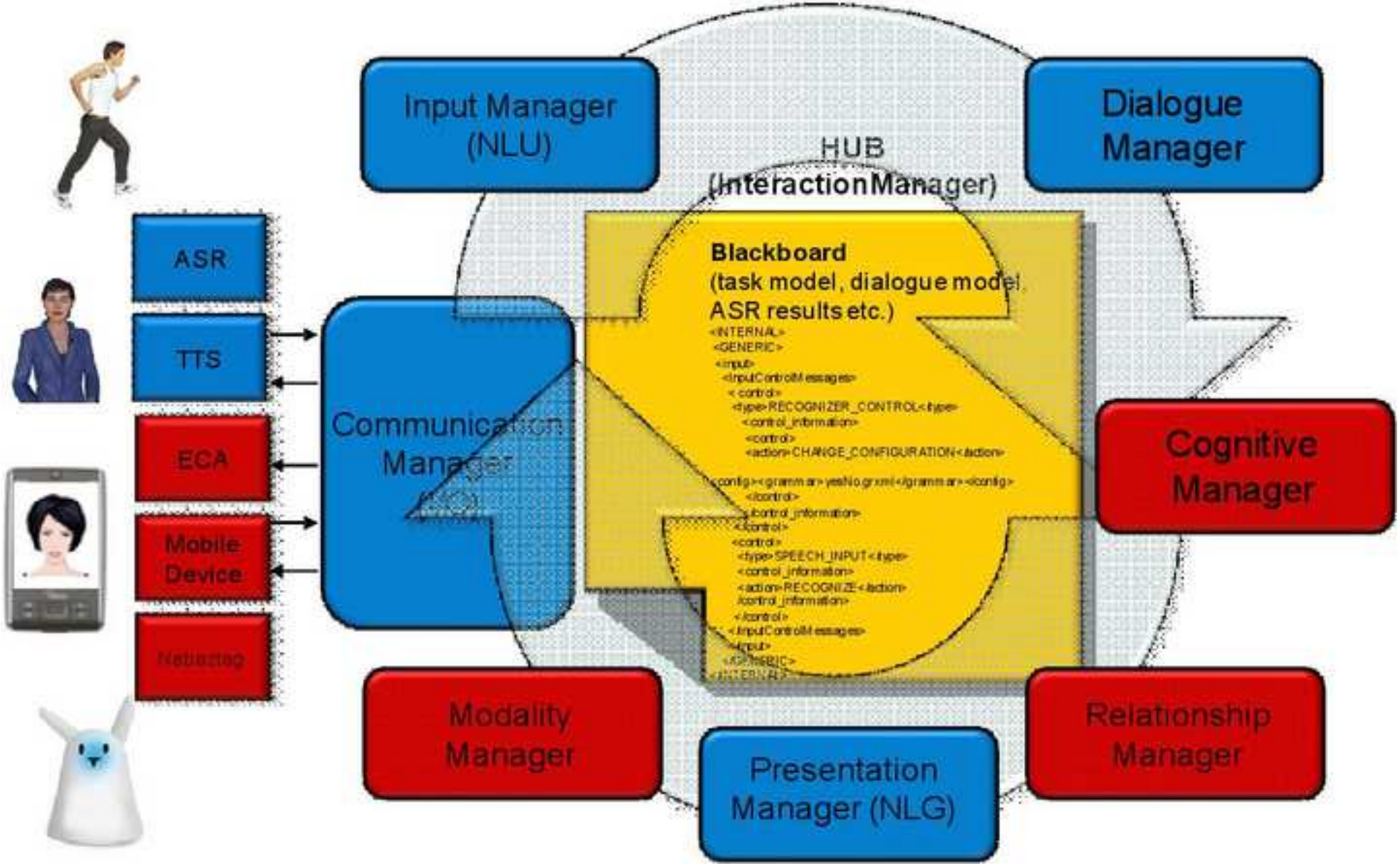
Figure


Generate Initial Activity Model
(Re-)Generated Activity Model
Generate New Dialogue Plans from Activity Model
Instantiation Plan
Reporting Plan
Re-Generate from Rejected/Failed Nodes
Previous Activity Model with Node Status Updated
Mark Nodes According to User Preferences or Reports
Dialogue Module
Send Plans to Dialogue Module

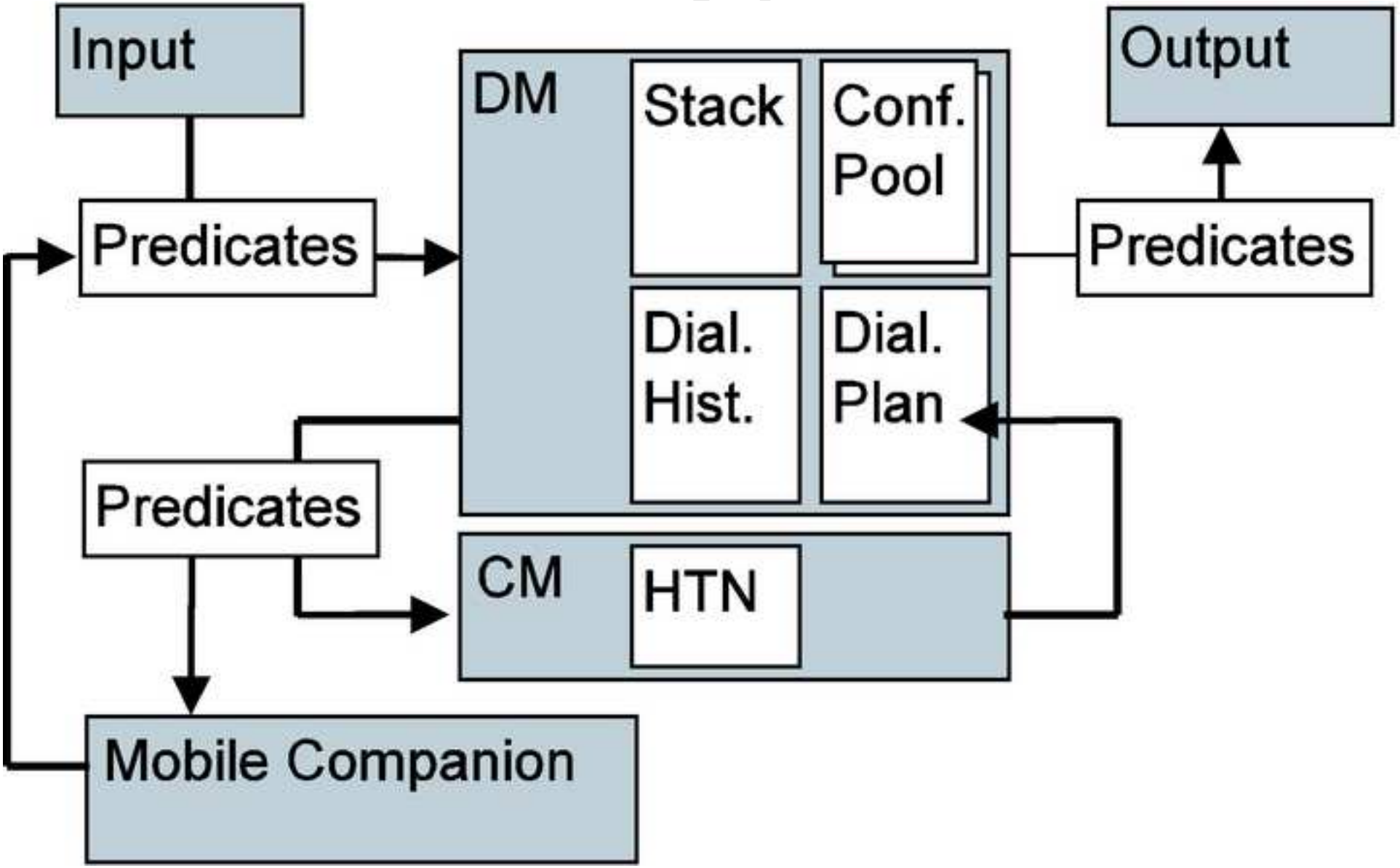
Figure


Paella
Steps:
Step 3
Add to this your saffron water, squid, chili and pimento and season with salt and pepper. Give it a good stir...
Cooking
companions

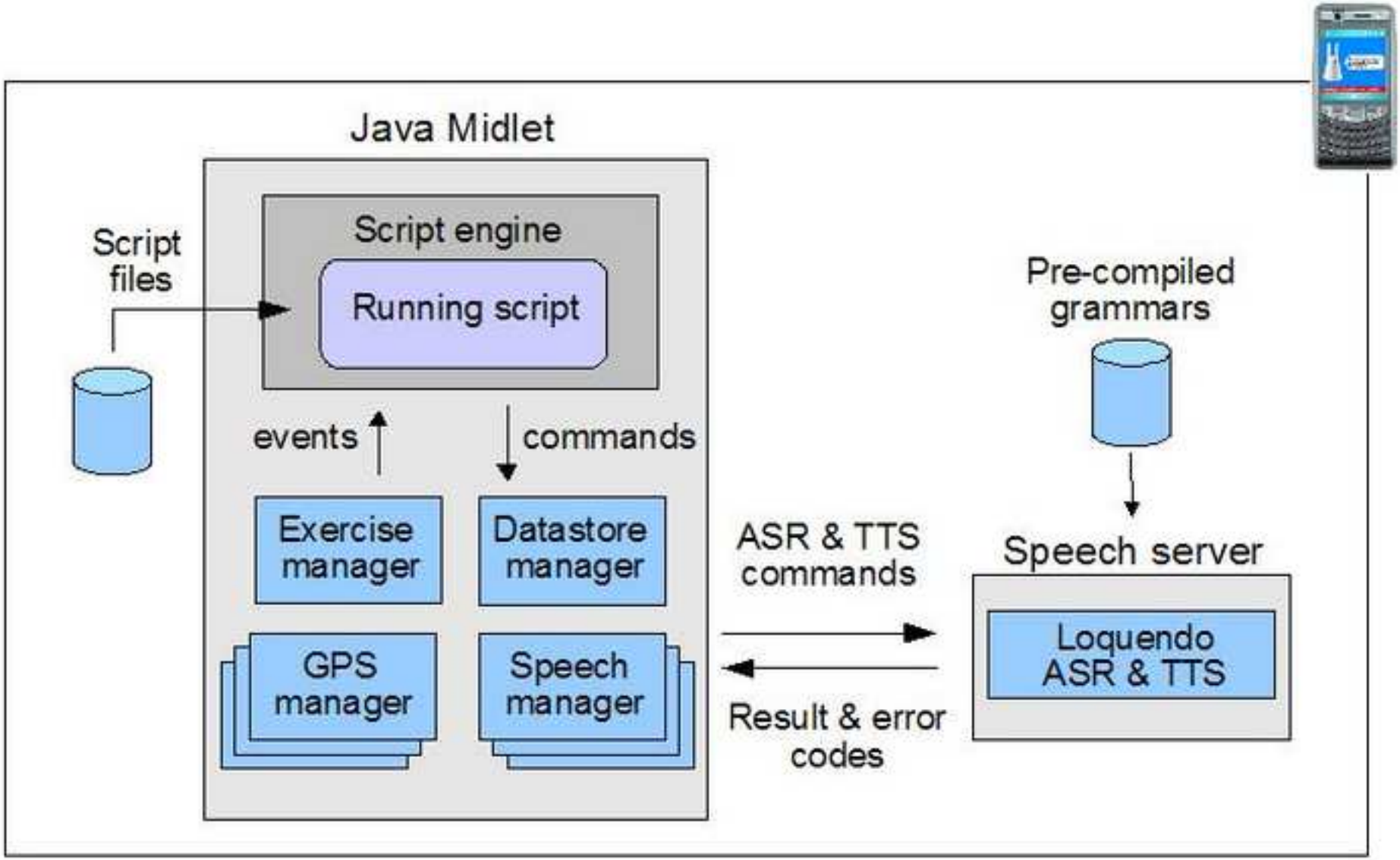
Figure

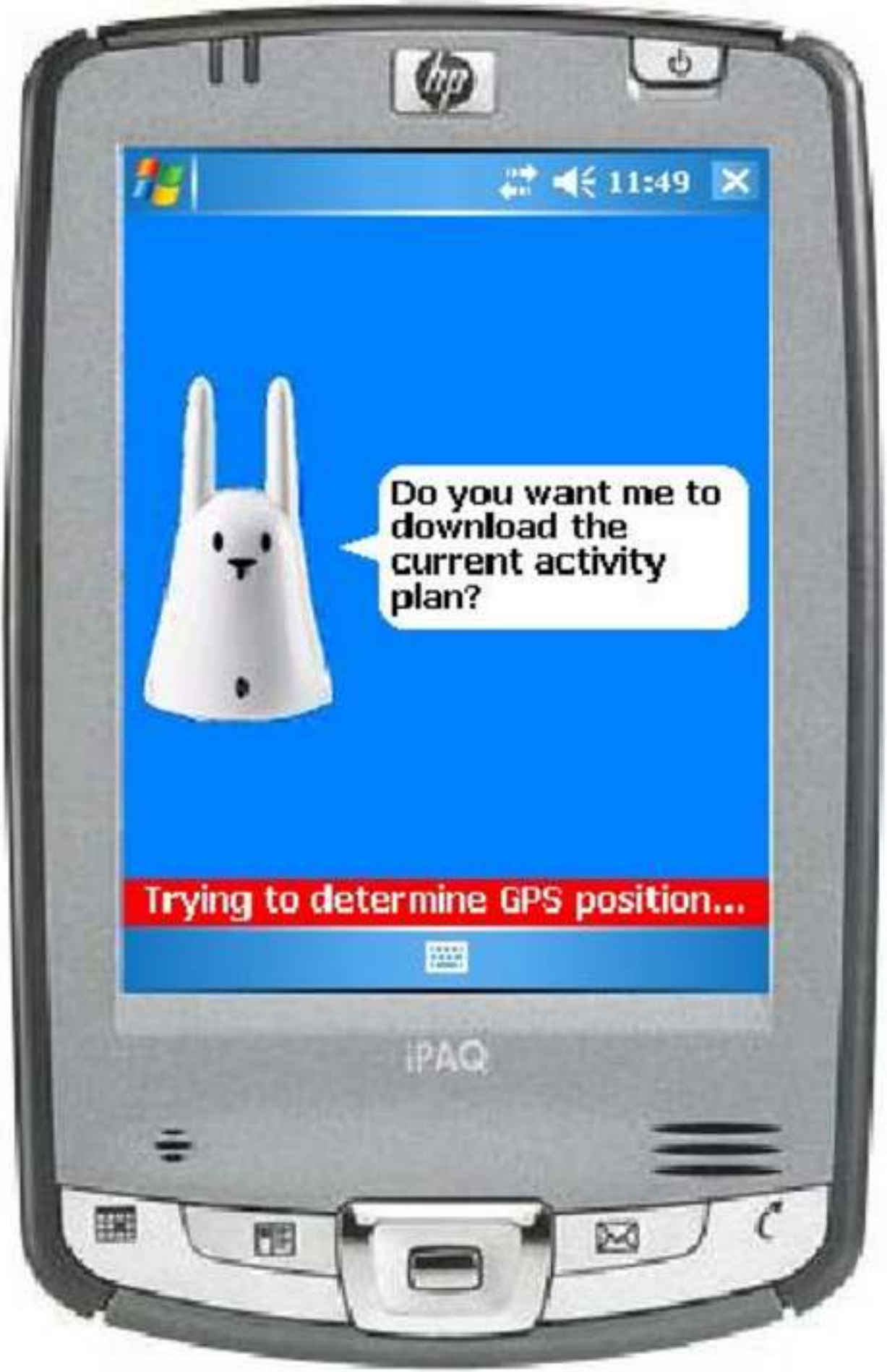

Input Manager (NLU)
HUB (InteractionManager)
Dialogue Manager
Blackboard
(task model, dialogue model, ASR results etc.)
ASR
TTS
ECA
Mobile Device
Nabaztag
Communication Manager
Cognitive Manager
Modality Manager
Presentation Manager (NLG)
Relationship Manager

Figure

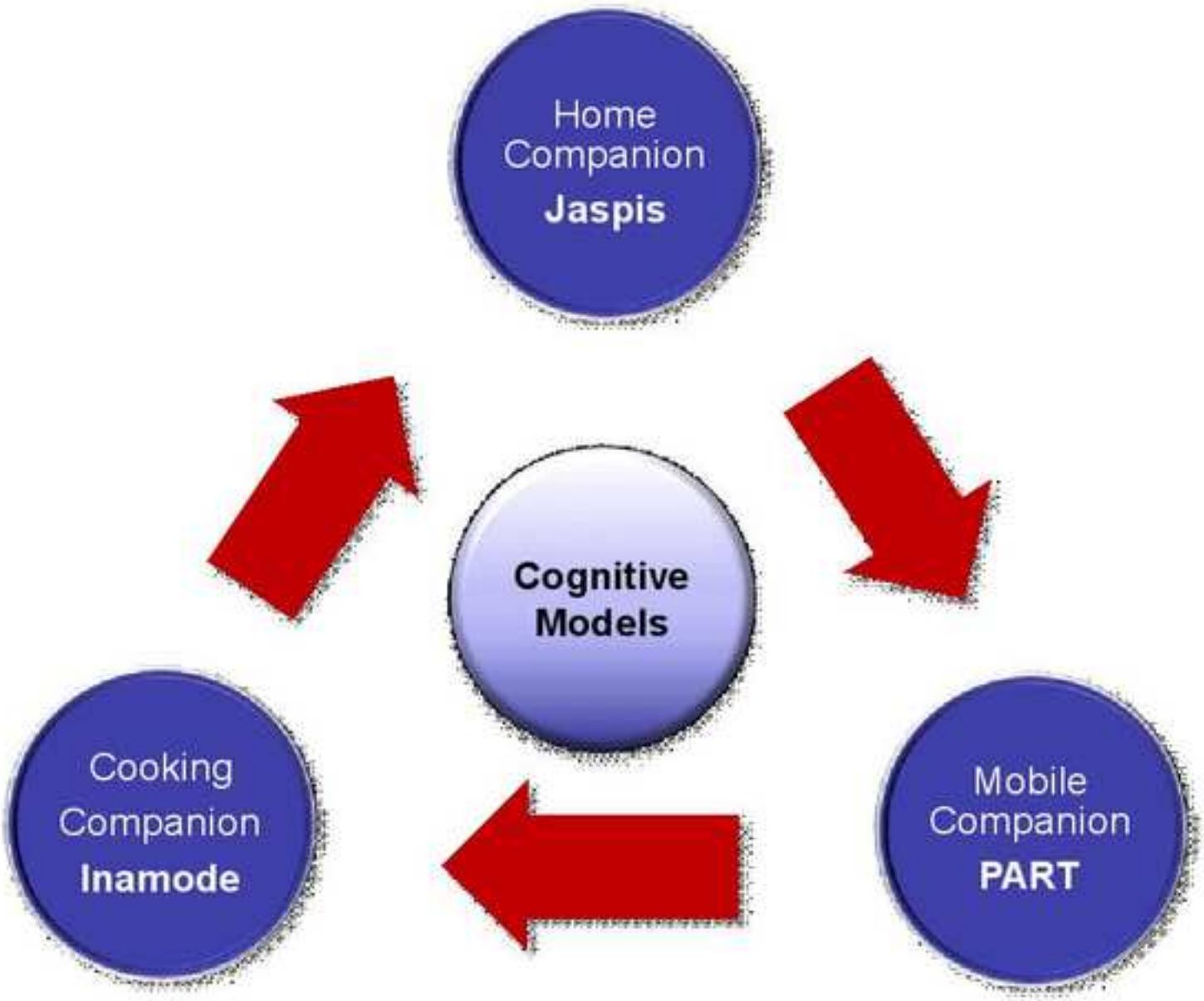

Input
Output
DM
Stack
Conf.
Pool
Predicates
Predicates
Dial.
Hist.
Dial.
Plan
Predicates
CM
HTN
Mobile Companion

Figure


Java Midlet
Script files
Script engine
Running script
events
commands
Exercise manager
Datastore manager
GPS manager
Speech manager
ASR & TTS commands
Result & error codes
Pre-compiled grammars
Speech server
Loquendo ASR & TTS



Figure


Home
Companion
Jaspis
Cognitive
Models
Cooking
Companion
Inamode
Mobile
Companion
PART

Figure