Hybrid Teams: Flexible Collaboration Between Humans, Robots and Virtual Agents

Tim Schwartz[1,4], Hans-Ulrich Krieger[1,4], Ingo Zinnikus[1,4], Christian Bürckert[1,4], Joachim Folz[1,5], Bernd Kiefer[1,4], Peter Hevesi[1,5], Christoph Lüth[1,3], Gerald Pirkl[1,5], Torsten Spieldenner[1,2,4], Norbert Schmitz[1,5], Malte Wirkus[1,3], and Sirko Straube[1,3]

[1] German Research Center for Artificial Intelligence, DFKI GmbH
[2] Saarbrücken Graduate School of Computer Science
[3] Robert-Hooke-Strasse 1, 28359 Bremen, Germany
[4] Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
[5] Trippstadter Strasse 122, 67663 Kaiserslautern, Germany,
firstname.lastname@dfki.de,
WWW home page: http://dfki.de

Abstract. With the increasing capabilities of agents using Artificial Intelligence, an opportunity opens up to form teamlike collaboration between humans and artificial agents. This paper describes the setting-up of a Hybrid Team consisting of humans, robots, virtual characters and softbots. The team is situated in a flexible industrial production. Once established, Hybrid Teams can generally accomplish diverse mission scenarios. The work presented here focuses on the architecture and the characteristics of the team members and components. To achieve the overall team goals, several challenges have to be met to find a balance between autonomous behaviours of individual agents and coordinated teamwork. A Hybrid Team can heavily benefit from the heterogeneity of the team members. Humans have the highest overall intelligence, so they are always in the center of the process and take over a leading role if necessary.

1 Introduction

In our everyday life, the role of robots and other agents using Artificial Intelligence (AI) changes rapidly: from purely automated machines and programs up to companions that make suggestions, advices or assist in physical tasks, e.g. carrying heavy weight loads. This process is most obvious in factories where robots become lightweight and work in close vicinity or even in direct collaboration with human workers. At the same time, the industrial production methods and requirements are also changing, demanding more flexible production lines. One concept to accomplish both – using these new possibilities of AI and meeting the requirements of Industrie 4.0 [17] of complex and flexible production – is to establish a new kind of collaboration of humans, robots and virtual agents as Hybrid Teams. As in all teams, the idea here is to benefit from the different characteristics of the individual team members and at the same time make

use of the fact, that team members can substitute each other temporarily in completing tasks when resources are running low. For purely human teams, this is completely natural behavior: if a team member drops out, the team tries to compensate this. However, industrial robots are still highly specialized in their task, so that a new level of flexibility and universality is required to make a robot capable to temporarily substitute a human or robotic team member. With this new robotic skill, the goal of the team can be achieved with higher robustness and flexibility at the same time.

The work presented in this article is based on results of the project *Hybrid Social Teams for Long-Term Collaboration in Cyber-physical Environments* (HySociaTea) [40] that targets the setting-up of a Hybrid Team. In the following, we describe the setup, types of team members and components of this Hybrid Team, and focus on the central communication architecture that defines how the team interacts, how tasks are assigned to team members, and which levels of autonomous actions are possible within the team. In addition, we discuss these aspects on a more general level.

2 Related Work

Agent-based approaches have already been used for some time in distributed manufacturing scenarios (for an overview see [22]). Team members in those scenarios have diverse capabilities, which need to be represented accordingly in order to be leveraged. [43] use autonomous agents to represent physical entities, processes and operations. For coordinating the activities among the agents, communication among team members is required which in general can be done using centralized or decentralized approaches. Agent-oriented approaches use negotiation protocols for resource allocation, e.g. the Contract Net Protocol or its modified versions, but also distributed market-based paradigms, auction-based models and cooperative auctions to coordinate agent and, in particular, robot actions. E.g. [26] propose a market-based multi-robot task allocation algorithm that produces optimal assignments.

An influential line of research has been initiated by the concept of *joint actions* [24], leading to shared plans, where planning and execution in teams need to be interleaved in order to react to unforeseen circumstances [7], [29]. In the STEAM framework [41], group activities are structured by team-oriented plans which are decomposed into specific sub-activities that individual robots can perform. Following that strand of research, [37] focuses on the impact that noisy perception of actors has on the coordination process and use cooperative perception techniques to address this problem.

While many approaches consider inter-robot coordination, the problem of Hybrid Teams, where members have a certain autonomy has received much less attention so far. A prototype for human-robot teams for specific domains such as urban disaster management has been proposed and developed by [34]. Nevertheless, the investigation of the challenges and requirements involved in flexible human-robot teams and their coordination is still an open topic. In our approach,

we focus on how to include human actions and human communication capabilities into the otherwise purely technical infrastructure for the artificial agents. Instead of using a rigid, machine planning-module, which controls the actions of each team member, we rely on the human's planning capabilities and thus keep him in the center and in control of the production process.

3 Setting and Agents

Hybrid Teams can have many goals depending on the field of application and the concrete mission at hand. In our vision, the team should organize itself according to the individual skills of each team member. Principally, the team members are one of three possible types of agents: Humans play the central role in the team, since they have the highest overall intelligence[6] and can thus react extremely flexible to new situations. If necessary, humans should be able to even reorganize the team by command (e.g. if the whole mission is in danger). Robots typically take over tedious or physically demanding tasks or go to hazardous (or even hostile to life) areas, and the virtual characters (VCs) have the role of assistants providing a straightforward interface to digitally available information.

A typical setting is a production scenario: The Hybrid Team handles jobs with batch size one, has to reorganize itself and even handle multiple tasks in parallel. While the actions of each worker in a standard manufacturing scenario are often predetermined, i.e. each team member follows a more or less rigorous plan, new production settings for highly customized products will demand a flexible behavior of the whole team. Tasks and responsibilities cannot easily be predetermined and the creativity of the team, especially by the human worker, plays an important role.

Consider, e.g., a customized packaging scenario [35] A customer brings an arbitrary item to the workshop in order to get it packaged in a very customized fashion. This could be, for example, a precious vase that is to be shipped overseas or a technical prototype of a new device that is to be presented to a potential investor. In such a scenario, the human will be the only team member to a) understand the customer's request and b) to come up with a rough plan on how to accomplish the request and c) to deal with all the problems that might arise during the endeavor. The remaining team members should assist and, if possible, they should do so in a helpful, proactive manner, i.e. without being directly instructed.

In the following the different agents, their characteristics and role in our realized Hybrid Team are described in short. An overview of the team is given in Figure 1.

3.1 Augmented Humans

Unlike robots and VCs, which are technological companions, humans need specialized devices, wearables and interfaces to communicate intuitively with the

[6] the sum of cognitive, social, practical, creative etc. intelligence

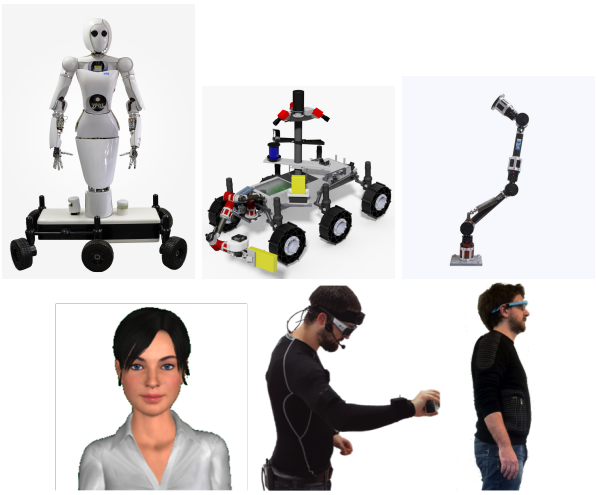

Fig. 1. Members of the Hybrid Team in the project HySociaTea(from top left to bottom right): Aila, the humanoid service robot, Artemis, the logistics robot, Compi the "helping hand", Gloria, the virtual character, a worker, wearing a sensor jacket and eye-tracking glasses, and a technician, wearing Google Glass.

other members of a Hybrid Team and to feed the whole system with information. For humans, speech is the most natural way of transmitting information, so information input and output via speech should be possible on all levels. In particular, the architecture connecting the team should contain an entity that decomposes speech acts, feeding planning- or task monitoring components (see Section 4) with information, so that, e.g., the human can ask for a specific item or action. In addition, the other team members need an elaborated speech recognition and speech generation system as well to interact with the human in a natural way.

The whole communication of the human with the team should be realized using massively multimodal man-machine interfaces, containing a large number of in- and output channels that use information coming from speech, gestures, haptic interaction, and facial expressions etc..

The human team member can interact with the system either using parts of the instrumented environment or via various wearables. The information of all human related sensors is ideally combined in a fusion module that is thus capable to produce data about the human that is similar to what the robotic team members can provide about themselves. Robots can then use this information for their own planning of movements or other actions. E.g., if a human is looking at a specific position, saying "I need this item", a combined information from speech, direction of gaze and location of objects in the environment will result in an understanding of the human intention. In the following we describe the subsystems that we currently obtain information about the human team members.

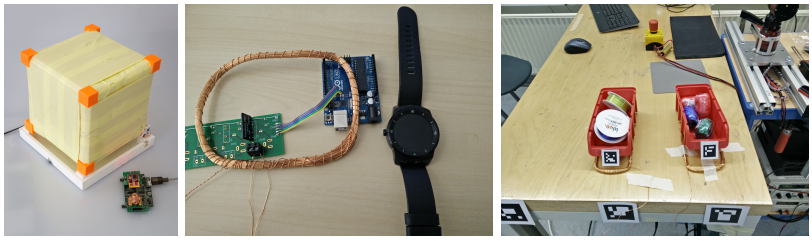

Fig. 2. Left: Indoor localization system used for determining the position of the human agent; Middle: mBeacon proximity detector; Right: Use-case for mBeacon in the scenario: the system registers when a worker with a smartwatch interacts with one of the boxes.

Localization The position of the human can be an essential information for understanding the current context and tracking activity. To interpret for example a simple voice command like "Bring me a box" the system has to locate – amongst other things – the worker, in order to determine the delivery position of the box. Due to the demanding environment involving both human and robotic agents, a robust localization is necessary:

For rough position estimation (error below 80cm) of humans, we developed a localization system based on oscillating magnetic fields (see [31]), relying on stationary anchor points, which are sequentially generating magnetic fields, and wearable receiver units, which measure the amplitude of the induced voltage signals at the human's position (Figure 2). Each anchor point has a 4 meter radius range. A magnetic field model can then determine a receiver's position based on the measured voltages and the known layout of the anchor points.

Tasks requiring higher accuracies are supported by our mBeacon system ([30]). The region of interest is tagged by magnetic field coils encoding a region ID in a quasi-static magnetic field, by applying a PWM signal to the coil. Using typical magnetic field sensors, which are included in most of the smartwatches or smartphones, we can detect an ID up to a distance of 30-40 cm. The system is used in this scenario to detect if the worker is interacting with specific tools or containers (see Figure 2).

Eye Tracking & Object Recognition Using eye-tracking glasses makes it much easier to determine what part of the scene an augmented human is looking at. Compared to a few years ago, technological advances have resulted in comfortable, much less intrusive, and lightweight devices, which makes wearing these glasses for extended periods of time much more realistic.

Along with gaze data, such as focused point and pupil radius, eye tracking glasses usually feature a world camera which captures the point of view of the wearer. We stream all provided data along with compressed video to a fusion module (see Section 3.1) for processing.

One of the unique applications of combining video with gaze is gaze-targeted object recognition. Since humans are more likely to attend objects rather than

Fig. 3. (a) Sensor jacket with visualization in the background (b) Stick figure model of the upper body motion tracking with six segments

background [10, 15, 45], using gaze information provided by eye tracking glasses helps to alleviate confusion caused by clutter by only analyzing a much smaller part of the image. Furthermore, the gaze information explicitly hints at the currently most important object in the scene.

Our object recognition system is based on the DeepSentiBank [4] deep convolutional neural network. It is implemented using the Caffe framework [11], which delivers real-time performance via GPU acceleration. To quickly adapt the network to our scenario, its $fc7$-layer output is fed to a linear support vector machine for classification. Initial tests were done on a small data set of 16 household objects with just 3 training images per class, that are bootstrapped to a total of 18000 samples by random rotation, cropping, noise, and brightness transformations.

Test precision for this system is 61.7%, which is already close to the best real-time capable method ORB [32] at 55.3%. Slower methods like SURF [2] (63.8%) and SIFT/SURF [27] (80.8%) can perform significantly better in this scenario, but we expect improved precision from the deep network as soon as more real training data is supplied to the network.

Sensory Jacket A sensory jacket [28] allows to sense the orientation of the upper torso and head, as well as the motion and position of the upper and lower arms. In HySociaTea, the worker is equipped with such a sensory jacket, which includes six motion sensors as depicted in Figure 3a. All motion sensors are 9-axis inertial motion sensors providing acceleration, gyroscope and magnetic field data from the trunk, the head and both arms (upper and lower arm separated). Each motion sensor data is streamed to an Extended Kalman Filter implemented closely related to Harada et al. [8] and [16]. The state of the filter consists of a quaternion representing the orientation of each segment in relation to an earth-centered north aligned global coordinate frame. In contrast to Harada et al. the measurement models are not implemented according to the proposed reliability detection but an adaptive measurement noise is applied to the updates corresponding to the deviations from static acceleration and static magnetic field measurements. This modeling technique avoids false reliability detections. Each orientation estimator provides the global orientation of one of the segments.

All segments together define the human skeleton model consisting of the six mentioned components as depicted in Figure 3b. The update rate of the filters are 100 Hz and the achievable orientation accuracy under moderate magnetic field disturbances is in the range of 10 degrees.

Fusion Module Though each of the sensors described previously can deliver useful information on its own, they only provide their full potential through sensor fusion where all generated data needs to be combined at one central hub. Also, wearable sensors need to be small, lightweight, and at low power: Complex processing should thus take place on an external machine. We realize this through a fusion module to which all worn sensors can connect through our communication middleware (see Section 4). From an architectural and semantical point of view, the fusion module feeds data about the human into the system that are similar to that of robots (which usually also contain such a module that combines e.g. the states of motors into information about the current pose of the robot).

There are several benefits from this: First, we reduce the amount of data that has to be streamed though the team network. The unprocessed data streams generated by these sensors would not be useful to other agents in the Hybrid Team and would unnecessarily take away bandwidth. For instance, the world camera of the eye tracking glasses generates $\sim 84MB/s$ (1280×960 pixels, $24Hz$) of uncompressed frame data. Later models feature cameras with even higher frame rates and resolutions. We also reduced transmission delays by connecting directly to the fusion module. Second, as more efficient algorithms are used and technology advances, later iterations of the Hybrid Team may apply the module closer to these sensors, so it will ultimately be a wearable component as well.

At its core the fusion module performs the following steps: (i) Preprocessing: convert to common data representation, filtering; (ii) Synchronization: compensate for different sampling rates, system times, and delays of sensors; (iii) Fusion: select best-fitting samples from each stream and perform sensor fusion; (iv) Interfacing: convert to compatible types of the team network and publish to outside world.

A simple example for fusion operation is the combination of location, orientation, and gaze direction to disambiguate the region of interest that the augmented human is currently attending. This can be used to enhance speech interactions. E.g., "Can you bring a box over there?", where other agents are able to resolve "over there" to the attended area (workbench, shelf, etc.). Further, combining this approach with object recognition allows determining which agent was addressed by checking recently attended agents.

3.2 Robots

The robots of Hybrid Teams are not classical industrial robots. Instead, these are autonomous agents, which are typically mobile and capable of performing certain manipulation tasks. In the future, these robots will gain higher levels of

autonomy and solve subproblems or tasks on their own. Because they share their workspace with humans, safety is an important issue which is nowadays typically addressed by using lightweight systems with low forces. Robots can substitute or share tasks with virtual characters and humans to a certain degree.

In the Hybrid Team presented here, three robots are integrated (see Figure 1) The robot *COMPI* [1] is the only stationary robot in the team. COMPI can switch between various stages of flexibility or stiffness and acts as a "helping hand" for a human, e.g. holding objects. *AILA* [23] is a humanoid, mobile dual-arm robot, which was originally developed to investigate aspects of mobile manipulation. A robot like AILA is an ideal candidate for a real world communication partner for humans. *ARTEMIS* [36] is a rover equipped with a manipulator. In the Hybrid Team, rovers like ARTEMIS can act as logistics robots, transporting tools, building-material or objects from and to different locations.

Robot Control The robots applied here are developed using BES-LANG [44], a set of domain-specific languages and tools for describing control systems, robot abilities and high-level missions for robotic mobile manipulation, based on ROCK [14], Syskit [12] and Roby [13].

For the mobile robots AILA and ARTEMIS, we have developed the abilities to generate geometrical and traversability maps from the data perceived from their laser scanners. The robots can localize themselves on the map and navigate on it. We provide contextual information on the map by defining distinct regions (e.g. "left shelf") using map coordinates. Navigation trajectories can be planned using the Anytime D-Star planning algorithm from the SBPL library [25]. A trajectory controller generates 2D motion commands from the planned trajectories, which are mapped to actual wheel actuation commands by the robot's motion controller module.

For grasping objects, we use visual servoing controllers that receive their visual feedback using the ArUco marker detector [5]. Currently, we use those labels for both, the objects and the storage locations. Additionally, there are basic abilities to follow joint or cartesian way-points or relative movements.

By sequencing these abilities more complex tasks are realized. An example is a *bring task* composed of a navigation-ability to the location where the desired item is located, followed by a grasp-ability and then again followed by a navigation- and a hand-over-ability.

For each robot, we map a set of high level tasks that is implemented in the robot and prioritize these tasks to reflect the robot's role within the team. To ensure determined robot behavior, only one such task can be executed at once per robot. During task execution feedback is reported to a Blackboard (see Section 4) indicating the progress of the task.

To provide feedback about the state of the robots to other team members, we created a bridge that is similar to the fusion module for the human-centric sensors.

Collision Detection In a Hybrid Team with (multiple) mobile robots, collision should be avoided with other robots, with objects, and–most importantly–with the humans. These concerns have actually been one of the main obstacles for the wider adaption of human-robot cooperation; current standards and practices make it nearly impossible to have the two cooperating in the same space, so it is of paramount importance to address this problem in HySociaTea.

Collision avoidance works in many ways: on the planning level, we can attempt to make sure that the robot plans his trajectory free of collisions. This breaks down when obstacles appear unexpectedly, and moreover to base a safety argument on it, one would have to verify the planning algorithm. Thus, we supplement high-level collision avoidance with a low-level *collision detection*. This is a module which supervises all movements of the robot, and checks whether the current movement will lead to a collision. Before this is the case, an emergency brake is initiated.

The key concept of our collision detection is a *safety zone*. This is the area of space which is covered by the robots manipulators at their current trajectories until breaking to a standstill. Calculating the safety zones efficiently in three dimensions is a hard problem which has been solved by the KCCD library [42]; it models the trajectories as sphere swept convex hulls (SSCH), which can be manipulated efficiently.

The library needs reliable sensor input to detect obstacles, which is provided by laser scanners mounted on the robots. This required an extension to the KCCD library to check collisions (intersections) of the SSCH with the point clouds returned from the laser scanners.

The collision detection is integrated into the ROCK framework, and runs locally on each robot.

3.3 Virtual Character

Virtual Characters (VCs) take over a special role in Hybrid Teams, because they are not physically present in the real world and so they cannot take-over physical tasks. Instead, VCs represent purely software-based components and serve as a more natural interface for humans than pure text output (written text or spoken text) without such a graphical impression of a human. In addition, they can also transmit emotions via gestures and facial expressions.

Our VC Gloria (Figure 1 is realized using a commercial VC SDK called CharActor, provided by Charamel[7]. The CharActor SDK already includes the complete rendering engine, a text-to-speech (TTS) engine including lip-synchronization, and a large library of facial expressions and motions.

3.4 SoftBots

In contrast to VCs, SoftBots are purely software based modules without physical or graphical embodiment. These SoftBots typically aggregate data produced by

[7] http://www.charamel.com

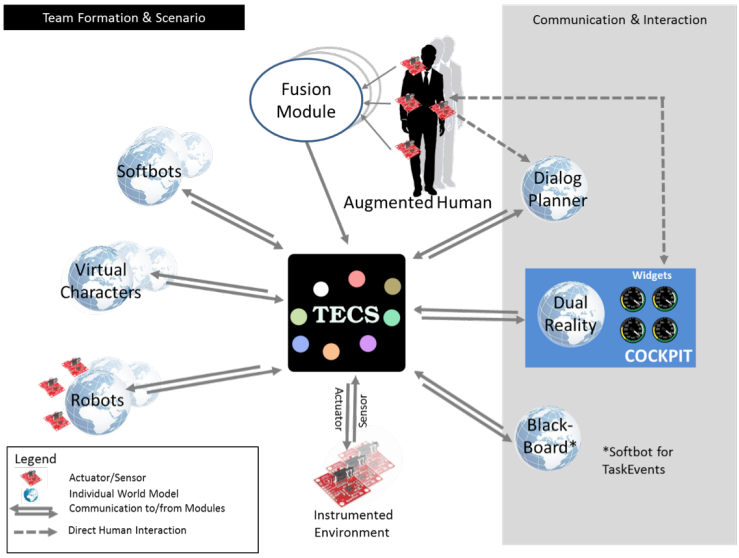

Fig. 4. Architecture of HySociaTea: Communication is established via our event based middleware TECS. Robots, virtual characters, SoftBots the Dual Reality module and the dialog engine each have their individual world model. The Blackboard is a SoftBot that contains all open tasks.

other team members (e.g. raw sensor data, speech acts) and in turn update databases or provide meaningful, refined data. There are several SoftBots in our current system, e.g. to keep track of the location of objects, tools and materials, or to convert numeric position-coordinates into semantic descriptions.

As an example for a more complex SoftBot, we implemented a module that collects information about the worker's requests for building-material, and that automatically learns, which materials are often used together. The VC uses the information provided by this SoftBot, to pro-actively ask if the additional material is also needed.

4 General Architecture and Communication

A central question when setting up a Hybrid Team is the realization of suitable interfaces. Humans usually use speech, gestures and facial expressions to transfer information. Artificial agents however can use direct data streams to communicate with the system and other artificial team members.

In our vision, the team is centered around the human worker. He is the one who uses his creativity, skills and knowledge to determine which tasks have to be fulfilled in order to reach the main goal. An example for a main goal is the already mentioned building of a sturdy packaging for a hand-built–and thus unique–vase. While taking measurements of the vase, the worker could for example dictate

the needed materials into the system. The team, which is then informed about the needs of the worker, should then autonomously fulfill these needs, if possible.

To deal with these different levels and requirements, we designed and implemented four core modules to realize the communication within a Hybrid Team: a communication middleware, a blackboard for task management, a dialog engine, and a dual reality module. The overall architecture with these core modules and the team members is shown in Figure 4.

The central idea is this: the worker issues commands or his needs using the dialog engine. The dialog engine creates tasks (e.g. bring(item, toLocation)) and puts them on the blackboard. Other team members access the blackboard to identify open tasks, and vouch to execute them, if they are able to fulfill them. In a nutshell: the blackboard exposes current tasks to the team, the middleware distributes the information within the team and the dialog planner enables a human-friendly translation (and access) to the middleware. The dual reality contains a representation of the scene on-site based on the information that is distributed via the central middleware. In the following, we describe each of these modules in more depth.

4.1 Event-based Middleware: TECS

Communication between all team members as well as between all submodules and potential sensors and actuators of the instrumented environment is established via an event-based middleware that has been developed in the project. TECS is short for Thrift Event-based Communication System. As the name already implies, it is based on Apache's cross-language services-framework Thrift[8]. TECS uses the Thrift IDL (Interface Definition Language) to describe data-structures, which can then be translated into data objects of various programming languages (e.g. Java, C#, C++, Python etc.) using the Thrift compiler. These objects can then be transmitted via the TECS server as event-based messages. All clients can be addressed via a publish-subscribe mechanism. For bound connections between clients, TECS also provides remote-procedure-calls and message-passing mechanisms. Communication partners find each other using a UDP multicast discovery strategy, which has also been integrated into TECS. In contrast to other systems that advertise services regularly, the TECS discovery strategy uses a client-side multicast request, which is answered by the service provider directly with a unicast description response. Services are always encoded into URIs, which are combined with service-type-descriptors and UUIDs to identify appropriate communication partners. For instance, each TECS server responses with `tecs-ps://<ip>:<port>` to a client-side request with service-type-descriptor `TECS-SERVER`. Since a TECS server can have multiple IPs and ports in the same scenario, depending on the hardware interfaces, each unique instance is identified by a 128-bit UUID. Remote-procedure-calls and message-passing providers can use the same strategy to find appropriate communication partners.

[8] https://thrift.apache.org

The independence from a particular programming language as well as a unified discovery strategy are very important features for Hybrid Teams, as the multiple subsystems –that are typically present in real production scenarios– are usually implemented in various programming languages, and clients can join and leave the environment regularly, which makes hard-wired connections at least impractical. We have published TECS under Creative Commons (CC BY-NC 4.0)[9] License and it can be downloaded via our website[10].

4.2 Task Management: Blackboard

As mentioned above, the blackboard is a viable component of the presented architecture of the Hybrid Team. It stores all subtasks that have to be fulfilled in order to accomplish the main goal. In a strict sense, the blackboard is a (rather single minded) SoftBot with a very restricted world model: it only cares about unfinished working tasks. All team members have access to the information on the blackboard. Artificial agents do this via TECS, as the blackboard broadcasts all tasks it receives. Humans have access to the blackboard through a graphical representation. Each team member can decide if they are capable to fulfill the task and can then commit themselves to it. This is done with a first-come- first-serve policy, i.e. the fastest team member gets the job. As of now, humans can commit to a task using speech commands (e.g. "I will do task number eight"), through a GUI on a tablet or via swipe gestures on Google Glass. In a more elaborated system this should be done by plan- or action-recognition, i.e. humans can just start fulfilling a task and the system will automatically assign the appropriate task to them.

4.3 Dialog Planner

The dialog planner processes speech input of the users and plans dialog acts for the artificial team members. In the Hybrid Team of the project HySociaTea the robots and the VC are capable of producing speech output via a Text-To-Speech (TTS) module. The components for natural language generation and interpretation themselves are located within the respective artificial team members. The dialog manager follows the *information state/update* approach [21], albeit in a modified form.

The implementation of the information state for the manager is based on RDF and description logics, which is uniformly used for specification as well as storage of dialog memory, belief states and user models. The extended functionality of the description logic reasoner *HFC* (see below) makes it possible to equip the collected data with time information, which allows us to use the whole history of dialogs, user and other data for decision making. The management engine itself consists of complex reactive rules which are triggered by incoming data, be it from automatic speech recognition (ASR) or other external sensors

[9] https://creativecommons.org/licenses/by-nc/4.0/
[10] http://tecs.dfki.de

or information sources, and additionally have access to the complete data and interaction history. An external probabilistic process helps to resolve alternative proposals for the next dialog moves, and, together with the uniform data representation, opens the door for machine learning approaches to improve the decision making. Although hierarchical state-machines are somehow the "standard" for dialogue management (aside from pure machine learning approaches à la Gasic & Young [6]) previous experiences have shown that they badly generalize to new situations, and are cumbersome when it comes to modularization and reuse, which is why we have decided to go for a rule-based formalism.

Ontologies In HySociaTea we have developed an ontology that consists of three sub-ontologies which are brought together via a set of interface axioms, encoded in OWL [9]. The first ontology is a minimal and stripped-down upper ontology. Most notable for HySociaTea is a representation which distinguishes between atomic Situations and decomposable Events. The second ontology represents knowledge about the HySociaTea domain, basically distinguishing between Actors and Objects. The domain ontology also defines further XSD datatypes, such as point or weight. The most sophisticated ontology integrates ideas from the DIT++ dialogues act hierarchy [3] and from FrameNet [33]. Frames and dialogue acts are modelled independently from one another, but dialog acts incorporates frames through a specific property. Dialogue acts encode the sender, the addressee, but also the succession of dialogue acts over time. Modelling the shallow semantic arguments inside the frame frees us from defining repeating properties on various dialogue acts over and over again. The ontology is used by the dialogue planner and also employed to encode time-varying data (e.g., *positional* information about tools or packaging material). Through the transitivity of spatial properties such as contains, natural language communication involving spatial reference then becomes more natural: instead of saying "Give me the tape from the *box* in the *upper shelf*", we might simply say "Give me the tape from the *rack*". The temporal representation extends the RDF triple model by two further arguments to implement a special form of *transaction time* [38] in *HFC* [20]. The ontologies are available as OWL-XML and N-triple files through the Open Ontology pages.[11] The ontology together with instance data (e.g., information from the dialogue) is hosted by the semantic repository and inference engine *HFC* [19].

4.4 Dual Reality

A dual reality component can be useful as a management or remote monitoring tool. It is visualizing information that is sent by the team via the central middleware. The dual reality also serves as an intuitive introspection in the system, visualizing what information is available. Therefore, a coordinator (even when not being on-site) can see what is going on and eventually influence the

[11] http://www.dfki.de/lt/onto/hysociatea/

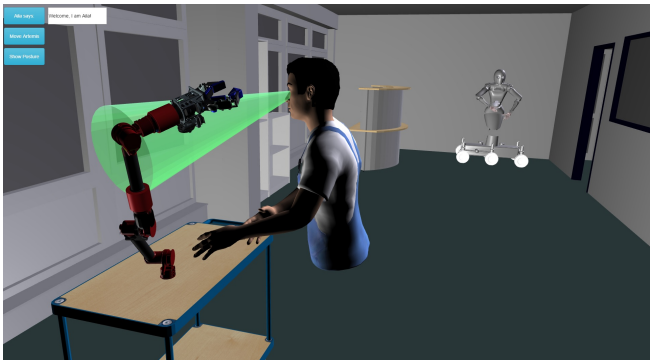

Fig. 5. The dual reality client: Positions, orientations and postures of actors, as well as the gaze direction of the human, are visualized with respect to the real world counterparts. The interactive 3D scene allows the dual reality to modify the scene at place, e.g. by sending movement commands to the robots.

scene. Furthermore, new team members have a direct access to the current and previous states.

The dual reality applied here is based on FiVES[12] (Flexible Virtual Environment Server) as server component, and a Web-browser based interactive 3D visualization, implemented as HTML5 / JavaScript application, using XML3D [39] as rendering framework. The respective client is shown in Figure 5: Robots and humans are represented as 3D models. Their position, orientation and posture of the actors, as well as the gaze direction of the worker (green cone), are visualized as they currently appear at the real world counterpart.

XML3D provides convenient mechanisms to express a 3D scene in terms of a assembly of configurable *Assets* [18], which fits very well to a scenario of independent actors in a shared work space. In this case, each actor is represented as individual Asset, with the data transferred via TECS being the input parameters for the individual asset configurations (e.g. individual joint values for different robot actors).

FiVES is a Virtual Environment server that provides real-time communication to heterogeneous clients. It stores the current world state in a generic Entity-Component-Attribute format and allows heterogeneous clients to receive updates to the world states in real-time in a publish- subscribe manner. We used the plug-in mechanism provided by FiVES to extend the existing server implementation with a C# TECS client, so we can directly apply events that are broadcasted in the TECS network, in particular, position updates of human worker and robots, as well as events that describe changes in joint angles of robots or the worker. Incoming TECS events are continuously applied to the local scene representation, so that the virtual counterpart is always consistent to the actual work site.

[12] http://catalogue.fiware.org/enablers/synchronization-fives

The benefit of introducing a server module over attaching the dual reality client directly to TECS is twofold: First, the data stored in FiVES is always reflecting the actual state of the work site. This means for actors that join the dual reality later after numerous changes to the scene happened already, they still find the correct World state in FiVES, while the individual events are unaware of their effect on the overall world state. Second, having the data transmitted by the events converted to the unified Entity-Component-Attribute representation as used by FiVES allows for simple serialization of the world data into a JSON (Java Script Object Notation) format which allows immediate application of the server data to our browser-based 3D visualization.

5 Conclusions

Setting up Hybrid Teams of humans, robots, virtual agents and several softbots is a powerful strategy to explore new application fields and deal with increasing demands for flexibility in manufacturing. However, the realization of these teams still bears a lot of challenges for artificial agents and their collaboration with humans, since suitable and intuitive interfaces have to be implemented to connect physical and digital information, and autonomous and flexible robots have to be perceived as adequate partners in the team.

With the presented work we have established a structural and architectural basis for Hybrid Teams to start executing missions. Still, recent techniques in AI contain many more options how the performance of such a team can be improved, so that this can be seen as a starting point to create teamwork within the team. A demonstration video of the working Hybrid Team can be seen on http://hysociatea.dfki.de/?p=441.

Besides research on the technical feasibility of setting-up a Hybrid Team, another key aspect is the development of (robotic) team-competencies as well as intelligent multiagent behavior, both of which are also important aspects in purely human teams. The technical systems developed in HySociaTea are mainly meant to be used as assistance systems for humans working in production plants; the robots should therefore be perceived as partners in the overall working process. On the long run, the team organization, as developed and examined here, can be used in different real-world scenarios, e.g. in modularized production facilities in the factory of the future, as rescue teams in emergency situations, or to realize the necessary division of labor between humans and machines for the safe deconstruction of nuclear power plants.

Acknowledgment

The research described in this paper has been funded by the German Federal Ministry of Education and Research (BMBF) through the project HySociaTea (grant no. 01IW14001).

References

1. Bargsten, V., de Gea Fernández, J.: COMPI: Development of a 6-DOF compliant robot arm for human-robot cooperation. In: Proceedings of the International Workshop on Human-Friendly Robotics. Technische Universität München (TUM) (2015)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Computer Vision–European Conference on Computer Vision (ECCV 2006), pp. 404–417. Springer (2006)
3. Bunt, H., Alexandersson, J., Choe, J.W., Fang, A.C., Hasida, K., Petukhova, V., Popescu-Belis, A., Traum, D.: Iso 24617-2: A semantically-based standard for dialogue annotation. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12). pp. 430–437 (2012)
4. Chen, T., Borth, D., Darrell, T., Chang, S.F.: DeepSentiBank: Visual sentiment concept classification with deep convolutional neural networks. arXiv preprint arXiv:1410.8586 (2014)
5. Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J., Marin-Jimenez, M.J.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition 47(6), 2280–2292 (2014)
6. Gasic, M., Young, S.J.: Gaussian processes for POMDP-based dialogue manager optimization. IEEE/ACM Trans. Audio, Speech & Language Processing 22(1), 28–40 (2014)
7. Grosz, B.J., Hunsberger, L., Kraus, S.: Planning and acting together. AI Magazine 20(4), 23–34 (1999)
8. Harada, T., Mori, T., Sato, T.: Development of a tiny orientation estimation device to operate under motion and magnetic disturbance. In: The International Journal of Robotics Research. vol. 26, pp. 547–559 (June 2007)
9. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S.: OWL 2 web ontology language primer (second edition). Tech. rep., W3C (2012)
10. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis & Machine Intelligence 20(11), 1254–1259 (1998)
11. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the International Conference on Multimedia. pp. 675–678. ACM (2014)
12. Joyeux, S., Albiez, J.: Robot development: from components to systems. In: 6th National Conference on Control Architectures of Robots (2011), http://hal.inria.fr/inria-00599679/
13. Joyeux, S., Kirchner, F., Lacroix, S.: Managing plans: Integrating deliberation and reactive execution schemes. Robotics and Autonomous Systems 58(9), 1057–1066 (sep 2010), http://linkinghub.elsevier.com/retrieve/pii/S0921889010001090
14. Joyeux, S., Schwendner, J., Roehr, T.M.: Modular software for an autonomous space rover. In: Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space. i-SAIRAS (2014), 8 pages
15. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: Proceedings of the International Conference on Computer Vision. pp. 2106–2113. IEEE (2009)
16. Jung, Y., Kang, D., Kim, J.: Upper body motion tracking with inertial sensors. In: Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on. pp. 1746–1751 (Dec 2010)

17. Kagermann, H., Helbig, J., Hellinger, A., Wahlster, W.: Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group. Acatech – National Acadamy of Science and Engineering (2013)
18. Klein, F., Spieldenner, T., Sons, K., Slusallek, P.: Configurable instances of 3d models for declarative 3d in the web. In: Proceedings of the Nineteenth International ACM Conference on 3D Web Technologies. ACM International Conference on 3D Web Technology (Web3D-14), 19th, August 8-10, Vancouver,, BC, Canada. pp. 71–79. ACM (2014), http://doi.acm.org/10.1145/2628588.2628594
19. Krieger, H.U.: An efficient implementation of equivalence relations in OWL via rule and query rewriting. In: Proceedings of the IEEE International Conference on Semantic Computing (ICSC'13). pp. 260–263. IEEE (2013)
20. Krieger, H.U.: Integrating graded knowledge and temporal change in a modal fragment of OWL. In: van den Herik, J., Filipe, J. (eds.) Agents and Artificial Intelligence. Springer, Berlin (2016), in press
21. Larsson, S., Traum, D.R.: Information state and dialogue management in the TRINDI dialogue move engine toolkit. Natural Language Engineering 6(3&4), 323–340 (2000)
22. Leitão, P.: Agent-based distributed manufacturing control: A state-of-the-art survey. Eng. Appl. Artif. Intell. 22(7), 979–991 (Oct 2009)
23. Lemburg, J., Mronga, D., Aggarwal, A., de Gea Fernández, J., Ronthaler, M., Kirchner, F.: A robotic platform for building and exploiting digital product memories. In: Wahlster, W. (ed.) SemProM – Foundations of Semantic Product Memories for the Internet of Things, pp. 91–106. Cognitive Technologies, Springer (2013)
24. Levesque, H.J., Cohen, P.R., Nunes, J.H.T.: On acting together. In: Proceedings of the Eighth National Conference on Artificial Intelligence - Volume 1. pp. 94–99. AAAI'90, AAAI Press (1990), http://dl.acm.org/citation.cfm?id=1865499.1865513
25. Likhachev, M., Ferguson, D., Gordon, G., Stentz, A., Thrun, S.: Anytime search in dynamic graphs. Artificial Intelligence 172(14), 1613–1643 (2008)
26. Liu, L., Shell, D.A.: Optimal market-based multi-robot task allocation via strategic pricing. In: Newman, P., Fox, D., Hsu, D. (eds.) Robotics: Science and Systems IX, Technische Universität Berlin, Berlin, Germany, June 24 - June 28, 2013 (2013), http://www.roboticsproceedings.org/rss09/p33.html
27. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2), 91–110 (2004)
28. Mourkani, S.S., Bleser, G., Schmitz, N., Stricker, D.: A low-cost and light-weight motion tracking suit. In: Proceedings of the International Conference on Ubiquitous Intelligence and Computing. pp. 474–479. IEEE (2013)
29. Nguyen, M.H., Wobcke, W.: AI 2006: Advances in Artificial Intelligence: 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, December 4-8, 2006. Proceedings, chap. A Flexible Framework for SharedPlans, pp. 393–402. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
30. Pirkl, G., Hevesi, P., Cheng, J., Lukowicz, P.: mBeacon: Accurate, robust proximity detection with smart phones and smart watches using low frequency modulated magnetic fields. In: Proceedings of the 10th EAI International Conference on Body Area Networks. pp. 186–191. ICST (2015)
31. Pirkl, G., Lukowicz, P.: Robust, low cost indoor positioning using magnetic resonant coupling. In: Proceedings of the International Conference on Ubiquitous Computing. pp. 431–440. ACM (2012)

32. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to sift or surf. In: Proceedings of the International Conference on Computer Vision (ICCV'11). pp. 2564–2571. IEEE (2011)
33. Ruppenhofer, J., Ellsworth, M., Petruck, M.R., Johnson, C.R., Scheffczyk, J.: FrameNet II: Extended theory and practice. Tech. rep., International Computer Science Institute (ICSI), University of California, Berkley (2006)
34. Scerri, P., Pynadath, D., Johnson, L., Rosenbloom, P., Si, M., Schurr, N., Tambe, M.: A prototype infrastructure for distributed robot-agent-person teams. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems. pp. 433–440. AAMAS '03, ACM, New York, NY, USA (2003)
35. Schwartz, T., Feld, M., Bürckert, C., Dimitrov, S., Folz, J., Hutter, D., Hevesi, P., Kiefer, B., Krieger, H.U., Lüth, C., Mronga, D., Pirkl, G., Röfer, T., Spieldenner, T., Wirkus, M., Zinnikus, I., Straube, S.: Hybrid teams of humans, robots and virtual agents in a production setting. In: Intelligent Environments (IE), 2016 International Conference on. p. accepted (2016)
36. Schwendner, J., Roehr, T.M., Haase, S., Wirkus, M., Manz, M., Arnold, S., Machowinski, J.: The Artemis rover as an example for model based engineering in space robotics. In: Workshop Proceedings of the International Conference on Robotics and Automation (ICRA-2014). IEEE (2014), 7 pages
37. Settembre, G.P., Scerri, P., Farinelli, A., Sycara, K., Nardi, D.: A decentralized approach to cooperative situation assessment in multi-robot systems. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1. pp. 31–38. AAMAS '08, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2008), http://dl.acm.org/citation.cfm?id=1402383.1402393
38. Snodgrass, R.T.: Developing Time-Oriented Database Applications in SQL. Morgan Kaufmann, San Francisco, CA (2000)
39. Sons, K., Klein, F., Rubinstein, D., Byelozyorov, S., Slusallek, P.: XML3D: Interactive 3D Graphics for the Web. In: Proceedings of the 15th International Conference on Web 3D Technology. pp. 175–184. Web3D '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1836049.1836076
40. Straube, S., Schwartz, T.: Hybrid teams in the digital network of the future – application, architecture and communication. Industrie 4.0 Management 2, 41–45 (2016)
41. Tambe, M.: Towards flexible teamwork. J. Artif. Int. Res. 7(1), 83–124 (Sep 1997), http://dl.acm.org/citation.cfm?id=1622776.1622781
42. Täubig, H., Frese, U.: A new library for real-time continuous collision detection. In: Proceedings of the 7th German Conference on Robotics. pp. 108–112. VDE (2012)
43. Van Dyke Parunak, H., Baker, A.D., Clark, S.J.: The aaria agent architecture: An example of requirements-driven agent-based system design. In: Proceedings of the First International Conference on Autonomous Agents. pp. 482–483. AGENTS '97, ACM, New York, NY, USA (1997)
44. Wirkus, M.: Towards Robot-independent Manipulation Behavior Description. In: Proceedings of the 5th International Workshop on Domain-Specific Languages and models for ROBotic systems. arxiv.org, Bergamo, Italy (2014)
45. Zhang, J., Sclaroff, S.: Exploiting surroundedness for saliency detection: A boolean map approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(5), 889–902 (2016)