A Personal Medical Digital Assistant Agent for Supporting Human Operators in Emergency Scenarios

Angelo Croatti, Sara Montagna, and Alessandro Ricci

DISI, University of Bologna, Via Sacchi 3 – Cesena, Italy
emails: {a.croatti|sara.montagna|a.ricci}@unibo.it

Abstract. Nowadays, the impressive development of mobile and wearable/eyewear computing technologies allows for designing novel kind of Personal Medical Digital Assistant Agents (PMDA) supporting healthcare professionals in their individual and cooperative activities, using hands-free and on-the-go interfaces. In this paper, after discussing some general aspects of this kind of personal agents, we describe a concrete case study in healthcare, about *trauma documentation* and management in Emergency Department—a project developed in cooperation with the Trauma Center of the "M. Bufalini" Hospital in Cesena, Italy.

1 Introduction

In the last decade, information and communication technologies (ICT) witnessed an impressive progress, in particular mobile and *wearable* ones, making the visions about pervasive computing in hospitals [2,22,36] more and more a reality. Nowadays Personal Digital Assistants (PDA) and tablets are widely deployed in various healthcare contexts [14]. Modern smartphones are powerful computing devices, featuring a variety of onboard sensors (camera, GPS, NFC reader,...), a robust support for pervasive interaction with an ecosystem of Bluetooth-enabled external devices and wireless networking. This makes it possible to design complex mobile computing applications, eventually interacting with services in local area networks and on the Internet/cloud.

Besides mobile computing, technologies for *wearable computing* [16] and *eyewear computing* [5] are achieving a level of maturity that makes it possible to exploit them out of labs, in real-world professional contexts. In particular, smart-glass technologies – e.g., Vuzix m100, Epson Moverio BT-200, *Microsoft Hololens* – allow to designing a new generation of (pervasive) software systems exploiting different degrees of Mixed and Augmented Reality [31]. These devices are basic bricks to realise *hands-free* or *use-on-the-go* systems [29,30], in which users can, e.g., asynchronously perceive information, data generated by the application without the need of changing the focus of their current activity and limiting as much as possible the use of hands to act/interact with the device.

The development of these technologies allows for devising new kind of *software personal agents*, assisting healthcare professionals in doing their job. In this paper we refer to this agent technology as Personal Medical Digital Assistant Agents (PMDAs). A main healthcare context where this kind of technology can be useful is the *emergency*. In this context, agent technologies have been already proposed e.g. for emergency coordination. In this paper, we present and discuss a further novel case, concerning the development of a PMDA for trauma documentation and management. The project – called TraumaTracker – is being developed in cooperation with the Emergency Department of a Hospital in Italy. The first prototype of TraumaTracker has been implemented using agent technologies—a version the JaCaMo multi-agent oriented programming platform running on Android-based mobile and wearable devices.

TraumaTracker has been designed in a modular way, to support increasing levels of functionalities and services. The base level concerns tracking events and data, for documentation purposes—which is the focus of this paper. Upper levels concern functionalities more oriented to real-time assistance, such as the generation of warnings—these are part of ongoing and future work. A first validation of the system has been carried on by the trauma team, remarking both the benefits with respect to the current practice, and current limits, providing feedbacks for further development of the system.

The remainder of the paper is organized as follows. In Section 2 we provide a background on the specific context – related to emergency and, in particular, trauma documentation and management – and an overview of existing related work about agent technologies applied to emergency. After that, in Section 3 we discuss the general characteristics of PMDAs, as novel kind of software personal agents, and then we go into the core contribution of the paper, i.e. the description of TraumaTracker—first describing its design and then its prototype implementation, discussing its first evaluation. In Section 6 we conclude the paper discussing current limits and future works.

2 Background and Related Work: Agents in Heathcare Emergency Scenarios and The Trauma Documentation Problem

The unquestioned benefit of the introduction of ICT into healthcare systems is already recognised worldwide since ICT successfully addresses the vast set of characteristics and situations proper of the healthcare scenario —such as mobility, time-critical, distribution and large-scale coordination, context-awareness, decision-making, interoperability, complexity. Last-generation ICT infrastructures and services, especially the emergence of wearable and mobile technologies, opened new frontiers in healthcare by efficaciously supplying the work of hospital staff, doctors, and patients. The so-called *e-Health* [40] and *m-Health* [28] improved the quality of health-services, providing technologies for different purposes, such as acquiring and sharing patient data through Electronic Medical Records (EMR), automating administrative health-related processes, providing telemedicine services, remote and mobile monitoring, and much more [38,23].

In particular, the adoption of the agent paradigm seems to be particularly suited to improve the performance of an ICT infrastructure in terms of interoperability, scalability and reconfigurability. Literature refers a wide range of applications of the agent framework in e-Health for different purposes. A comprehensive review is provided in [12], where the main categories of applications identified are: *(i)* Medical data management: accessing, integrating and sharing patient' data from different remote sources is crucial for easing the work of physicians and for statistical analysis purposes [37,23]; *(ii)* Decision support systems: supporting physicians in their fast-paced work can reduce human errors, and safe time [6]; *(iii)* Planning and resource allocation: scheduling decisions on the allocation of professional and physical resources must be coordinated by planning techniques [34]; *(iv)* Remote care: mainly devoted to remote patient monitoring, it allows on one side patients with reduced mobility to not travel towards healthcare facilities for vital signs check-up, on the other side physicians to observe the dynamic of the patient's health and provide opportune recommendations tailored to the patient [32]. Moreover, the literature proposes the development of agent systems for *Chronic diseases management*, such as diabetes, respiratory illnesses, and cardiovascular diseases [27]. The goal is to develop applications enabling a shift of the control of chronic illness from the caregiver to the patients themselves, namely the *self-management* of chronic diseases [19]. In this context an agent-based platform enables: *healthcare professionals* to be continuously updated on the patient's health by receiving data such as vital signs measures decreasing the occasions for patients to travel to health facilities; *patients* to be supported in daily decisions by instructions delivered by the application that is based on the elaboration of such data.

In this paper, we discuss specific issues related to the Emergency Department (ED), that is one of the most critical and challenging hospital departments, since it requires reactivity, quick and coordinated response, fast-paced and accurate decision-making. In this scenario, there are three key issues that may be tackled by the agent technology. First of all, the fast-paced sequence of events during a trauma resuscitation leaves little time for physicians to reason about the best treatment and care. A PMDA can thus support human operators by autonomously providing suggestions on the best choice. This would reduce human errors while saving time and increasing team performance. Secondly, an accurate documentation of trauma resuscitation seems to be crucial to improve the quality of trauma care where, according to [8] "Quality of trauma care can be defined as achieving the best possible outcome for a given set of clinical circumstances". Third, agent technologies – in particular Multi-Agent Systems – can be exploited to support the coordination among the various actors involved in the trauma management. An example is presented in [10], about the Ubimedic2 agent framework for supporting rescue operations. [23] presents CASCOM, a distributed multi-agent system for the execution of smart emergencies by providing efficient remote healthcare in case of unexpected events. Within the platform distributed data and information can be retrieved and make available to physicians everywhere, thus enabling easier and faster choices.

In this paper, we specifically focus on the second issue. The documentation of a trauma, known in the literature as *trauma documentation*, should be acquired *during* the process of trauma resuscitation and should report where and when crucial events occurred, which and when treatments are given, procedures are performed, and finally it should report repeated vital signs measures. It is crucial because it is used to make the most informed choice for patient medication and management, and later to evaluate the work of team members by producing data and statistics such as: time of team activation, primary assessment, arrival time of attending physician.

However, producing an accurate documentation in the context of trauma resuscitation is a challenging task [26]. First of all, trauma resuscitation is a fast-paced process, and very few time is left for documenting the process whilst some of the data to be documented are instead time-consuming. Secondly, multiple events happen simultaneously. To treat severe injuries, potentially life-threatening, team members perform concurrent tasks and parallel activities. Monitoring all of them is not trivial. Finally, the person in charge of documenting is often multitasking, his/her resources are not completely dedicated to the documentation task but he/she also performs other activities.

Nowadays most of the EDs adopt *handwritten paper records and flow sheets* for acquiring data [26,20]. The process of data acquisition is mainly conducted during the trauma resuscitation – or sometimes immediately after by collective memory and verbal communication – by the *recorder*, a person with that specific function. Papers are then sent to central bureau where data are manually entered from the sheets into a computerised databank. The overall procedure produces incomplete or even wrong documentation for two main reasons: *(i)* data acquisition is often inaccurate and crucial data are lacking. This is due to several reasons, some of which are due to the intrinsic characteristics of the context, as cited above: multitasking of the person in charge of acquiring data, parallel activities of the different members of the team, multiple data to be recorded, retrospective documenting from collective memory; *(ii)* manual transfer of data from paper to electronic format can introduce oversights. Furthermore, it is expansive in terms of time spent to complete the overall procedure and of workload. Exploiting the context-awareness capability of a PMDA can support human operators in producing accurate trauma documentation.

3 Personal Medical Digital Assistant Agents based on Wearables

3.1 Software Personal Agents for Eyewear Computing

Personal assistants are a well-known application of software agents [17,15,21]. Existing proposal and technologies have been developed for different kind of purposes and capabilities, from scheduling joint activities [18,39,1], monitoring and reminding users of key timepoints [7,35], sharing information, assisting in negotiation decision support [13]. Generally speaking, with Software Personal Agents (SPAs) the user is engaged in a cooperative process in which human and computer agents both initiate communication, monitor events, and perform tasks [15]. This is based on the metaphor that of a *personal assistant* who is *collaborating with the user* in the same work environment [15].

Traditional SPAs are typically deployed on desktop computers or mobile devices such as smartphone. The availability of wearable devices such as smart-glasses discloses further interesting perspectives about designing SPA for applications with an *hands-free* interface [29,30]. Handsfree systems and *on-the-go* interfaces are designed to avoid as much as possible distracting the user from what she is doing – if not desired – providing an information and suggestion flow which is seamlessly perceived and exploited by the user in her activity. A SPA in this case is not really perceived as a separate entity to interact with, but it essentially disappears being perceived as *an extension of the self*, whose perceptions, beliefs, and possibly goals could be thought be an extension of the user's ones. This perspective goes toward the design of interface agents that – in spite of agent autonomy – should make the user always "feel in control" [11].

Among the others, a main feature that could characterise this kind of personal agents would the capability to "see what the user sees" – by means of the camera on the eyewear device – and, more generally, to know what the user is perceiving about her context, given the sensors equipped with the device(s) worn by the user. This allows to frame a kind of pro-active assistance in which SPAs reason not only about the context of the user, but about what the user is perceiving – and not perceiving – from that context, what she is looking at, etc.

The definition and development of this kind of personal agents introduce interesting new research challenges for the agent community, partially tackled in related research context about eyewear computing [5], cognition-aware computing [4], activity-based computing [33] and context-aware computing [9]. These agents are meant to build dynamically a model about what the user is perceiving, and use this knowledge along with the information about user's goals, the state of ongoing activity, the actual state of the physical environment, to provide a pro-active and smart assistance, possibly anticipating and notifying problems and suggesting actions to do.

3.2 Personal Medical Digital Assistant Agents based on Wearables

We are interested in exploring the design and development of SPA that assists health-care professionals in their individual and cooperative work inside the hospital. In the following, we refer to this kind of SPA as *Intelligent Personal Medical Digital Assistant* (PMDA), following the terminology already introduced in the literature.

Generally speaking, the detailed design of a PMDA depends on the specific role of the assisted professional – nurses, doctors, rescuers, etc. – each one providing specific requirements and challenges. In spite of this, it is possible to devise some basic functionalities and capabilities that are useful in general for PMDAs.

- Identification of the context where the user is acting and of the elements that can be relevant for the user's activity. Examples: identifying the room where the user currently is, the patient who is currently target of the activities, etc.
- The (anytime/anyware) capability of retrieving and presenting relevant data for the user's activity, by interacting with the hospital information system (HIS) and devices. For instance: retrieving the information stored in the Electronic Medial Record, or current value of vital signs.
- Notification of messages to the user. Examples include e.g., messages sent by other colleagues (SMS, email, ad hoc...), warning automatically generated by the system about some situation.
- Support for taking note and track relevant events in terms of photo, video, audio taken through the camera/microphone available on the smart-glasses.
- Support for setting up remote audio-video communication. For instance: requesting the assistance of a colleague.

A key aspect of the system is the input, i.e., how the PMDA gets input from the user. In the last years, speech recognition witnessed an impressive development, so that nowadays it is a main modality for software running on smartphone devices — either exploiting online services or using offline libraries. Therefore it is a main promising modality to realize full-fledged hands-free systems. However, healthcare contexts such as emergency and trauma management – which will be considered as the case study in this paper – puts forth challenges that make current speech recognition technologies still not effective enough in every situation, calling for adopting a *multimodal* approach. A

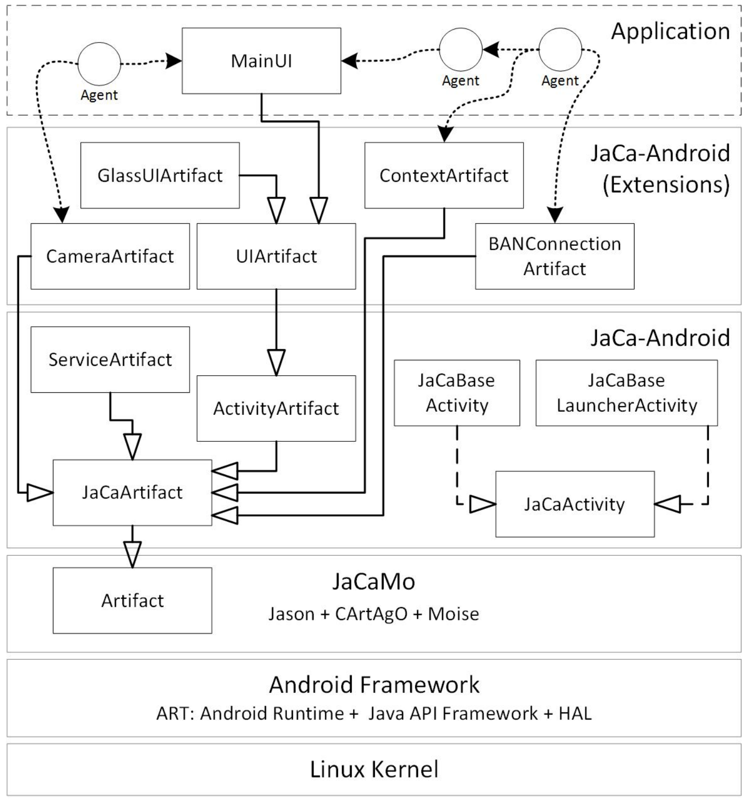

Fig. 1: PMDA Framework: an abstract representation of main components.

further modality is through touch-screens, not limited to the one of the mobile device where the PMDA is running, but including displays and video-wall available in the environment where the activity is taking place. In this case, a PMDA should be able to exploit such facilities as a service, allowing the user to switch the input – and output – interfaces flexibly and dynamically.

3.3 An Agent-Oriented Framework for Developing PMDA

From a programming point of view, the design and the development of PMDA needs to be supported by frameworks allowing to manipulate first class concepts like agents – in the meaning of the agent-oriented paradigm – and able to build systems that can be easily deployed on mobile and wearable devices. To build this kind of frameworks, two are the alternatives: introducing agent model directly into standards mobile applications frameworks – i.e. the Android Framework with the addition of new dedicated first class abstractions – or allow to an agent oriented framework to run on the top of the other mobile application developing frameworks. Within the case study presented in Section 4 (see prototype details in Section 5) we have opted for the second alternative. Starting from JaCaMo [3] – a well-known and used agent-oriented framework based on the A&A model – we have considered its already available incarnation for developing mobile applications using agents metaphors called JaCa-Android [25] extended and, in some parts, re-engineered with many new features oriented to develop PMDA for wearable devices and in particular for eyewear computing devices. In Fig. 1 is reported an abstract representation of the complete framework, starting from the Android Framework—since almost all the wearable device are based on it. This approach allows to design and develop PMDA in a quite effective way. In particular, the PMDA can be completely designed as a agent-based system, without any abstraction gap, even if is running on a mobile/wearable device—that is within the context of an Android application. The design phase is rooted on the A&A meta-model which allow to conceive an application: (*i*) choosing the set of agents to include on the base of the application needs and (*ii*) by properly identifying the set of artifacts to introduce in order to facilitate the agents' works and to interact with the Android Framework and device sensors. Furthermore, this approach makes it possible to easy deal with main challenges of Android and mobile applications like asynchronous event management – that can be managed by properly specifying agents' reactive behaviour – and proactive behaviours; challenges that are also typically retrievable in PMDA functional requirements.

4 A Case Study: The Trauma Tracker Project

The Emergency Department (ED) under study in this paper is the Trauma Center of a Hospital in Italy. The team leader – the so called *Trauma Leader*, usually a senior official – is in charge of producing the documentation paper. However this is just one of the several functions she/he has. During trauma resuscitation Trauma Leaders supervise the work of their teams and are actively involved in the actual resuscitation, and only *after* that work is finished they produce the report. That is, they recall and write down in prose the main facts of the trauma resuscitation process, documenting from memory and not real-time.

Actually, this is the typical situation of hospital emergency departments in Italy—besides the specific trauma center considered in this paper. Therefore, in this case, the availability of a system based on mobile and wearable technologies for trauma tracking and assistance, not only would improve the accuracy of the trauma documentation, but also significantly reduce the cognitive burden of the trauma team – of the Trauma Leader in particular – to create the reports.

4.1 Objectives

The short-term objective of the project is the development of smart technologies for tracking the events and data during the management of a trauma for trauma documentation purposes, in order to improve the accuracy and automate as much as possible the development of reports.

Medium-term objectives include the development of two further supporting levels of increasing complexity. The second level is about a first basic assistance, in terms of

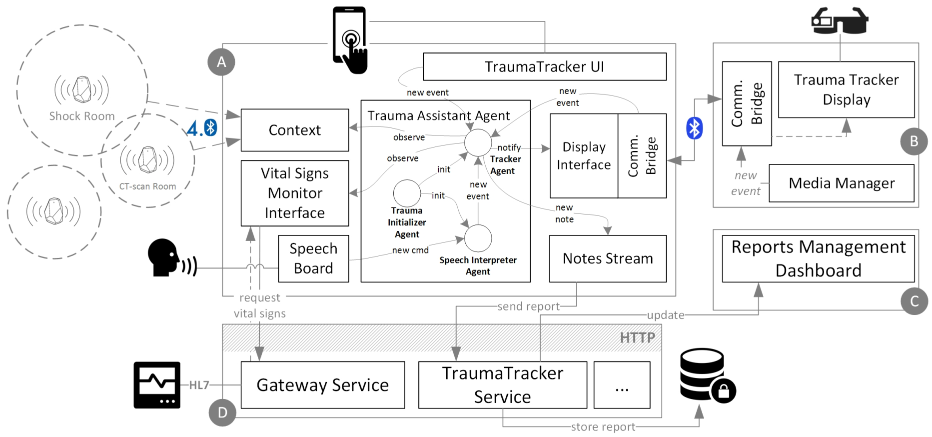

Fig. 2: TraumaTracker System Logical Architecture. Parts (A) and (B) represent the TraumaLeader subsystem and in particular the TraumaTracker Assistant Agent, Part (C) is the Reports Management subsystem and, finally, Part (D) represent the GT^2 infrastructure.

monitoring. Monitoring is about the automatic generation of *warnings* that are displayed on the smartglasses, about situations that the trauma leader may want to be notified without necessarily interrupting her activity flow. Situations may concern both the current state of the patient or its evolution – e.g., a warning could be generated if/when the value v of a vital parameter dynamically tracked by the system falls outside some predefined range $[min, max]$ –, and the temporal flow of actions carried on by the trauma team – e.g. some time t has elapsed after the administration of some drugs. Situations can be modelled as predicates over the *context*, including tracked information, current time and place where the trauma leader is, the identity trauma leader, and so on.

The third level is about a more *pro-active* kind of assistance, such as *suggesting* the workflow of steps to do according to in specific cases that require an ad-hoc treatment according to the trauma protocols adopted.

4.2 The TraumaTracker System Design

The coarse-grained architecture of TraumaTracker, represented in Fig. 2, includes three main parts: (*i*) the *TT Assistant Agent* (parts A and B in the figure), (*ii*) the GT^2 *infrastructure* (part D in the figure) and (*iii*) the *TT Report Dashboard* web application (part C in the figure).

Abstracting from details, which will be provided in Section 5, the *TT Assistant Agent* is the PMDA assisting the Trauma Leader during a trauma, running on her mobile and smart-glass wearable device. It encapsulates the control logic related to the different levels of functionalities, from tracking to monitoring and suggesting. Even if from a logical point of view it is useful to refer to this part as a single agent, its implementation may include multiple agents interacting with their local environment abstractions.

The GT^2 *infrastructure* is the local HTTP-based area network, including a set of web services that are exploited by the *TT Assistant Agent*. Currently, these services include (1) the *TT Report Management Service* – that provides a standard REST-ful APIs for collecting and managing trauma reports and accessing to related statistical data – and (2) the *TT Vital Signs Monitor Service* – that provides APIs for dynamically retrieving the vital signs parameters of a patient under trauma, and services to realise continuous monitoring. Data collected by these services are made available to other hospital applications running on the same infrastructure, in an open ecosystem perspective.

Finally, the *TT Report Dashboard* is a web application that allows users to access the trauma documentation i.e. the reports, providing functionalities to print and export them, as well as to do basic statistics. In next section we provide some details of the implementation of a first prototype for TraumaTracker, focusing in particular on the design and development of the *TT Assistant Agent*.

5 A First Prototype

In the first prototype, we focused on the tracking functionalities only, which were considered by the trauma team a priority with respect to the other levels. The main tasks of the Trauma Assistant Agent in this prototype are: *(i)* tracking events occurring in the emergency rooms related to a specific patient, inferring as much as possible data from the context (e.g., the place where a procedure is performed), and *(ii)* at the end of the trauma management, producing and sending a report to be sent to the *TT Report Management Service*.

Event tracking is mainly a reactive activity, in which the Trauma Assistant Agent:

- keeps track of the actions performed by the trauma team. These actions can be either procedures (e.g., endotracheal intubation, thoracic drainage, application of a tourniquet and many others) or drug/blood product infusion (e.g., millilitres of crystalloids or hypertonic solution, adrenaline, atropine, pools of cryoprecipitates, etc.);
- allows the trauma leader to take snapshots, record video or audio annotation, to be included in the report, exploiting the camera equipped with the smartglasses;
- allows to retrieve, display and track the current value of patient's vital signs—by interacting with the *TT Vital Signs Monitor Service* service.

Every event/note tracked by the agent includes both temporal (date and time) and spatial (location, specific room) information.

An important and challenging aspect of the agent is the strategy used to keep track of the actions performed on the patient by the trauma team.Currently, this occurs by reacting to commands that are explicitly requested by the Trauma Leader, either *(i)* as speech commands, or *(ii)* exploiting the user interface (UI) provided on the smartphone device where the agent is running, or *(iii)* exploiting the UI provided by touch displays available in the rooms where the trauma is managed (i.e., either the shock-room, the TAC room, the angio-room). The UI must be necessarily very simple and effective, minimizing the number of interactions and taps required to specify the action performed.

5.1 The Trauma Assistant Agent: Design and Implementation

The Trauma Assistant Agent has been implemented using JaCaMo, a BDI-based multi-agent oriented programming platform, which integrates the Jason agent programming language (to program BDI agents), the CArtAgO environment programming framework (to program agents' environment) and MOISE organization framework (to specify MAS organization) [3]. In particular, a version of JaCaMo running on mobile and wearable devices has been exploited, based on JaCa-Android [25], which introduces an agent-oriented programming model to design, develop and run agent-based applications on top of the Google Android platform (see Subsection 3.3 for details).

As already mentioned, even if at the logical level the Trauma Assistant Agent is a single conceptual high-level agent, the design and implementation in JaCaMo include multiple Jason agents that work together inside an environment composed of a set of resources and services wrapped inside a proper set of CArtAgO artifacts. In particular, current design include three Jason agents (see Fig. 2, part A), each one in charge of a different task:

- The *Trauma Initializer Agent*, responsible for initializing the tracking phase and collecting (and store) all preliminary information, essential for reports creation, including the identity of the trauma leader and the patient's initial health status.
- The *Tracker Agent*, in charge of keeping track of the trauma events, by monitoring and reacting to Trauma Leader actions and changes in the Trauma Leader context (e.g., the room where she is working). To this purpose, the agent uses a bunch of artifacts (some shared with the other agents), including:
  - a Notes Stream artifact to persistently store the stream of notes carrying information about the events;
  - a Display Interface artifact to display messages and warning that are perceived by the Trauma Leader either through the smart glasses, or other output displays (depending on the configuration of the system);
  - a Context artifact to be aware and observe information about current location of the Trauma Leader (i.e., the room), exploiting a BLE beacon-based localisation infrastructure deployed in the hospital environment (see below for more details);
  - a Vital Signs Monitoring Interface artifact, to dynamically retrieve the real-time data about patient's vital signs, encapsulating and hiding the interaction with the *TT Vital Signs Monitor Service* service.
- The *Speech Interpreter Agent*, responsible for the interpretation of speech commands. In particular, this agent recognises if a particular command may be accepted in that particular instant of the trauma management process – according to the actions work-flow – and, if so, translate it into an appropriate event to be sent to the Tracker Agent.

Fig. 3 shows a code snippet of the Tracking Agent, with in evidence the tools (artifacts) used by the agent and some of the plans adopted to react to relevant events occurring within the system[1].

[1] The full source code of the project is available at: `https://bitbucket.org/pslabteam/traumatracker`

```
+!init
<- makeArtifact("mainUI", ".MainUI", [], Id);
   +art_id(mainUI, Id);
   focus(Id).

+ui_ready [artifact_name(Id,mainUI)]
<- !setupTools;
   configureConnections;
   onEnvironmentSetupDone.

+!setupTools
<- makeArtifact("userCmds",".UserCmdChannel",[],Queue);
   focus(Queue);
   makeArtifact("cam",".Camera",[],_);
   makeArtifact("display",".DisplayInterface",[],_);
   makeArtifact("notes", ".NoteStream",[],_);
   makeArtifact("monitor",".VitalSignsMonitor",[],_);
   makeArtifact("context",".PhysicalContext",[],Ctx);
   focus(Ctx);
   makeArtifact("patient",".PatientHealthStatus",[],P);
   focus(P).

+user_cmd("new_procedure",Type) : place(Where)
<- showNewProcedurePerformedMessage(Type);
   addProcedureNote(Where, Type).

+user_cmd("new_drug",Type,Unit,Qty) : place(Where)
<- showNewDrugMessage(Type, Unit, Qty);
   addDrugNote(Where,Type,Unit,Qty).

+user_cmd("start_tracking", TL)
<- startRanging;
   registerStartEvent(TL);
   showTrackingStatus("active").

+user_cmd("end_tracking", Dest)
<-  registerTerminationEvent(Dest);
    showTrackingStatus("terminated");
    buildReport(Report);
    sendReport(Report, Res);
    !saveReport(Report);
    stopRanging.

+user_changed(User) : place(Where)
<- addTraumaLeaderNote(User,Where).

+place(P)
<- updatePlaceInfo(P).
```

Fig. 3: Tracker Agent code snippet in Jason.

Note that, to allow the Trauma Assistant Agent to detect the specific room in which the Trauma Leader currently is, we opted to enrich the environment (each relevant room) with an adequate number of Bluetooth low energy (BLE) Beacon. Beacons are battery equipped small devices that periodically broadcast their identifiers – that can be properly configured – to nearby smart devices enabled to their identification. In this case, we used an *Estimote* Beacon for each room, choosing the power of the transmitted signal according to the size of each room and avoiding area overlappings. The system collects informations from the environment and makes them available to the Trauma Assistant Agent exploiting the Context artifact that is focused/observed by the agent.

5.2 Evaluation

A first qualitative validation of the system has been carried on by the trauma team in a simulated environment, focusing in particular on the usability of the user interface (UI) and the responsiveness of the system. The basic configuration of the system – using the mobile devices only, without wearable technologies – appears to be ready and robust enough for being experimented in the real-world, providing already clear and measurable benefits – in terms of saved time and accuracy of the data – with respect to the current paper-based practice. So the next step, in this case, will be a further validation stage, in which the system is gradually introduced in the current practice, starting from the management of less critical trauma. Besides, this further stage will be essential to realise a more rigorous quantitative analysis and validation of the approach.

The adoption of smart-glasses – e.g., the intermediate configuration of the system – proved to be valuable to show information about vital signs in particular when the Trauma Leader is far from the vital signs monitor, and to take snapshots/video. As expected, the key benefit is to enable a first form of hands-free support, so that the Trauma Leader can take a snapshot or perceive information about the parameters without changing her focus and distracting from the scene. A critical point to be tackled before moving to the next validation stage is about the physical head mounting of the smart-glasses. On the one hand, current setting proved to be stable enough to deal with (abrupt) Trauma Leader's movements; on the other hand, we need to improve the flexibility in allowing the Trauma Leader to dynamically and seamlessly raise/drop the smart-glasses in some moments (to get them back, then), as well as reducing the effort (and time) needed to adjust the device.

Not surprisingly, the aspect that needs to be strongly improved in order to be usable in the trauma real-world settings is speech recognition. Currently, the performance is acceptable only for recognising basic commands used sporadically. This is not unexpected, since we are not using in the prototype specific speech recognition engine but the one available with the basic Android platform, functioning in offline mode. We expect to improve this aspect by investigating speech recognition technologies specifically tailored to the medical context.

6 Conclusion and Future Work

Currently, the TraumaTracker prototype implements just the tracking functionality, which is however useful already to improve the quality of the trauma documentation and to automate the generation of trauma reports. Next steps will be devoted to develop and integrate higher functionality levels, as described in Section 4. In particular, the very first next step will be to provide functionalities in terms of real-time *assistance* to the Trauma Leader/team, fully exploiting the hands-free characteristics of the system. The first kind of assistance which is being implemented is about the automatic generation of *warnings* that are displayed on the smart-glasses, about situations that the Trauma Leader may want to be notified without necessarily interrupting her activity flow.

The modular design adopted for the Trauma Assistant Agent makes it possible to implement the extensions without substantially change the behaviour of the existing internal agents. For instance, we plan to implement the new assistance functionalities as a new Jason Monitoring Agent, which observes and reasons about the notes created in the Notes Stream artifact, generating proper warnings to be displayed through the Display Interface artifact.

Finally, our medium-term research objective accounts for fully exploring the idea of *augmented hospital*, integrating agents with (mobile) augmented reality technologies for creating novel kinds of smart environments [24], providing more advanced functionalities to support individual and cooperative work. Finally, our medium-term research objective includes the investigation about the idea of *augmented hospital*, in which also the physical environment where the trauma is managed is augmented – besides the healthcare professionals. In that view, (mobile) augmented reality technologies can be integrated with pervasive ones, so as to create novel kinds of smart environments [24], providing more advanced functionalities to support individual and cooperative work.

References

1. Multiple negotiations among agents for a distributed meeting scheduler. In: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000). pp. 435–. ICMAS '00, IEEE Computer Society, Washington, DC, USA (2000), http://dl.acm.org/citation.cfm?id=518904.878931
2. Bardram, J., Baldus, H., Favela, J.: Pervasive computing in hospitals, pp. 48–77. CRC Press (2006), http://dx.doi.org/10.1201/9781420005332.ch3
3. Boissier, O., Bordini, R.H., Hübner, J.F., Ricci, A., Santi, A.: Multi-agent oriented programming with jacamo. Science of Computer Programming 78(6), 747–761 (2013)
4. Bulling, A., Zander, T.O.: Cognition-aware computing. IEEE Pervasive Computing 13(3), 80–83 (July 2014)
5. Bulling, A., Cakmakci, O., Kunze, K., Rehg, J.M.: Eyewear Computing – Augmenting the Human with Head-mounted Wearable Assistants (Dagstuhl Seminar 16042). Dagstuhl Reports 6(1), 160–206 (2016), http://drops.dagstuhl.de/opus/volltexte/2016/5820
6. Burstein, F., Zaslavsky, A., Arora, N.: Context-aware mobile agents for decision-making support in healthcare emergency applications. In: Workshop on Context Modeling and Decision Support (2005)
7. Chalupsky, H., Gil, Y., Knoblock, C.A., Lerman, K., Oh, J., Pynadath, D.V., Russ, T.A., Tambe, M.: Electric elves: Applying agent technology to support human organizations. In: Proceedings of the Thirteenth Conference on Innovative Applications of Artificial Intelligence Conference. pp. 51–58. AAAI Press (2001), http://dl.acm.org/citation.cfm?id=645453.652996
8. Civil, I.: What is quality care in trauma? Injury 38(5), 525 – 526 (2007), //www.sciencedirect.com/science/article/pii/S0020138307001040
9. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. 5(1), 4–7 (Jan 2001), http://dx.doi.org/10.1007/s007790170019
10. Domnori, E., Cabri, G., Leonardi, L.: Ubimedic2: An agent-based approach in territorial emergency management. In: 2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops. pp. 176–183 (May 2011)
11. Don, A., Brennan, S., Laurel, B., Shneiderman, B.: Anthropomorphism: From eliza to terminator 2. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 67–70. CHI '92, ACM, New York, NY, USA (1992), http://doi.acm.org/10.1145/142750.142760
12. Isern, D., Sánchez, D., Moreno, A.: Agents applied in health care: A review. International Journal of Medical Informatics 79(3), 145 – 166 (2010), http://www.sciencedirect.com/science/article/pii/S138650561000016X
13. Li, C., Giampapa, J.A., Sycara, K.P.: Bilateral negotiation decisions with uncertain dynamic outside options. IEEE Trans. Systems, Man, and Cybernetics, Part C 36(1), 31–44 (2006), http://dx.doi.org/10.1109/TSMCC.2005.860573
14. Lindquist, M.A., Johansson, E.P., Petersson, I.G., Saveman, B.I., Nilsson, C.G.: The use of the personal digital assistant (pda) among personnel and students in health care: A review. J Med Internet Res 10(4), e31 (Oct 2008), http://www.jmir.org/2008/4/e31/
15. Maes, P.: Agents that reduce work and information overload. Commun. ACM 37(7), 30–40 (Jul 1994), http://doi.acm.org/10.1145/176789.176792

16. Mann, S.: Wearable Computing as Means for Personal Empowerment. IEEE Computer Society Press, Fairfax, VA (May 1998), http://www.eyetap.org/wearcam/icwc98/keynote.html
17. Mitchell, T.M., Caruana, R., Freitag, D., McDermott, J., Zabowski, D.: Experience with a learning personal assistant. Commun. ACM 37(7), 80–91 (Jul 1994), http://doi.acm.org/10.1145/176789.176798
18. Modi, P.J., Veloso, M., Smith, S.F., Oh, J.: Cmradar: A personal assistant agent for calendar management. In: Proceedings of the 6th International Conference on Agent-Oriented Information Systems II. pp. 169–181. AOIS'04, Springer-Verlag, Berlin, Heidelberg (2005), http://dx.doi.org/10.1007/11426714_12
19. Newman, S., Steed, L., Mulligan, K.: Self-management interventions for chronic illness. The Lancet 364(9444), 1523–1537 (2004)
20. O'Connor, T., Raposo, E.A., Heller-Wescott, T.: Improving trauma documentation in the emergency department. Journal of Trauma Nursing 21(5), 238–243 (2014)
21. Okamoto, S., Scerri, P., Sycara, K.: Toward an understanding of the impact of software personal assistants on human organizations. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. pp. 630–637. AAMAS '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1160633.1160745
22. Orwat, C., Graefe, A., Faulwasser, T.: Towards pervasive computing in health care – a literature review. BMC Medical Informatics and Decision Making 8(1), 26 (2008), http://dx.doi.org/10.1186/1472-6947-8-26
23. Poggi, A., Bergenti, F.: Developing smart emergency applications with multi-agent systems. Int. J. E-Health Med. Commun. 1(4), 1–13 (Oct 2010), http://dx.doi.org/10.4018/jehmc.2010100101
24. Ricci, A., Piunti, M., Tummolini, L., Castelfranchi, C.: The mirror world: Preparing for mixed-reality living. IEEE Pervasive Computing 14(2), 60–63 (Apr 2015)
25. Santi, A., Guidi, M., Ricci, A.: JaCa-Android: An Agent-Based Platform for Building Smart Mobile Applications, pp. 95–114. Springer Berlin Heidelberg, Berlin, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-22723-3_6
26. Sarcevic, A.: "who's scribing?": Documenting patient encounter during trauma resuscitation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1899–1908. CHI '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1753326.1753611
27. Shankararaman, V., Ambrosiadou, V., Loomes, M., Panchal, T.: Patient care management using a multi-agent approach. In: Systems, Man, and Cybernetics, 2000 IEEE International Conference on. vol. 3, pp. 1817–1821 vol.3 (2000)
28. Silva, B.M., Rodrigues, J.J., de la Torre Díez, I., López-Coronado, M., Saleem, K.: Mobile-health: A review of current state in 2015. Journal of Biomedical Informatics 56, 265–272 (2015), http://www.sciencedirect.com/science/article/pii/S1532046415001136
29. Starner, T.: The challenges of wearable computing: Part 1. IEEE Micro 21(4), 44–52 (Jul 2001), http://dx.doi.org/10.1109/40.946681
30. Starner, T.: The challenges of wearable computing: Part 2. IEEE Micro 21(4), 54–67 (2001)
31. Starner, T., Mann, S., Rhodes, B., Levine, J., Healey, J., Kirsch, D., Picard, R.W., Pentland, A.: Augmented reality through wearable computing. Presence: Teleoper. Virtual Environ. 6(4), 386–398 (Aug 1997), http://dx.doi.org/10.1162/pres.1997.6.4.386
32. Su, C.J., Wu, C.Y.: {JADE} implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring. Applied Soft Computing 11(1), 315 – 325 (2011), http://www.sciencedirect.com/science/article/pii/S156849460900235X

33. Sukthankar, R., Davies, N., Siewiorek, D.P.: Activity-based computing. IEEE Pervasive Computing 7(undefined), 20–21 (2008)
34. Taboada, M., Cabrera, E., Iglesias, M.L., Epelde, F., Luque, E.: An agent-based decision support system for hospitals emergency departments. Procedia Computer Science 4, 1870 – 1879 (2011), http://www.sciencedirect.com/science/article/pii/S1877050911002614
35. Tambe, M.: Electric elves: What went wrong and why. AI Magazine 29(2), 23–27 (2008), http://www.aaai.org/ojs/index.php/aimagazine/article/view/2123
36. Tentori, M., Hayes, G.R., Reddy, M.: Pervasive computing for hospital, chronic, and preventive care. Foundations and Trends® in Human–Computer Interaction 5(1), 1–95 (2012), http://dx.doi.org/10.1561/1100000024
37. d. l. Torre, A.B., Lluch-Ariet, M., Pegueroles-Vallés, J.: Security analysis of a protocol based on multiagents systems for clinical data exchange. In: 2013 Seventh International Conference on Complex, Intelligent, and Software Intensive Systems. pp. 305–311 (July 2013)
38. Varshney, U.: Mobile health: Four emerging themes of research. Decision Support Systems 66, 20–35 (2014), http://www.sciencedirect.com/science/article/pii/S0167923614001754
39. Wagner, T., Phelps, J., Guralnik, V., VanRiper, R.: Coordinators: Coordination managers for first responders. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3. pp. 1140–1147. AAMAS '04, IEEE Computer Society, Washington, DC, USA (2004), http://dx.doi.org/10.1109/AAMAS.2004.97
40. Whitten, P., Holtz, B., LaPlante, C.: Telemedicine: What have we learned? Applied Clinical Informatics 1(2), 132–121 (2010)