A Hybrid Approach to Activity Recognition of Humans in a Human-Robot Rescue Team

Author:
Bas BOOTSMA
s.c.u.bootsma@student.ru.nl

Internal supervisor:
Dr. Pim HASELAGER
External supervisor:
Nanja SMETS M.Sc.

Master thesis presented for the degree of
Master of Science

October 4, 2015
Artificial Intelligence, Department of Social Sciences

Contents

List of Figures

List of Tables

Preface

Doing an internship at TNO, and writing this thesis has been quite a learning experience, not only at a academic level, but also an a personal level. So, I am grateful for all the persons who supported me during the process. The order in which persons appear does not relate to the amount of gratitude.

I would like to express my deepest gratitude to Nanja with whom I had many brainstorms, and who supported me during the entire process at TNO. I would also like to thank Tina for temporarily being my supervisor, and providing constructive criticism on my topic. And, I would like to thank all the interns (Pim, Jelte, Ramona, Serena, Owren, Mike, and others) for making sure coming to TNO was a lot of fun (e.g., table tennis matches, the volleyball tournament, etc.). I would like to thank Jelte in particular for observing during T-JEx which provided the necessary results with which I could evaluate my system. Also, I would like to thank Ramona for translating the interviews. Furthermore, I would like to thank all the other employees at TNO who contributed in any way (e.g., providing smart phones, ideas, etc.) to my thesis.

I would like to thank internal supervisor Pim for asking all the difficult questions I did not think of, or did not want to answer, and provide me with the necessary insights that supported me during the writing of my thesis.

I would also like to thank Carel for providing a place to sleep during the week, share meals, and stories with.

I would like to thank Maaike for supporting me during the entire process, and persevere while I was gone most of the week during my internship.

Also, I would like to thank all the other people I did not think of that contributed in any way to my thesis.

1. Introduction

Urban search and rescue (USAR) is the emergency response involving the location, and extraction of victims trapped in confined spaces (Federal Emergency Management Agency 2015). Often the entrapment occurs due to the collapse of man-made structures, but transportation or mining accidents are also possible. Urban search and rescue is a challenging and hazardous domain. Environments are dangerous and unpredictable, due to fires, chemical leaks, or structurally unsafe surroundings. The rescue workers are often deprived of sleep, stressed, and under time pressure (Murphy 2004). All these aspects ensure the work is not only physically, but also cognitively demanding.

Due to the various challenges within urban search and rescue robots can provide a useful contribution. Robots are able to enter voids too narrow for rescue workers, or explore structurally or environmentally unsafe surroundings (e.g., danger of collapse, fire, etc.) (Casper and Murphy 2003). While robots are able to provide many benefits to rescue workers, various challenges remain in perception, mobility, but also in human-robot interaction. Robot operators have difficulties being aware of the environment, for example, problems detecting victims, or unable to estimate whether rubble is crossable. Also, due to improvements in robot autonomy it is important to understand the decision-making process of the robot. Various approaches to support human-robot rescue teams exist, for example, adaptive automation (e.g., Kaber and Endsley (2004)), dynamic task allocation (e.g., Lerman (2006)), and shared mental models (e.g., Giele et al. (2015)). However, all these approaches rely on knowing what is going on, which includes knowledge about the task, the environment, and the agents. Also, in most cases assumptions about various aspects of knowledge are made in order to reduce the dependency on information. Human activity recognition in a human-robot rescue team tries to provides part of that knowledge and lessen the need for some assumptions. Human activity recognition is the process of recognizing human activities based on observations about a human, and the environment using an automated system. So, the following question is to be answered:

> Can human activities be recognized in real-time for a human-robot rescue team?

Also, since collaboration between humans is an important aspect in rescue work, the following sub-question is to be answered:

> Can both individual and team activities be recognized?

In order to provide answers to the research questions, this thesis presents a hybrid approach to activity recognition of humans in a human-robot rescue team. The hybrid approach combines data-driven and knowledge-driven techniques into a single framework (similar to Riboni and Bettini (2010)). The data-driven techniques provide various sources of information, while the knowledge-driven techniques integrates that information into a coherent structure and provide reasoning capabilities. The sources of information include three human behaviours, namely physical motion, communication, and interface actions, and additional information related to the hierarchical structure of a human-robot rescue team. The activity recognition system is evaluated during a high-fidelity exercise with actual fire fighters, of which the results are compared with two other approaches, namely the most common activity, and random activities. The most common activity approach predicts the most common activity, defined as the most occurring, and activity with the longest duration, at any time. The random activities approach predicts one or multiple random activities at any time. The activity recognition system should at a minimum outperform those approaches. Ultimately, I will argue that while doing activity recognition just for the sake of activity recognition is scientifically relevant, it is difficult to determine which evaluation metrics are important without knowing what the recognized activities are going to be used for.

The remainder of the thesis is organized as follows:

- Chapter 2 Background provides an overview of urban search and rescue, research related to human-robot teaming, activity recognition, and several design principles.
- Chapter 3 Model provides a conceptual overview of the activity recognition system.
- Chapter 4 Architecture outlines the underlying implementation of the different components of the activity recognition system.
- Chapter 5 Evaluation provides an overview of the evaluation of the activity recognition system.
- Chapter 6 Results provides an overview of all the results based on the evaluation.
- Chapter 7 Discussion provides a reflection on the results, and discusses the hybrid approach to activity recognition.

2. Background

The literature in the following chapter provides the context in which this research has been conducted. Due to the interdisciplinary nature of this research a wide range of topics is discussed:

- Section 2.1 Urban Search and Rescue provides a description of urban search and rescue with a focus on rescue robotics, and the envisioned human-robot rescue team according to the TRADR project.
- Section 2.2 Activity Recognition provides an extensive overview of different approaches to activity recognition with an important distinction between data- and knowledge-driven methods.
- Section 2.3 Design Principles provides an overview of various socio-technical and technical design principles with regard to activity recognition systems.
- Section 2.4 Summary provides a summary specifying which aspects are considered relevant for this project.

2.1. Urban Search and Rescue

The following section is intended to provide the basis to relate all other topics in the background to. A brief overview of robotics and issues related to human-robot interaction in an urban search and rescue context is provided. Also, an overview is given of the TRADR project which tries to support urban search and rescue forces using robots and technology.

The Federal Emergency Management Agency (FEMA) describes urban search and rescue as follows (Federal Emergency Management Agency 2015):

> "Urban search-and-rescue (US&R) involves the location, rescue (extrication), and initial medical stabilization of victims trapped in confined spaces. Structural collapse is most often the cause of victims being trapped, but victims may also be trapped in transportation accidents, mines and collapsed trenches.
>
> Urban search-and-rescue is considered a 'multi-hazard' discipline, as it may be needed for a variety of emergencies or disasters, including earthquakes, hurricanes, typhoons, storms and tornadoes, floods, dam failures, technological accidents, terrorist activities, and hazardous materials releases. The events may be slow in developing, as in the case of hurricanes, or sudden, as in the case of earthquakes."

Urban search and rescue is a domain with many challenges and hazards. Environments are dangerous and unpredictable due to various dynamic events, such as aftershocks in case of an earthquake, or chemical fires. Collapsed buildings are often structurally unsafe, and are difficult to traverse searching for victims.

2.1.1. Human-robot Interaction in Rescue Contexts

Due to the challenges of urban search and rescue robots are able to provide a useful contribution. Robots are able to access a wide variety of areas otherwise inaccessible to humans. For example, voids can be explored due to the smaller size of a robot, or robots are unimpaired by particular hazardous factors (e.g., smoke, fires, or chemical leaks) due to protective casings. Also, while expensive, in general robots are considered more expendable than human lives. While robots in urban search and rescue missions provide many benefits to rescue workers, various issues are still unresolved. These include technical issues (e.g., low battery life, unreliable wireless communications, or limited autonomy), but also issues related to the interaction between humans and robots. Robot operators controlling the robots have difficulties being aware of the environment surrounding the robot. For example, it is difficult to determine whether a robot is able to enter a cavity, or whether an obstacle can be crossed. Also, interpreting the multitude of sensor information robots are able to perceive is difficult, while technological advances are able to provide support in this area (e.g., automated victim detection). While a complete listing of issues related to human-robot interaction is outside the scope of this thesis, extensive overviews of human-robot interaction within the domain of urban search and rescue are presented by Casper and Murphy (2003), and Murphy and Burke (2005). A recent analysis of human-robot interaction at the DARPA Robotic Challenge Trials, designed to evaluate humanoid robots in a disaster response scenario, is presented by Yanco et al. (2015).

In the following paragraph various approaches to supporting human-robot interaction are discussed. Adaptive automation influences the level of automation of control over robots between humans and an automated system. Adaptive automation reduces workload, and increases situation awareness, in human-robot teams (Kaber and Endsley 2004; Parasuraman, Barnes, et al. 2007; Parasuraman, Cosenzo, and Visser 2009; Visser and Parasuraman 2011). Related to adaptive automation is dynamic task allocation in which tasks are distributed to agents based on various criteria. Even though dynamic task allocation in general assumes systems with homogeneous agents (e.g., only robots, for example Lerman (2006)), the method is applicable to systems with heterogeneous agents. Giele et al. (2015) proposed a framework to improve team performance in mixed human-robot teams, providing promising results. Another approach uses shared mental models to provide insight into team functioning by capturing particular notions of teamwork. In human-robot teams shared mental models are important for situation awareness, and team effectiveness (Burke and Murphy 2004; Murphy and Burke 2005).

All approaches supporting human-robot teamwork rely on knowing what is going on, which includes knowledge about the task, the environment, and the agents. Also, many approaches make assumptions (e.g., using wizard-of-oz experiments) about certain aspects in order to reduce the dependency on information. However, activity recognition of humans in a human-robot rescue team tries to provide part of that knowledge and lessen the need for some assumptions.

2.1.2. Project: TRADR

Long-Term Human-Robot Teaming for Robot Assisted Disaster Response (TRADR) is a European funded research project which develops novel science and technology for human-robot teams to assist in disaster responses. A user-centric design methodology (i.e., situated Cognitive Engineering (sCE- method), discussed in Section 2.3.2) is employed to support the development of technological systems. The TRADR project expands upon the earlier NIFTi project (Kruijff et al. 2014) by providing persistence of all aspects of a human-robot team across multiple sorties during a mission. Therefore, the following three main scientific objects have been formulated:

1. **Persistent environment model**: Models of dynamic environments through the fusion of multi-modal information based on observations of multiple robots collected across multiple sorties during a mission.

2. **Persistent models for multi-robot acting**: Models of individual and multi-robot planning and execution across multiple sorties during a mission.

3. **Persistent models for human-robot teaming**: Models of collaboration between humans and robots to improve alignment of mutual expectations across multiple sorties during a mission.

Even though robots have been deployed in urban search and rescue missions, due to the many challenges and lack of standardization no formal definition of a human-robot rescue team exists. The TRADR project envisions a particular composition of humans, and robots within a human-robot rescue team using various types of technology. Even though the composition of a human-robot rescue team is continuously being refined, the following section provides an overview of the most relevant concepts related to a human-robot rescue team.

Human-robot Rescue Team

A human-robot rescue team consists of several human team members within various roles, namely: a team leader, multiple robot operators, multiple in-field rescuers, and various robots (e.g., unmanned ground vehicles (UGVs), and unmanned air vehicles (UAVs)). Each member (both humans and robots) has different attributes, capabilities, and responsibilities. In the following sections an overview of the most important characteristics of each role within a human-robot rescue team is provided.

Team leader The team leader is responsible for managing the individual members of the team. Using a tactical display system the team leader is able to receive information about the mission. Using direct verbal communication through two-way radio devices (e.g., walkie- talkies) with other human team members, the team leader is able to direct the mission according to protocol and personal insight.

Robot Operator The robot operator is responsible for controlling one or multiple robots (i.e. a UGV, and a UAV). Positioned behind an operator control unit (OCU) the robot operator is able to send commands to the robot and receive information via the sensors of the robot (e.g., video images, or point clouds). The robot operator remains in a fairly static position, but occasionally leaves its post to, for example, interact with the robot directly (e.g., unpacking, or cleaning the robot).

In-field Rescuer The in-field rescuer is responsible for providing additional support, especially in situations in which the robots are incapable of providing support (e.g., a live victim). Equipped with a hand-held device (e.g., tablet), and a two-way radio device the in-field rescuer is located near or at the actual disaster area. The in-field rescuer is able to communicate with the team leader, and is able to receive and submit information via the hand-held device.

Robots Within TRADR two types of robots exist, namely UGV's (BlueBotics 2015), and UAV's (AscTec 2015). The UGV's are intended for the exploration of the disaster area from the ground. They are able to enter structures, or voids in collapsed structures, and provide video images from RGB-D camera's. They can be equipped with various sensors to measure different types of information, such as heat, chemicals, etc. Furthermore, the UGV's can be outfitted with a robotic arm (Kinova Robotics 2015) which can be used to manipulate objects, for example, to open doors, or collect samples.

UAV's are intended to provide an overview of the disaster area using video from normal, or infrared camera's. They are able to fly at a high altitude, or even enter buildings if there is sufficient room to fly.

Technology Technology refers to the all the tools (both hardware and software) used to support the human members of the human-robot rescue team. Within the TRADR project two types of software tools are employed, namely the Tactical Display System, and the Operator Control Unit. The Tactical Display System (TDS) is intended to provide all human team members with information to support work-related activities. For example, the TDS contains a map of the current location with markers regarding important information (e.g., obstacles, victims, etc.). An example of an functional TDS used in the NIFTi project is described by Diggelen, Grootjen, and Ubink (2013). The Operator Control Unit (OCU) is intended to allow a user to control one or multiple robots. Therefore, it provides functionality to control a robot (and any additional devices, e.g., a robotic arm) and receive sensor information from the robot. The sensor information includes, for example, camera images of the robot, and orientation of the robot. Since multiple types of robots are used in the TRADR project the OCU has to be flexible enough to support both types.

2.2. Activity Recognition

In the following section an overview is presented of different approaches to human activity recognition. Due to the abundance of activity recognition research several authors provide a categorization of approaches to activity recognition literature. Therefore, I present a succinct overview of different categorizations of the approaches to human activity recognition.

Turaga et al. (2008) present an overview of a categorization approaches to activity recognition which are categorized according to *methods that rely on human body models* and *methods that do not rely on human body models*. In *methods that rely on body models* the categorization of approaches is as follows: *a*) non-parametric (e.g., 2D-templates, 3D object models); or *b*) volumetric (e.g., spatio-temporal filtering, sub-volume matching); or *c*) parametric (e.g., hidden Markov models, linear dynamical systems). In *methods that do not rely on body models* the categorization of approaches is as follows: *a*) graphical models (e.g., Bayesian networks); or *b*) syntactic (e.g., context-free grammars); or *c*) knowledge and logic-based (e.g., ontologies).

In Aggarwal and Ryoo (2011) activity recognition is viewed from the perspective of computer vision. Based on the complexity of an activity, the following division is made: gestures, actions, interactions, and group activities. Activities are categorized according to a hierarchical taxonomy with a major distinction between: *1*) single-layered approaches; and *2*) hierarchical approaches. Single-layered approaches recognize activities directly based on sequences of images, and are suitable for the recognition of lower level activities, e.g., gestures, and actions. Single-layered approaches are classified into space-time approaches (i.e., space-time volume, trajectories, and space-time features), and sequential approaches (i.e., exemplar-based, and state model-based). Hierarchical approaches (i.e., statistical, syntactic, and description-based) represent higher level activities, e.g., interactions, and group activities, based on a combination of lower level activities.

L. Chen, Nugent, and Wang (2012) present an overview of a categorization of activity recognition with a major distinction between: *1*) *vision-based* versus *sensor-based*; or *2*) *data-driven* versus *knowledge-driven*. In order to recognize activities vision-based approaches use visual information (e.g., video images), while sensor-based approaches use data from various sensors (e.g., accelerometer, RFID tags). Data-driven approaches build implicit models through various machine learning techniques, and can be categorized as follows: *a*) generative modelling; or *b*) discriminative modelling; or *c*) heuristic/other (combination of the previous categories). Knowledge-driven approaches build explicit models using expert knowledge, and can be categorized as follows: *a*) mining-based; or *b*) logic-based; or *c*) ontology-based.

All surveys provide a thorough overview of the different approaches to activity recognition. In Turaga et al. (2008) and Aggarwal and Ryoo (2011) vision-based approaches are an important aspect of the categorization, but are less important in our case, since vision-based approaches are impractical in an urban search and rescue context. This is due to the fact that, in general, vision-based approaches assume a static environment with fixed positions for the camera's, which are unrealistic assumptions in an urban search and rescue context. Therefore, the data-driven versus knowledge-driven categorization of L. Chen, Nugent, and Wang (2012) is used to provide a brief explanation per category followed by an overview of the relevant literature if applicable.

2.2.1. Data-Driven

Data-driven approaches build implicit models through various machine learning techniques, and are categorized as followed: *a*) generative modelling; or *b*) discriminative modelling; or *c*) heuristic/other (combination of the previous categories).

Generative Modelling

Generative models provide a probability distribution over the observations (e.g., sensor data) and the labels (i.e. activities) (Jordan 2002). In order to generate predictions with a generative model a conditional probability density function is computed from the joint probability distribution, which provides the probability of a label given an observation. Common classifiers used for learning general models include naïeve Bayes classifiers (NCB), and hidden Markov models (HMM). To train an adequate model it is necessary to have sufficient data of all observations in order to estimate a probability distribution. Nonetheless, generative models have been successfully used in different applications of activity recognition.

In Bao and Intille (2004) a wide variety of activities were recognized using five bi-axial accelerometers. Participants were tasked to perform activities in a semi-naturalistic setting during everyday life. Each day participants had to attach the accelerometers on particular positions and perform a random sequence of 20 predefined activities. Using the collected data several classifiers (e.g., C4.5 decision tree, naive Bayes, etc.) were trained and tested using various types of cross-validation. An overall recognition rate of 84.26% was achieved using a decision tree.

In a similar domain, but using a different approach, Patterson et al. (2005) provide activity recognition through abstract object usage. In a kitchen environment a total of 60 objects were outfitted with an RFID tag. Each object was related to one or more of eleven activities, i.e., using the bathroom, making oatmeal, making soft-boiled eggs, preparing orange juice, making coffee, making tea, making or answering a phone call, taking out the trash, setting the table, eating breakfast, and clearing the table. A single person was equipped with two gloves able to detect whenever an object with a RFID tag was within a certain range. The following four models were trained and evaluated (accuracy in brackets) with each other, namely: independent hidden Markov models (68%), connected hidden Markov models (88%), object-centred hidden Markov models (87%), and dynamic Bayes net with aggregates (88%).

Discriminative Modelling

In contrast with generative models, discriminative models provide only a conditional probability density function of a label given an observation, or a direct mapping from observations to labels. Using a discriminative model it is possible to generate predictions directly. Examples of classifiers used for training discriminative models include nearest neighbour, decision trees, and support vector machine.

In Ravi et al. (2009) eight activities, i.e., standing, walking, running, climbing up stairs, climbing down stairs, sit-ups, vacuuming, and brushing teeth, were recognized using a single tri-axial accelerometer worn around the participants waist. Two participants performed the activities in multiple rounds over different days. A total of twelve features, i.e., mean, standard deviation, energy, and correlation, for all three axes of the accelerometer are extracted. The data was evaluated using a total of 18 classifiers, including decision tables, decision trees, support vector machines, or naive Bayes, using various types of cross-validation. For three out of the four cross-validation types plurality voting has the highest accuracy with at least 90.61%, and in the other type of cross-validation boosted SVM has the highest accuracy with 73.33%.

In a similar approach by Kwapisz, Weiss, and Moore (2011) six activities, i.e., walking, jogging, ascending stairs, descending stairs, sitting, and standing, were recognized using cell phone accelerometers. The participants carried a smart phone in their front pants leg pocket during their everyday activities, and were tasked to perform the activities for specific periods of time. The data was modelled using a number of classifiers, namely: decision trees, logistic regression, and multi-layer perceptrons. On all activities, except ascending and descending stairs, the prediction rate of the models was 89.9% or higher. The prediction rate of the activities ascending and descending stairs ranged between 12.3% and 61.5%, but after combining the activities the prediction rate 77.6% for the decision tree model.

In Wyss and Mäder (2010) various military-specific activities, i.e., walking, marching with backpack, lifting and lowering loads, lifting and carrying loads, digging, and running, were recognized using multiple body-fixed sensors. The sensors were able to measure waist acceleration in vertical direction, step frequency, heart rate, and whether a backpack was being carried. Using a manually constructed decision tree model the system was evaluated in both a laboratory setting as well as during a military training session. In the laboratory setting the overall activity recognition was $87,5\%$, and during the military training session $85,5\%$.

Heuristic/other Models

A heuristic approach to activity recognition uses a combination of generative models, discriminative models, and heuristic information.

In Zhu and Sheng (2009) a variety of activities, e.g., running, walking, standing, walking downstairs, were recognized using two accelerometers, placed on the waist and foot. The data was processed in two steps, namely: coarse-grained classification, and fine-grained classification. In the coarse-grained classification one neural network for each sensor to output the type of an activity, i.e., stationary, transitional, and cyclic. In the fine-grained classification the type of the activity determined whether heuristic discrimination, or a hidden Markov model was applied. Heuristic discrimination was applied to activities with no displacement, e.g., standing, or sitting, while the hidden Markov model was applied to activities with strong displacements, e.g., walking, or running. The hidden Markov model provides an accuracy of 0.8701 or higher for four activities.

A hybrid discriminative/generative approach is discussed by Lester, Choudhury, and Kern (2005). A single device with eight different sensors, i.e. accelerometer, audio, IR/visible light, high-frequency light, barometric pressure, humidity, temperature, and compass, was used to collect the data. Two participants performed a total of ten activities, i.e., sitting, standing, walking, jogging, walking up stairs, walking down stairs, riding a bicycle, driving car, riding elevator down, and riding elevator up, over a period of six weeks. From the data over 600 hundred features were computed which were used in an activity recognition pipeline. For each activity a decision-stump classifier is applied to the feature vector, which produces a sequence of decision stump margins for each feature. From the decision stump margins a sequence of posterior probabilities is computed and used as input for a hidden Markov model. An activity is predicted by taking the maximum likelihood of the output of all hidden Markov models. By using a combination of decision stumps and hidden Markov models a precision of 99%, and a recall of 91% is achieved.

Hakeem and Shah (2004) provide another example of a combination of multiple approaches, but is discussed in another section.

2.2.2. Knowledge-Driven

Knowledge-driven approaches build explicit models using expert knowledge, and can be categorized as follows: *a*) mining-based; or *b*) logic-based; or *c*) ontology-based.

Mining-Based

Mining-based approaches take advantage of the wealth of information available from public sources. Examples of public sources are recipes, training manuals, and how-tos. The approach tries to identify associations between activities and object usage. In many cases similar techniques as in generative and discriminative modelling are used, since most techniques need a lot of data in order to build adequate models.

Mining-based approaches are unlikely to be of use for determining activities of human-robot rescue teams. Even though data of common activities (e.g., walking, sitting, or climbing stairs) is available through various data sets, the urban search and rescue domain is very specific with the usage of particular objects and activities, meaning public information specific for the subject is unlikely to be available. Therefore, mining-based approaches will not be discussed here.

Logic-Based

Logic-based approaches try to model activity recognition using a logical formalism. Capturing domain knowledge in logical rules it is possible to reason about the statements in the knowledge. Therefore, logic-based approaches provide certain features, such as prediction, and explanation.

In Shet, Harwood, and Davis (2005) a multi-level approach is used in the domain of surveillance, in which multiple cameras monitored the lobby of a building. On the lowest level various features (e.g., an object's path) were extracted from the video images. On the middle level the features were used to generate facts, such as a person has been found standing near a vending machine at a specific point in time. On the highest level a reasoning module evaluates the facts according to specific rules in order to recognize various activities, such as theft, or entry violation. As a proof-of-concept the system was implemented and evaluated on a computational level, which shows a linear increase in computation time with respect to the number of facts.

Ontology-Based

Ontology-based approaches define activities using an explicit representation in an ontology. An ontology is a data structure in which entities and relations between the entities are captured. Other approaches often have difficulty in generalizing from particular instances and require specific data to capture all characteristics. Ontology-based approaches are more easily applied to different instantiations in the same domain, and are in some cases even applicable to other domains.

In Riboni, Pareschi, et al. (2011) ontology-based activity recognition is viewed from a meta perspective. A generative modelling approach (i.e. a hidden Markov model (HMM)) is compared with an ontology-based approach with respect to temporal reasoning. The hidden Markov model and various modelled ontologies with explicit temporal relations were applied to the domain of daily living. A data set containing annotated activities of a single person living in a smart-home for 28 days was used to compare the different approaches. The results show that the hidden Markov model outperforms the default ontology in which no past temporal information was included. However, when past temporal information was included in the ontology, the recognition performance is higher when compared to the hidden Markov model. It is also important to note that in ontologies with temporal information included it is assumed the user only performs a single activity at a time, due to practical reasons.

Another way of combining ontologies with temporal information is described by D. Chen, Yang, and Wactlar (2004), in which social interactions in a nursing home environment are modelled. Various features (e.g., speed, distance, and relative direction of entities) from multiple video images are extracted and manually annotated with the activities according to the ontology. By mapping the ontology onto a dynamic Bayesian network (DBN) it is possible to represent the relations between the entities (e.g., person, or wheelchair) and aspects of the temporal dimension. Evaluation of the model yields a recognition rate is 75% and higher for five out of six social interactions. The other social interaction *greeting* needs more precise or other features to yield a better recognition rate.

In Hakeem and Shah (2004) activity recognition in the context of meetings is discussed using multiple approaches. A meeting ontology is described as a multi-level hierarchy, of three levels, of concepts of which each layer is mapped onto a different approach. The lowest layer contains movements of human body parts (e.g., hands, or head) which are detected using various features based on video images. The middle layer contains events (i.e., a sequence of movements over time) which are detected using a finite state machine (FSM). The highest layer contains behaviours (i.e., a set of events over time) which are modelled using a rule-based system (RBS). Using 12 meeting videos the evaluation results precision values of 76.47% and higher, and recall values of 90.91% and higher.

In order to provide a foundation among different domains, a top-level ontology of activity recognition in the context of smart environments is provided by Ye, Stevenson, and Dobson (2011). Smart environments are spaces in which sensors, actuators, and displays are embedded seamlessly in everyday objects. Applications of smart environments include for example homes, offices, and healthcare facilities. The authors make a distinction between domain ontologies and application ontologies, and provide a generic framework of activity recognition in which the commonalities between the different applications are captured. The framework consists of multiple elements, namely: a concept model, a context model, and a activity model. The concept model captures the structure of the information spaces, such as location, time, distance, etc. The context model captures context predicates, which encapsulate the relation between two abstract values belonging to an information space. The activity model captures aspects related to activities or states of a user, which are possibly interesting to particular applications. In all models various definitions, relationships, and properties are defined. Combining the framework with an application ontology provides additional aspects without the need to explicitly specify these aspects in the application ontology.

Riboni and Bettini (2010) describes a generic system which combines statistical methods with ontological reasoning. Input into the system is provided via body-worn sensors, environmental sensors, and sensors on the mobile device. On the mobile device the data from the sensors are fused and provided to a single system composed of several modules. On the mobile device the data from the sensors is fused, and provided to multiple modules. The pattern recognition module provides statistical activity models. The geographic information system module provides location-based on the user's location (provided by the embedded sensors, and environmental sensors). The ontological reasoner module provides potential and complex activities. The system combines the information from all modules and outputs the recognized activities. In an evaluation of the system multiple humans had to perform ten different indoor and outdoor activities, with a bias towards physical activities. The system achieved a recognition rate of 93.44% which is higher than the best statistical classifier, namely multi-class logistic regression with a recognition rate of 80.21%.

2.3. Design Principles

When designing an activity recognition system various aspects have to be taken into consideration. For example, existing technologies, usability aspects, evaluation aspects, or particular standards.

- Section 2.3.1 Robot and Technology Standards standards regarding robots, and technology are described.
- Section 2.3.2 Situated Cognitive Engineering Method describes a method to support the development of the activity recognition system.
- Section 2.3.3 Technological describes a particular aspect of the development of the activity recognition system, namely the technological aspects.

2.3.1. Robot and Technology Standards

In the context of urban search and rescue all personal, and equipment have high minimum standards[1] in order to cope with the rescue work. Similar standards should also apply to the robots, and technology

[1]For example, the National Fire Protection Association (NFPA) provides over 300 documents related to fire safety (http://www.nfpa.org/codes-and-standards/document-information-pages).

developed within the TRADR project, including the activity recognition system. While organizations such as the Federal Emergency Management Agency (FEMA) (Federal Emergency Management Agency 2015), the National Institute of Standards and Technology (NIST) (National Institute of Standards and Technology 2015) try to provide certain standards (for example: Messina and Jacoff (2006)), many of them are only related to rescue work in general, or consider only particular aspects (such as the robots). Also, with multiple standards out there no general consensus is reached. Therefore, I would like to introduce two characteristics to which the technology should adhere to, namely reliability, and robustness. Reliability encompasses the dependability of the technology under the stated conditions for a particular period of time. Robustness is the ability to cope with errors, or abnormalities of the input. For example, the activity recognition system is reliable if it is always able to detect a predefined percentage of the activities, and robust if it is able to cope with unexpected actions.

2.3.2. Situated Cognitive Engineering Method

The situated Cognitive Engineering (sCE) method (Neerincx and Lindenberg 2008) is aimed at supporting the development of human-centered automation. The sCE-method structures and guides the evaluation and documentation of technological designs by enabling the sharing and re-usage of design knowledge in a multi-disciplinary community. The following three main segments are defined:

A. Foundation

B. Specification

C. Evaluation

Within TRADR the sCE-method is applied to iteratively design the technology as described in Section 2.1.2. While not all aspects of the methodology have been adhered to during the development of the activity recognition system, the methodology did provide a useful framework to aid the development. Therefore, the three main segments of the sCE-method are discussed in order to provide a brief overview of the way the development of the activity recognition system has been supported.

Foundation

In segment A. Foundation the design rational is described in terms of *a*) operational demands; *b*) relevant human factors knowledge; and *c*) envisioned technologies.

The *a*) operational demands describe the current practice without the envisioned technologies. The *b*) relevant human factors knowledge describe the available knowledge concerning human factors, such as, functional design, task support, ergonomics, etc. The *c*) envisioned technologies describes the options of using existing technology or the lack thereof (meaning novel technology has to be developed.

Specification

In segment B. Specification the solution is specified in terms of relevant human factors knowledge, and the envisioned technologies. The specification consists of *a*) design scenarios; *b*) actors and use cases; *c*) requirements; *d*) claims; and *e*) ontology. The *a*) design scenarios prescribe the specification using short stories describing the way the user will work with the system (showing the benefits of the solution). The *b*) actors and use cases provide a step-by-step interactions between the relevant actors and the system. The *c*) requirements describe the specific functionality the system should provide to the user (derived from the use cases). The *d*) claims describe the relation between the requirements and the hypotheses to be tested during evaluations. The *e*) ontology captures all concepts and relations relevant to the system, to provide consistent semantics throughout the development.

Evaluation

Segment C. Evaluation prescribes aspects related to the evaluation of the system in terms of *a*) artefact; *b*) evaluation method; and *c*) evaluation results. The *a*) artefact is a prototype incorporating a particular set of requirements, design patterns, and technologies. The *b*) evaluation method describes the type of evaluation (e.g., human-in-the-loop study, or use-case-based simulation) employed to evaluate the artefact. The *c*) evaluation results describe the results of the evaluation, used to further specify the specification of the system.

2.3.3. Technological

The following section provides an overview of the relevant technological aspects when designing an activity recognition system. In Lara and Labrador (2013) the following aspects are considered: *1*) selection of attributes; *2*) obtrusiveness; *3*) data collection protocol; *4*) recognition performance; *5*) energy consumption; *6*) processing; and *7*) flexibility. While not all aspects are equally important for activity recognition in the context of a human-robot rescue team, all aspects will be briefly discussed.

Selection of Attributes There are several types of attributes measured from wearable sensors, namely: *1*) environmental attributes; *2*) acceleration; *3*) location; and *4*) physiological signals. Environmental attributes contain information about the individual's surrounding, such as temperature, humidity, or audio level. In general environmental attributes provide sufficient contextual information, but insufficient information to directly infer activities of an individual. Acceleration information is broadly applied to ambulation activities, and a single tri-axial accelerometer is sufficient to provide high recognition accuracy. Location information provides useful context information through for example GPS. Physiological signals contain information about the individual's internal state, such as heart rate, respiration rate, or skin temperature. While physiological signals could provide sufficient information to infer activities of an individual, the signals are difficult to interpret, and the accuracy is generally low.

In the context of a human-robot rescue team environmental attributes, and acceleration information seem the obvious choice as types of attributes. Environmental attributes are able to provide information about the interaction with a device and acceleration information is able to provide a distinction between a wide variety of activities.

Obtrusiveness Obtrusiveness is the undesirable awareness of activity recognition by the individual. Meaning, the individual is negatively influenced by wires, sensors, particular required actions, or the activity recognition system. In the ideal case the individual should be able to perform their daily activities while being unaware of the activity recognition system.

While humans of a human-robot rescue team are in general equipped with a variety of tools, e.g., a helmet, a walkie-talkie, and a tool belt, it is important to prevent additional straining. Therefore, it is preferable to use unobtrusive sensors.

Data Collection Protocol Data should preferably be collected in a natural setting. Since, it might be difficult to generalize activities from data collected in an experimental setting to activities from data collected in a natural setting.

Due to the small amount of human-robot rescue teams, and the variety and low frequency of the missions it is difficult to gather data in a natural setting. However, gathering data in high-fidelity experimental settings might be sufficient to generalize to natural settings.

Energy Consumption Energy consumption is generally not taken into account in activity recognition systems. While all sensors require energy to operate, and store or send information to other devices, in most cases the energy consumption is low enough for sensors to operate extended periods of time.

Since human-robot rescue teams operate for extended periods of time without being able to change batteries, it is important that the sensors operate for the full duration of the mission. However, for the proposed researched the data is collected in an experimental setting making energy consumption less likely to be an issue.

Processing Processing is considered with regard to device which will perform the activity recognition. In most cases the data is send to a central server on which the data is processed, but in some cases it is possible to perform data processing on the device which performs the data gathering.

The activity recognition system runs within the TRADR system and uses information from various sources. Therefore, it is not necessary for the processing to take place on the data gathering device.

Flexibility Flexibility is related to whether an activity recognition model is tailored towards a particular individual, or whether an activity recognition model is suitable for different individuals. Subsequently multiple types of analysis exist to evaluate an activity recognition model, namely subject-dependent, or subject-independent evaluations. In subject-depended evaluations the activity recognition model is trained and tested for each individual, while in subject-independent evaluations a single activity recognition model is trained and tested using data of multiple individuals.

Gathering data for a human-robot rescue team is difficult, due to the small amount of teams, and the low frequency of missions. Furthermore, it is undesirable if the system has to be trained for each individual human of a human-robot rescue team. Therefore, it is preferred to build a model suitable for activity recognition of different individuals.

2.4. Summary

In the previous sections an overview is given of various topics related to this thesis. An overview of human-robot interaction in the context of urban search and rescue, and the relation with the TRADR project is presented. Also, multiple approaches to human activity recognition, and various design principles regarding the development of an activity recognition system are covered. However, it may still not be clear how all topics are related to one another in this thesis, and what aspects of the topics are relevant for this thesis. Therefore, a summary is presented in order to provide a brief complete overview of the background for the remainder of this thesis.

Section 2.1 Urban Search and Rescue presents the characteristics and challenges of human-robot interaction in the context of urban search and rescue. It provides the basis to relate all other topics of the background to.

Section 2.2 Activity Recognition presents an overview of various approaches to activity recognition based on the categorization by L. Chen, Nugent, and Wang (2012). The decision was made to employ a hybrid approach to human activity recognition, based on the characteristics of a human-robot team in the context of urban search and rescue. The framework by Riboni and Bettini (2010) provided the inspiration for the hybrid approach to human activity recognition in a human-robot rescue team. In both cases data-driven and knowledge-driven approaches to human activity recognition are combined into a single framework. In the case of Riboni and Bettini (2010) activities recognized via data-driven approaches are refined with contextual information applied to the domain of daily living. The contextual information consists of the symbolic location of the human (e.g., kitchen, living room, etc.). In the approach detailed in this thesis the knowledge-driven approach forms the basis of the activity recognition system, in which data-driven approaches act as sources of information. Also, while spatial information contains valuable information, the information is difficult to provide in an urban search and rescue context, due to limitations of positioning systems (e.g., the accuracy of GPS is in the range of meters, and does not work well indoors). Having established a basic framework it is important to specify the requirements of an activity recognition system. These include not only a model of human activities in a human-robot rescue team used by the knowledge-driven aspect, but also the types of behaviour relevant to a human-robot rescue team. Also, when designing an activity recognition system various design principles have to be taken into consideration, which support the design and development of the activity recognition system.

For the knowledge-driven aspect of the activity recognition system three approaches are available, namely mining-based, logic-based, and ontology-based approaches. Based on the framework by Riboni and Bettini (2010), other research available on ontology-based approaches to activity recognition, and the fact other partners in the TRADR project were already developing an ontology, the decision was made to pursue an ontology-based approach for the knowledge-driven aspect. With an ontology-based approach an ontology represents the domain, and reasoning makes it possible to infer additional knowledge. According to L. Chen, Nugent, and Wang (2012) ontology-based approaches have the advantages of no 'cold start' problems, interoperability, reusability, and the ability of combining various data sources explicitly, with the disadvantages of being weak in handling uncertainty, and time. The disadvantages are relieved by employing a logic-based formalism making it feasible handling temporal relations.

Section 2.3 Design Principles describes various design principles the activity recognition system has to adhere to. The sCE-method supports the development of human-centred automation and defines three main segments, namely: Foundation, Specification, and Evaluation. Only two terms of the Specification segment, namely requirements, and claims, are explicitly described in Section 3.3 Specification of Requirements and Claims. The requirements, and claims have been formulated according to the research question. Various terms of the Foundation and Evaluation segment are implicitly described throughout this thesis. The fact they are not made explicit according to the sCE-method is due to the fact that the activity recognition system is primarily self-contained with only a partial integration within the TRADR project. This makes it difficult to employ already gained design knowledge. Lara and Labrador (2013) defined several technological design principles related to the development of a human activity recognition system. While, all individual principles have been briefly discussed the following are important for human activity recognition in the context of urban search and rescue: selection of attributes, obtrusiveness, data collection protocol, and flexibility.

3. Model

The following chapter provides a conceptual overview of the activity recognition system, while the next chapter (Chapter 4 Architecture) discusses the design and implementation details of the activity recognition system.

The activity recognition system (as seen in Figure 3.1) consists of two layers: a *sensing* layer, and an *activity recognition* layer. The sensing layer perceives and transforms the behaviour of humans in a human-robot rescue team into a suitable format for the activity recognition layer. Humans in a human-robot rescue team express a variety of different types of behaviour. Three particular types of behaviour are used by the activity recognition system, namely: physical motion, communication, and interface actions. The motivation for the choice of these behaviours is the relevance and feasibility in urban search and rescue missions, which is explained in more detail in the sensing section. Other types of behaviours are able to provide useful information as well, such as object usage behaviour (e.g., Patterson et al. (2005)), or gaze behaviour (e.g., Courtemanche et al. (2011)). Object usage behaviour describes the usage of particular objects, for example, turning a flash light on or off, using a fire extinguisher, or using breathing equipment. However, this means all relevant objects need additional sensors in order to perceive the interaction usage. Also, while this information could be useful for inferring activities of in-field rescuers, this research focusses on team leaders and robot operators. Gaze behaviour refers to conscious and unconscious eye and head movements, for example, looking at objects, or structures (e.g., walls, or roofs). Due to the fact other partners are already performing research on gaze behaviour, and because gaze behaviour is primarily useful for in-field rescuers, gaze behaviour is outside the scope of this project.

The *activity recognition* layer uses three types of human behaviour perceived by the *sensing* layer to infer activities. Besides human behaviour the activity recognition layer also uses one source of information, namely team composition. Many other sources of information could potentially be exploited, for example: spatial information, guidelines for particular events, structural information, cadastral maps, etc. However, while other sources of information could offer a lot of information, this research tries to infer activities from human behaviour alone. Also, the size of this project limits the amount of sources of information which can be taken into account.

The chapter is organized as followed:

- Section 3.1 Sensing provides an overview of the sensing layer, including the sensing components: physical, communication, and interface actions.
- Section 3.2 Activity Recognition provides an overview of the way activities are inferred from the perceived behaviour.
- Section 3.3 Specification of Requirements and Claims provides a specification of the requirements, and claims of the activity recognition system according to the sCE-method.

3.1. Sensing

The sensing layer perceives and transforms the behaviour of humans in a human-robot rescue team into a suitable format for the activity recognition layer. Three types of behaviour are perceived, namely: physical motion, communication, and interface actions. The sensing layer is composed of different components

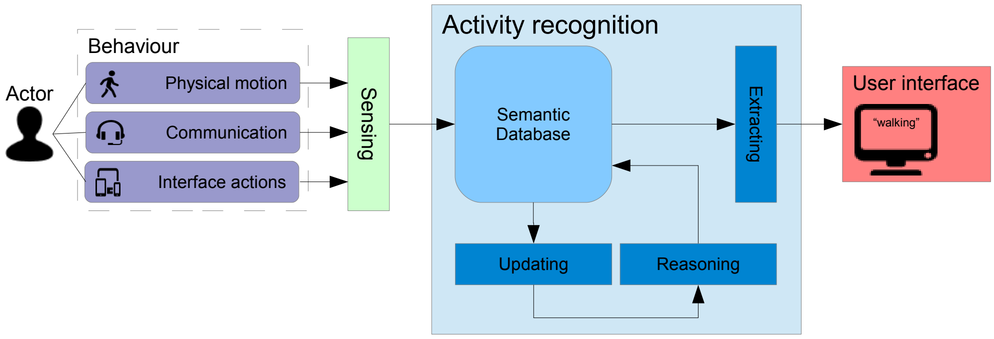


Figure 3.1.: Conceptual overview of the activity recognition system

which are individually responsible for perceiving a particular type of behaviour. Each component uses different sensors to perceive behaviour and generates behaviour specific *events*. All behaviours, the corresponding component in the sensing layer, and the generated events are discussed in more detail in the following sections.

3.1.1. Physical Motion

An important aspect of the activities of rescue workers are physical actions. Rescue workers are constantly moving from one place to another, lifting objects, crouching to enter voids or cross rubble piles (e.g., Burke, Murphy, et al. (2004)). Lara and Labrador (2013) distinguishes different groups of physical actions, including ambulation, fitness, and military. The ambulation group contains actions related to moving from one place to another, such as walking, running, ascending and descending stairs, but also sitting, and standing still. The fitness group contains actions related to fitness exercises, such as lifting weights, spinning, and doing push-ups. The military group contains actions usually performed by military personnel, such as crawling, and kneeling. While a typical rescue worker performs physical actions from all groups, this research emphasizes on activity recognition of humans within the role of team leader and robot operator. This means that only a subset of all physical actions is used, namely: sitting, walking, and standing still. However, even this subset of physical activities provides useful information. For example, when the robot operator is walking for a particular period of time, and then stands still this could indicate the robot operator is taking a break. It is important to note that a human is always performing one physical action at all times. This means it is not possible for a human to perform multiple different physical actions at the same time, or not perform any physical action at any point in time.

The physical sensing component in the sensing layer perceives the physical motion of the rescue workers and stores physical events in the semantic database. Since we are only interested in three physical actions, only three possible physical sensing events can be created, namely: `WalkingPhysicalEvent`, `StandingStillPhysicalEvent`, and `SittingPhysicalEvent`. Each event occurs at a certain instant in time and is performed by a single rescue worker.

3.1.2. Communication

In human-robot rescue teams communication is the exchange of information between agents (both humans and robots) via a variety of channels. Communication encompasses different aspects, such semantic information, context (e.g., who, and when), and representation (e.g., auditory, or visual). Examples of communication include conversations between rescue workers, the transmission of camera images from a robot to a robot operator, or the representation of obstacles on a map. Communication is an important instrument to promote effective collaboration between team members in a human-robot rescue team (Casper and Murphy 2003), but too diverse and complex in order to take all aspects into account. Therefore, we limit ourselves to the conversational interaction between humans. In human-robot rescue teams communication occurs face-to-face, and via two-way radio communication (e.g., walkie-talkies). Face-to-face communication is preferred, since it is easier to use and more reliable in most cases, which is especially the case for communication between in-field rescuers. However, in face-to-face communication it is difficult to automatically extract information. When team members are spatially dispersed (e.g., between in-field rescuers and team leaders) the use of two-way radio communication becomes mandatory. While natural language processing could process the contents of two-way radio communication, the quality suffers due to noisy environments. Therefore, only meta information, i.e. from who, to whom, start time, and end time, is captured from two-way radio communication.

The communication sensing component perceives when rescue workers start and stop talking. The following communication events are generated: `StartedTalkingCommunicationEvent`, and `StoppedTalkingCommunicationEvent`. Both events include at what instant in time they took place, the sending person, and the receiving person.

3.1.3. Interface Actions[1]

Both the physical motion and communication are apparent in current rescue workers activities. The use of software tools to exchange information between rescue workers (TDS), and to control the robots (OCU) is part of the technology of the TRADR project (as described in Section 2.1.2 Project: TRADR). The TDS potentially provides useful information about team, or organizational aspects. However, due to the lack of functionality in the current implementation of the TDS the decision was made to only take the interface actions of the OCU into account. Furthermore, the interface actions are limited to the interactions of the robot operator controlling the UGV with the OCU. This means only interface actions of a UGV robot operator are considered, and none for other robot operators, team leaders, or in-field rescuers.

The interface actions sensing component perceives the following events: `DrivingInterfaceEvent`, `TurningInterfaceEvent`, `MovingFlippersInterfaceEvent`, `MovingCameraInterfaceEvent`, and `ZoomingCameraInterfaceEvent`. Each interface action occurs at a certain instant in time and is performed by a single rescue worker. Even though the OCU contains additional interface actions (e.g., ‘taking a snapshot’), the ones listed above are essential in order to control the robot.

3.2. Activity Recognition

The activity recognition layer transforms the events perceived by the sensing layer into actions, and infers activities of those actions. The activity recognition layer consists of a *semantic database*, and three processes, namely *updating*, *reasoning*, and *extracting*. The semantic database provides persistence by maintaining all current and past actions and activities. The data in the database is structured according to an ontology, which provides a representation of concepts related to activities in human-robot rescue

[1]A better name would have been ‘interface interactions’, to reduce confusion with actions.

teams. Based on the ontology and a rule-like formalism activities are inferred from the data in the semantic database.

In order to illustrate the activity recognition process an example is provided. The sensing layer perceives physical, and interface actions from a robot operator. The robot operator is sitting, and actively using the camera of the robot, while the robot itself remains stationary. From the perceived behaviour the activity recognition system infers that the robot operator is searching for something.

3.2.1. Semantic Database

The semantic database provides storage and reasoning functionality in the activity recognition system. Since all past and current activities are stored in the database, persistence during and across missions is provided. The data is structured according to an ontology, which provides a representation of concepts and relationships related to activities of humans in a human-robot rescue team. The ontology used by the activity recognition system consists of information about the structure of a human-robot rescue team, and three other parts related to events, actions, and activities. The temporal information is provided by a temporal ontology (O'Connor and Das 2011), which allows to represent temporal information in a concise manner. The complete ontology used by TRADR encompasses many more concepts, related to spatial information, and task-based information.

In Figure 3.2 an overview is provided of three parts of the ontology. The different entities are categorized by color, in which red entities indicate a type, green entities indicate a class, and blue entities indicate an instance. An solid arrow indicates a relation between two entities, and an dashed arrow indicates a optional relationship. The *Event* ontology (see Figure 3.2a) represents events based on perceived behaviour by the sensing layer. Each event is performed by one or two humans at a particular instant in time. The *Action* ontology (see Figure 3.2b) represents a group of events, performed by one or two humans at a particular period in time. The *Activity* ontology (see Figure 3.2c) represents a combination of actions, performed by one or two humans at a particular period in time. The temporal ontology allows to represent instants, and periods directly in an ontology using a number of classes and properties (as described by O'Connor and Das (2011)). However, due to the inability of the semantic database to reason directly with the temporal information, and the increased representational complexity, another approach was used. All necessary temporal information is represented using three properties, namely: `temporal:hasTime`, `temporal:hasStartTime`, and `temporal:hasFinishTime`.

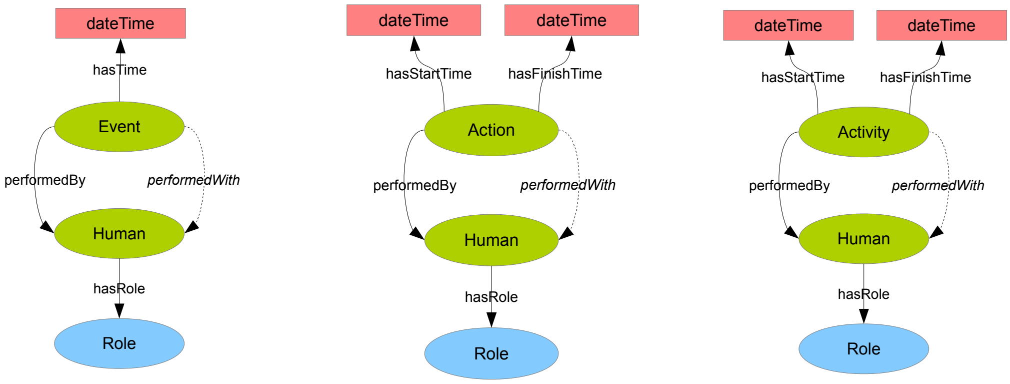
3.2.2. Processes

The activity recognition layer consists of three processes, namely *updating*, *reasoning*, and *extracting*, described in the following sections.

Updating

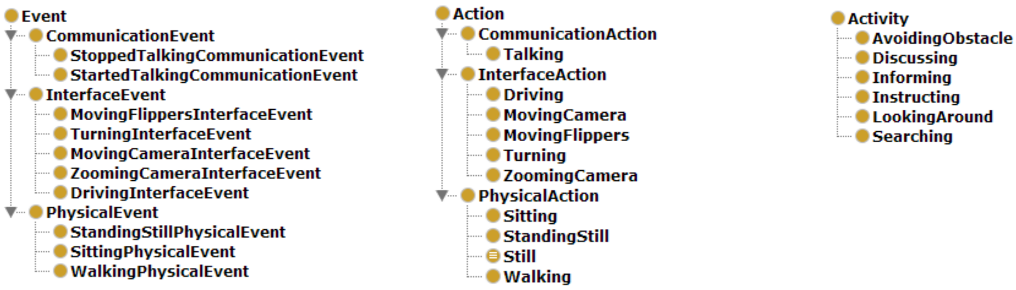
The updating[2] process is responsible for transforming events into actions. Events occur at an instant in time which is problematic for the reasoning process, since the reasoning process infers activities based on periods in time. While some actions inherently occur at an instant in time (e.g., *TakenSnapshot*[3]), the

[2] The name 'updating' refers to the previous implementation of the process prior to the integration within the TRADR project. In the previous process the sensing layer stored actions directly in the semantic database which meant that the finish time had to be updated with the current to provide real-time activity recognition. Also, updating referred to updating the ontology with the latest information as well. After the integration it became apparent an event-like sensing system seemed more sensible and practical. Due to the current implementation the finish time is no longer updated explicitly, but implicitly in the way the updating process handles events.



(a) Event ontology (b) Action ontology (c) Activity ontology

Figure 3.2.: Event, Action, and Activity ontologies



(a) Event class hierarchy (b) Action class hierarchy (c) Activity class hierarchy

Figure 3.3.: Class hierarchies for the Event, Action, and Action ontology

current reasoning process only considers actions with a period in time. The basic idea of updating is to group similar events together, but the actual grouping depends on the type of events based on the individual sensing components. Therefore, the grouping will be discussed based on the type of events.

Physical The grouping of physical events is based on the type, the person who performs the event, and the temporal adjacency to other physical events. The type and the person who performs the event have to be identical, and the temporal window in which two events occur is 5 seconds. The 5 second temporal window is used, because with a frequency of ∼1Hz for physical events it compensates for some latency issues. Figure 3.4 shows an example of how the grouping of physical events into physical actions occur for a single human. The grouping process for physical events has several unresolved issues:

- **Faulty events**: It is unclear how to handle faulty events. For example, if the stream of physical events contains multiple *sitting* events, then a single *walking* event, and then multiple *sitting* events

[3]The *TakenSnapshot* event refers to the process in which a robot operator uses the camera of the robot to take a snapshot, which can then be shared with other team members within the TRADR system. In the current activity recognition system the event is not handled, but shown here as an example of a type of event which is unnecessary to transform into an action.

again. It is likely the *walking* event was misinterpreted by the physical sensing component. A possible solution is to collect multiple physical events and take the most occurring physical event (a similar approach is adopted in Riboni and Bettini (2010)).

- **No events**: It is possible that for a certain period of time (at least not within 5 seconds) no physical events are available (e.g., due to latency issues). This would lead to a gap between physical events, which is per definition not possible, since a human is always performing some kind of physical action[4].

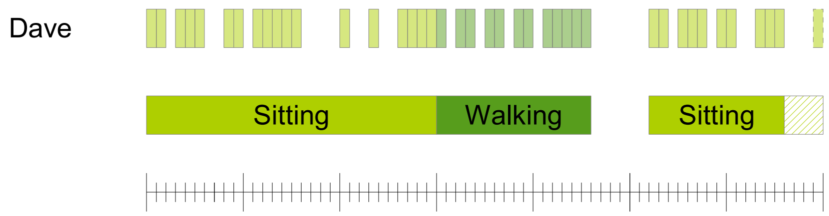


Figure 3.4.: Example of the updating process for physical events

Communication The grouping of communication events is different from the grouping of other events. This is due to the fact only two communication events are possible, namely `StartedTalking-CommunicationEvent` (i.e., *started talking*), and `StoppedTalkingCommunicationEvent` (i.e., *stopped talking*). Since the activity recognition process performs reasoning using actions in real-time, it is necessary for actions to be available as soon as possible. However, if communication events are only grouped if both a *started talking* and *stopped talking* have been received, then there is a gap during which no reasoning using communication actions can be performed. Therefore, whenever a *started talking* event is received a communication action is created of which the period is updated each time the communication updating component is processed. This happens until the *stopped talking* event is received.

Interface The grouping interface events is similar to the grouping of physical events. Interface events of the same type, and performer, which are temporally related should be grouped together into an single interface action. However, the temporal window in which interface events are grouped together might be different. For example, when using an interface an agent might perform shorter bursts of interface actions periodically. This makes it difficult to determine the correct temporal window beforehand. Also, an additional difficulty is the fact that different types of interface events occur simultaneously.

Reasoning

The reasoning process is responsible for inferring activities. Before the reasoning process starts, the semantic database is updated by the updating process to ensure the semantic database contains the latest

[4] Of course a human could perform an unknown physical action, which would be detected as a different physical action, but still stored in the semantic database.

Type	Actions	Description
PhysicalAction	Sitting	
	Walking	
	StandingStill	
CommunicationAction	Talking	
InterfaceAction	Driving	Driving the robot in any direction (i.e., left, right, forward, and backward)
	Turning	Turning the robot in place (i.e., clockwise, and counter clockwise)
	MovingFlippers	Moving the flippers of the robot (i.e., up and down)
	ZoomingCamera	Zooming the camera mounted on top of the robot
	MovingCamera	Moving the camera mounted on top of the robot (using pan and tilt)

Table 3.1.: Overview of all actions categorized by type

information. At the moment the reasoning process infers activities using only actions, but the reasoning process could of course use any type of information from the semantic database. Inferring is the process of deriving conclusions based on premises. The premises are based on a rule-like formalism. The rule-like formalism describes the mapping from actions to activities. For each type of activity one or multiple rules are defined. The inferred activities are stored in the semantic database. It is possible that multiple rules fire simultaneously, of which all results are stored in the semantic database.

The mapping from actions to activities is based on temporal relations between actions, temporal information about actions, and role-based information. The temporal relations are based on Allen's interval-based operators (Allen 1983). With the temporal operators it is possible to describe dependencies between instants and periods. While a multitude of temporal operators are defined, only the following are used: *temporal:before*, and *temporal:during*. The temporal information describes the duration of actions, namely: *temporal:duration*, *temporal:durationGreaterThan*, and *temporal:durationLessThan*. With the temporal information it is possible to ensure an action takes at least or at most a fixed amount of time. Also, it is possible to ensure that one actions takes shorter or longer than another action. The role-based information describes the particular role an agent has at any time. At the moment three roles are defined: *TeamLeader*, *RobotOperator*, and *InFieldRescuer*. Using the role-based information it is possible to ensure that agents of particular types are interacting with each other.

In the following paragraphs an overview is given of all activities and the related rules currently being fired by the activity recognition process. Each activity is defined conceptually, while each rule provides a specific realization of the activity. All rules are represented using a diagram in which all actions are coloured by type: green for physical actions, red for communication actions, and blue for interface actions. If the role of a human is displayed as well, then it is used by the rule as role information. Temporal relations between actions are displayed in a dashed block connecting two actions. Also, temporal information applicable to an action are displayed in brackets in the action. While at Section 3.1 Sensing all actions have already been mentioned, Table 3.1 provides a complete overview of all actions categorized by type.

Activities The activities recognized by the reasoning process provided a balanced set between not only the various behaviours, but also between individual, and team activities.

The activities *instructing*, *informing*, and *discussing* relate to the hierarchical structure commonly found in fire departments. Within the structure a command hierarchy is established in which humans of higher rank provide instructions to other humans lower in rank. In response humans of lower rank inform other humans higher up in the rank. This mechanism ensures a strict information flow ensuring everyone has the necessary information to perform their tasks (for a complete overview of different command hierarchies during emergency management see Waugh and Streib (2006)). The hierarchical structure of the envisioned human-robot team is flat, and the activities may appear somewhat arbitrary. However, in the long-term a human-robot team becomes part of a much larger hierarchy in which information flow becomes more apparent.

The activities *avoiding obstacle*, *looking around*, and *searching* belong to the basic set of activities of a robot operator of an UGV in order to perform their tasks. All these activities are on an individual level in which physical, and interface actions are combined using temporal relations.

Instructing The activity *instructing*[5] is defined as the exchange of information by a human within the role of team leader and another human. The activity is realized by the rule as seen in Figure 3.5, when a human in the role of team leader talks to a human in the role of robot operator or in-field rescuer, and that human responds briefly within a given time.

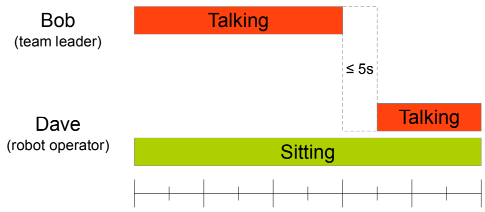


Figure 3.5.: Activity instructing

Informing The activity *informing* is the exchange of information by a human with another human in the role of team leader. The activity is realized by the rule as seen in Figure 3.6, when a human in the role of robot operator or in-field rescuer talks to a human in the role team leader, and that human responds briefly within a given time.

[5]The activity *instructing* is actually realized by two rules. One rule relates to the case in which a human within the role of team leader talks (i.e., the request) to another human who talks back (i.e., the response), as seen in Figure 3.5. The other rule only has a request. Many rules were based on observations during the T-JEx 2014 in Italy of which the end-users consisted of professionals fire fighters from the Prato fire brigade. The fire fighters employed strict communication protocols which demonstrate a clear request-response style of communication. However, based on observations during the T-JEx 2015 in Dortmund, it became apparent that the fire fighters from the Dortmund fire brigade were more flexible in the communication protocols. This meant that in most cases no response was provided by either the team leader, or the robot operator. Therefore, the rule with only a request was applied to provide a more accurate recognition of activities. The reason behind the difference in communication protocols is unclear. It might be culturally related, or due to the different set-up in scenario. A similar case holds for the activity *informing*.

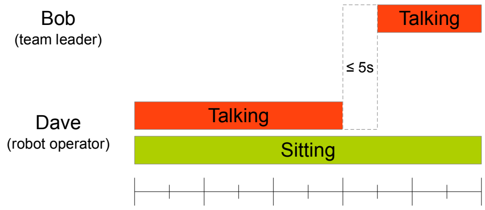


Figure 3.6.: Activity informing

Discussing The activity *discussing* is defined as two humans who are talking back and forth. The activity is realized by the rule as seen in Figure 3.7, when two humans are alternatively talking to each other with a with a brief period in between.

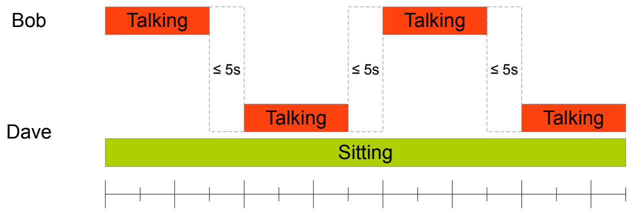


Figure 3.7.: Activity discussing

Avoiding Obstacle The activity *avoiding obstacle* is defined as a robot operator driving a robot around or over an obstacle. The activity is realized by two rules as seen in Figure 3.8. All rules are a combination of the interface actions *driving*, and *moving flippers*, occurring in a particular order with temporal constraints, or simultaneously.

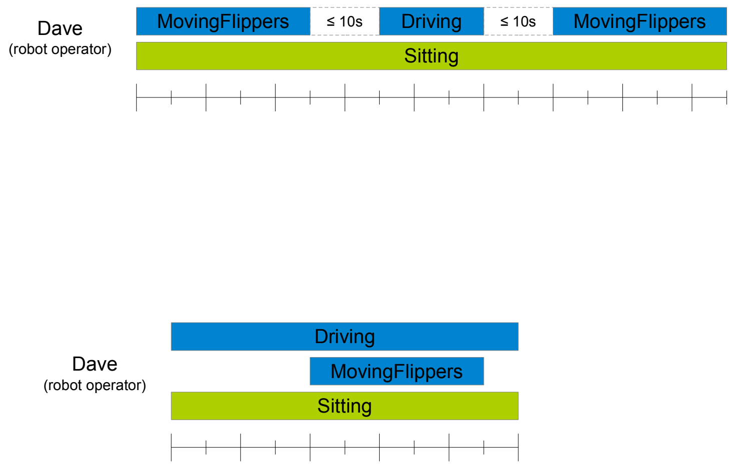


Figure 3.8.: Activity avoiding obstacle

Looking Around The activity *looking around* is defined as a robot operator who uses the robot to look around the location. A robot operator spends a large amount of time building situation awareness. This means the robot operator is busy finding out what is going on around the robot. Examples of difficulties for the robot operator include finding out where the robot is in relation to the environment, and determining whether obstacles are passable. Reasons for spending a large amount of time on figuring out what is going on around the robot include: a different point of view (e.g., closer to the ground, difficult to estimate distances), and obscured vision (e.g., bad lighting).

The activity is realized by three rules as seen in Figure 3.9. All rules are a combination of the interface actions *driving*, and *turning*, occurring in a particular order with temporal constraints.

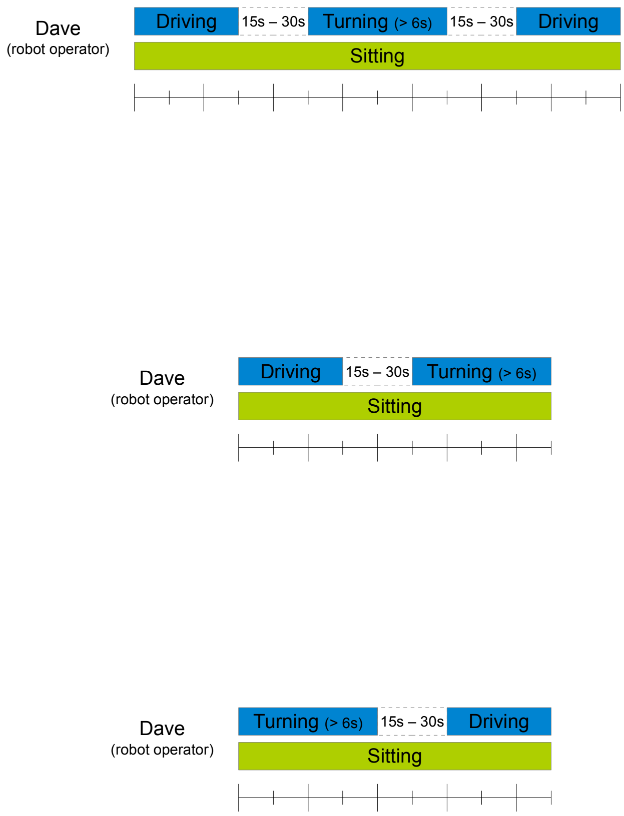


Figure 3.9.: Activity looking around

Searching The activity *searching* is defined as a robot operator who uses the camera of the robot to look for a specific point of interest (e.g., object, or void). The activity is realized by three rules a seen in Figure 3.10. All rules are a combination of the interface actions *moving camera*, and *zooming camera*, occurring in a particular order with temporal constraints.

Figure 3.10.: Activity searching

Extracting

The extracting process extracts information about the perceived actions and recognized activities such that it can be displayed on an interface. A collection containing all actions and activities for all humans is provided by the extracting process. An example of an interface is provided in Chapter 5 Evaluation Figure 5.2b.

3.3. Specification of Requirements and Claims

Based on the situated Cognitive Engineering (sCE) method as described in Section 2.3.2 Situated Cognitive Engineering Method various requirements, and claims for the activity recognition system have been defined. The requirements describe the proposed functionality the system provides to the user, and the claims describe the relationship between the requirements and the hypotheses of the evaluations. Since the activity recognition system is engaged in explorative research the requirements, and claims only indicate expected results, and are not backed by hypothesis. Therefore, no formal criteria are provided in order to evaluate the requirements, and claims.

3.3.1. Requirements

Technical Requirement 1 (TR1) The activity recognition system should monitor behaviour (i.e., physical, communication, and interface actions) of human team members and infer individual and team activities.

Functional Requirement 1 (FR1) The team leader should be able to view activities of all team members.

Functional Requirement 2 (FR2) The activity recognition system should monitor the behaviour in a non-obtrusive manner.

3.3.2. Claims

Claim 1 (CL1) By showing the activities of all team members to the team leader, the awareness of the team leader will increase.

	Claim	Description
Pros	+ situation awareness	Since the team leader is better aware of what all team members are doing, the situation awareness of the team leader increases.
Cons	+ cognitive task load	Since the team leader has to process additional information, the cognitive load of the team leader increases.
	- privacy	Since the team leader is able to view activities of all team members, the privacy of the team members is reduced.

Claim 2 (CL2) The efficiency of the cooperation between team members is increased, because the team leader is more aware of the activities of the team members.

	Claim	Description
Pros	+ efficiency	Since the team leader is better aware of what all team members are doing, the efficiency of the cooperate is increased, due to the team leader being able to provide better instructions.
	- amount of communication	Since the team leader is better aware of what all team members are doing, the amount of communication necessary for cooperation is reduced.
Cons	- trust in team members	Since the team leader is better aware of what the team members are doing, less trust has to be placed in the team members which could negatively influence the cooperation within the team.
	- privacy	Since the team leader is better aware of what the team members are doing, the privacy of the team members is reduced.

4. Architecture

The following chapter provides an overview of the architecture of the activity recognition system. Most design and technical considerations regarding the activity recognition system will be thoroughly discussed. Since the layers and components in the architecture are identical to those in the model, a similar structure is used to discuss the parts:

- Section 4.1 Sensing provides a description of the methods used to perceive human behaviour.
- Section 4.2 Activity Recognition provides an overview of the implementation of the activity recognition.
- Section 4.3 TRADR Architecture provides an overview of how the activity recognition system is integrated within the TRADR architecture.

4.1. Sensing

The sensing layer perceives human behaviour and transforms the behaviour into events which are stored in the semantic database. Even though the intention was to integrate all sensing components, only the physical motion sensing component has been fully integrated within the activity recognition system. The communication and interface sensing components depended on information from the TRADR system, which was not readily available. Therefore, due to time constraints, no integration of the communication and interface sensing components between the activity recognition system and the TRADR system was made. In order to completely understand the technical aspects and design considerations of the sensing components, it is useful to understand the architecture of the TARDR system as discussed in Section 4.3 TRADR Architecture.

4.1.1. Physical Motion

In Section 2.2 Activity Recognition several articles related to recognizing physical activities (e.g. Bao and Intille (2004); Ravi et al. (2009)) are discussed. Many articles describe similar sensors (e.g., accelerometers, and gyroscopes) and approaches to get from sensor data to recognizing activities. Based on a variety of requirements, such as the real-time aspect, accessible hardware, exchange of data, technical skills, we opted for the use of smart phones. Since the implementation and integration of the physical sensing component was only a small part of the project many design decisions were based on the methods discussed in Shoaib et al. (2015).

To perceive physical motion a single tri-axial accelerometer from a Samsung Galaxy SIII (Samsung 2012) smart phone was used. The smart phone was attached in an upright position, with the screen facing away from the human, attached to the pelvic area of a human. An Android application running on the smart phone acquires data from the accelerometer at a frequency of $\sim$20Hz. As shown in Figure 4.1 the smart phone publishes messages with the accelerometer data on the topic `/accelerometer/agent` in which *agent* refers to the particular human carrying the smart phone. The node *physical_modeller* (written in Python) collects a fixed number of messages of which six features are extracted, namely the mean and standard deviation for each axis (i.e., x, y, and z). A linear support vector machine classifies the features into one of three classes, namely: *sitting*, *walking*, and *standing_still*, which correspond with the events *SittingPhysicalEvent*, *WalkingPhysicalEvent*, and *StandingStillPhysicalEvent* respectively. The fixed number of messages is 20 which corresponds with ∼1 second of data. The number was based on the literature discussed in Shoaib et al. (2015) and on a limited number of tests balancing the accuracy of the classifier and the real-time constraint. Events are stored using the SPARQLWrapper (Herman, Fernández, and Tejo 2015) library for accessing the semantic database via the HTTP interface.

For example, if the operator of the robot UGV1 is currently sitting, the following triplets are stored in the semantic database (disregarding the unique identifier of the individual *SittingPhysicalEvent*):
⟨ `SittingPhysicalEvent :performedBy UGV1` ⟩
⟨ `SittingPhysicalEvent temporal:hasTime "2015-02-04T14:12:53"` ⟩

4.1.2. Communication

As discussed in Section 3.1.2 communication is the exchange of information between two rescue workers via two-way radio. However, within the TRADR project communication occurs via the voice over IP (VoIP) client-server application Mumble (Natvig 2015). The application allows for an easier integration of communication and related functionality (e.g., natural language processing) within the TRADR system. In order for the sensing component to be aware of communication between rescue worker, the component needs to be coupled with the Mumble server via any of the application programming interfaces.

Since the communication sensing process is not integrated within the TRADR system, only a small example is provided. Considering the following situation in which the team leader talks with the operator of robot UGV1 for a period of 6 seconds. Initially the following triplets are stored in the semantic database (again disregarding the unique identifier):
⟨ `StartedTalkingEvent :performedBy TL` ⟩
⟨ `StartedTalkingEvent :performedWith UGV1` ⟩
⟨ `StartedTalkingEvent temporal:hasTime "2015-02-04T14:12:31"` ⟩
And 6 seconds later the following triplets are stored in the semantic database:
⟨ `StoppedTalkingEvent :performedBy TL` ⟩
⟨ `StoppedTalkingEvent :performedWith UGV1` ⟩
⟨ `StartedTalkingEvent temporal:hasTime "2015-02-04T14:12:37"` ⟩

4.1.3. Interface Actions

Similarly as with the communication sensing component, the interface actions sensing component is also not integrated within the TRADR system. Therefore, only an overview of the necessary interface events is provided. For example, if the robot operator is driving the robot in any direction, then whenever a driving event is send to the robot, the following triplets are stored in the semantic database (again disregarding the unique identifier):
⟨ `DrivingInterfaceEvent :performedBy UGV1` ⟩
⟨ `DrivingInterfaceEvent temporal:hasTime "2015-02-04T14:13:21"` ⟩
In the current situation a robot operator controls only a single robot, but it is possible that in the future a single robot operator controls multiple robots. This means additional properties have to be added.

4.2. Activity Recognition

In the following an overview of the activity recognition layer regarding implementation details and design decisions is provided. The activity recognition layer consists of the same components as discussed in Section 3.2 Activity Recognition, and therefore the components are discussed in same order.

4.2.1. Semantic Database

In the activity recognition system the semantic database provides storage and reasoning functionality. While several good candidates as a semantic database exist (e.g., AllegroGraph (Franz Inc. 2015), and Virtuoso Universal Server (Semantic Web Company 2015)), based on discussions with other partners of the TRADR project, the semantic database Stardog (Clark & Parsia 2014) was selected. Stardog provides suitable licenses, reasoning support for OWL2 and user-defined rules, and is extensible in various ways. While Stardog supports access via many different protocols (e.g., HTTP, or SNARL) including a built-in web interface, programmatic access via the Java API was preferred, due to the stability and additional functionality. In Stardog databases are composed of two different parts: the schema (i.e. Terminological Box (TBox)), and the data (i.e. Assertional Box (ABox)). Using the ontology editor Protégé Desktop 5.0 beta (Stanford Center for Biomedical Informatics Research 2014) an ontology (based on the ontologies in Figure 3.2) was created and imported into Stardog. While an ontology is able to contain both a schema and data part, the decision was made to keep only the schema part and have all the data inserted programmatically into the semantic database. This means the ontology only consists of classes, and object and datatype properties. The storage and retrieval of triplets was performed using queries written in SPARQL.

4.2.2. Processes

The activity recognition layer consists of three processes, namely *updating*, *reasoning*, and *extracting*. The updating and reasoning process are coupled and are run in sequence every 10 seconds. The extracting process is run dependently from the other processes every 30 seconds.

Updating

Since the updating process is specific for each sensing component, and only the physical sensing component has been fully integrated, the following section only discusses the physical updating component. However, many ideas from the physical updating component can be easily translated into other updating components.

Physical Updating Component The current implementation of the physical updating component is straightforward and not optimized for computational complexity [1]. Nonetheless, the physical updating component runs in a real-time manner.

The physical updating component algorithm is shown in pseudo code in Algorithm 1. The algorithm selects all physical events for all agents ordered by the performer, and time. All physical events are processed and grouped together based on the type, performer, and whether the current group and a new event are temporally related (as shown on line 14 and 15). When this is not the case, one of the following things has occurred:

- **Different type**: For example *action.type = PhysicalWalkingEvent* and *type = SittingPhysicalEvent*.
- **Different performer**: For example *action.performedBy = Bob* and *performedBy = Dave*.
- **Temporally unrelated**: For example *action.finishTime = 2015-05-21T12:22:40* and *time = 2015-05-21T12:22:47*, yielding a duration of 7 seconds.

Whatever case has occurred, it means a physical action is 'finished' and is added into the semantic database. Adding an action to the semantic event is based on the identifier of the event. For example,

[1] One of the main improvements to the physical updating component is to keep track of the latest physical action per performer, which is sufficient since each performer is only performing a single physical action at any given time. However, this information has to be updated, and persist between multiple instances of the activity recognition system.

assume we have the event *WalkingPhysicalEvent_067e6162*. The identifier[2] *067e6162* is reused for the physical action *WalkingPhysicalAction_067e6162*. Due to the way Stardog handles identical individuals, inserting an identical individual simply overwrites the values.

Algorithm 1 Physical updating algorithm

```
results ← query semantic database for physical events
action ← {}
while result ← results do
    event ← result['event']
    type ← result['type']
    performedBy ← result['performedBy']
    time ← result['time']
    if action = {} then
        ▷ Assign result values to action with startTime/finishTime = time.
    else if action.type = type and action.performedBy = performedBy and
                            duration(action.finishTime, time) ≤ 5s then
        action.finishTime ← time
    else
        ▷ Add action to semantic database based on the identifier of the event.
        ▷ Assign result values to action with startTime/finishTime = time.
    end if
end while
if action ≠ {} then
    ▷ Add action to semantic database based on the identifier of the event.
end if
```

Reasoning

The reasoning process is responsible for inferring activities from actions. The logical formalism as described in Section 3.2.2 Reasoning uses rules to map actions onto activities using temporal relations, temporal information, and role-like information. The rules are defined using SPARQL 1.1 (W3C 2013)[3]. SPARQL is an acronym for SPARQL Protocol and RDF Query Language and is semantic query language for database. The syntax is similar to SQL-like query languages, but adapted for accessing semantic data. All rules are based on the `CONSTRUCT`-statement with which it is possible to create new triplets in a query (a similar approach was used by Meditskos et al. (2013)). However, the triplets have to be explicitly created (known as materialization) in the semantic database, since the triplets are only part of the results of the query. Materialization has several issues of which one is problematic in this case, namely: preventing redundancy. The reasoner has to prevent creating the same activity multiple times. The solution is similar to the one discussed in Paragraph 4.2.2 Physical Updating Component, namely: reusing the unique identifiers of the actions. The exact strategy depends on the rule, since some rules have a lot of actions (in which it is possible to use a combination of identifiers of actions), and others only have few actions (in which only the identifier of a single action is used). And, due to way Stardog handles individuals with identical identifiers no unnecessary activities are created. Stardog provides reasoning capabilities which is used by the activity recognition layer in order to infer activities. However, many of the temporal relations are by default not supported by the reasoner. Stardog does provide an application

[2]For readability the identifier has been shortened, but in the actual implementation a type 4 UUID (Wikipedia 2015) is used.
[3]Initially another approach was opted for, which is described in Appendix B Semantic Web Rule Language (SWRL).

programming interface[4] with which custom functions have been implemented. All implementations of the rules of the reasoning process can be found in Appendix A.

It is important to note that while all diagrams of the activities in Section 3.2.2 Reasoning show a physical action, this has not been implemented in rules. This is due to two reasons, namely: the increased computational complexity hampers performance, and it was ensured that all the physical actions were satisfied. All physical actions of the rules apply to the robot operator who was sitting at all times during the evaluation.

Extracting

In the current implementation the *extracting* process simply extracts all actions and activities per agent (as seen in Chapter 5 Evaluation Figure 5.2b). The process could be more efficient by only retrieving actions and activities which are newer than the last time the process ran, but in the current implementation extracting all is easier (and computationally efficient).

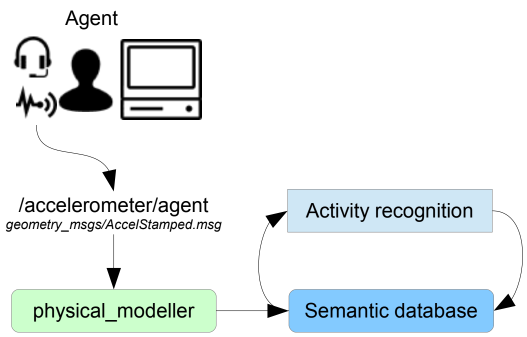


Figure 4.1.: Integration of activity recognition system within the TRADR architecture

4.3. TRADR Architecture

The activity recognition system is integrated within the TRADR system. Due to technical and time constraints, only the physical sensing component is completely integrated. However, the activity recognition system provides the foundation for the communication and interface actions sensing components, and possibly other as well.

The decision behind integrating the activity recognition system within the TRADR system is not only conceptual, but also practical. The TRADR system uses the Robot Operating System (ROS) (Open Source Robotics Foundation 2015) as a framework in the development. ROS not only provides drivers, state-of-the-art algorithms, and developers tools to control a multitude of robots, but also functions as an architecture with which other tools can be integrated. The architecture allows so called nodes (i.e., computational processes) to exchange information through messages via a Blackboard architecture[5]. Figure 4.1 shows an overview of the way the physical sensing component of the sensing layer, and the activity recognition layer

[4] Documentation on how to implement custom function is found at `https://github.com/complexible/stardog-examples/tree/master/examples/function`.

are integrated within the TRADR system. The 'signal' icon denotes messages containing accelerometer data being published for a particular agent to the topic `/accelerometer/agent`. The node *physical_modeller* listens to the topic, collects a fixed number of messages and classifies the accelerometer data into an event of the ontological type *PhysicalEvent* (see Figure 3.3a). The physical event is stored in the semantic database with which the activity recognition process is able to infer activities. In contrast with Figure 3.1 the semantic database and the activity recognition process are separated, since in the actual TRADR system the semantic database is used by a variety of different processes. While there are plans to integrate communication within the TRADR system, at the moment communication takes place outside the system via Mumble. Robot operators are able to control robots via interface interactions with the OCU. This means robot control messages are published via the TRADR system making them in theory available for the activity recognition system. However, the robot control messages would have to be translated into interface actions events which would have been too time consuming. Also, the TDS is currently being completely rewritten from scratch, since it was difficult to integrated the old TDS into the TRADR system. Since the new TDS will be completely integrated within the TRADR system it will be possible to process the interface interactions of the TDS with the activity recognition system.

[5] For a detailed explanation see https://en.wikipedia.org/wiki/Blackboard_system.

5. Evaluation

The following chapter provides an overview of the way the activity recognition system was evaluated:

- Section 5.1 Setting provides an overview of the setting in which the evaluation took place.
- Section 5.2 Activity Recognition System provides an overview of how the activity recognition system specifically was evaluated.

5.1. Setting

The activity recognition system was evaluated during the TRADR Joint Exercise 2015 in Dortmund (hereafter called T-JEx). T-JEx is an opportunity for all partners to meet up for week during which individual systems are integrated into the TRADR system, ideas are discussed, and various evaluations are performed. All the work was performed on site at an old blast furnace in Dortmund (see Figure 5.1 for an impression). The location provides an high-fidelity environment mimicking real urban search and rescue situations as closely as possible. The Fire Department of the city of Dortmund (FDDo) also uses the location to conduct training exercises.

During T-JEx two slots were reserved for a complete system evaluation with actual end-users. At both evaluations the same scenario consisting of two sorties was held in which the end-users were assigned to the roles of team leader, and robot operators. Between the sorties a break was included during which the roles of the end-users were switched. After both sorties a briefing was held between all team members in order to exchange information.

5.1.1. Participants

During both evaluations a total of 6 participants were available. All participants were male, and professional fire fighters of the Fire Department of the city of Dortmund (FDDO).

5.1.2. Materials and Setup

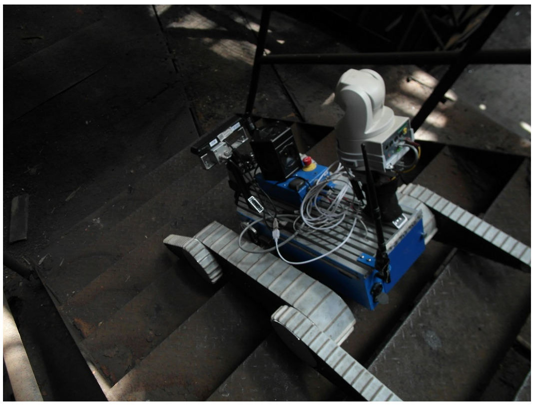
Before the evaluation participants in the role of team leader and robot operator of UGV 1 were asked to place a smart phone (Samsung Galaxy SIII) in the chest pocket of their jacket. Each smart phone ran an application (see Figure 5.3a) to transmit the accelerometer data via the TRADR system to the activity recognition system. Also, each participant was equipped with a head set in order to communicate with other team members. Participants in the role of team leader had access to a tablet containing the TDS. Participants in the role of robot operator of a UGV were placed at a desk in front of multiple laptops containing the OCU and TDS.

5.1.3. Scenario

During the end-user evaluations a location-specific scenario was created conform the present use cases of the TRADR project. The scenario provides a test bed for partners to gather data, and evaluate the system. In the following sections a brief overview is provided of the scenario (see Appendix C Evaluation

(a) Overview of site

(b) Entrance to scenario



(c) Robot descending stairs

(d) Robots at furnace

(e) Robot operator driving robot

(f) Team leader communicating via headset

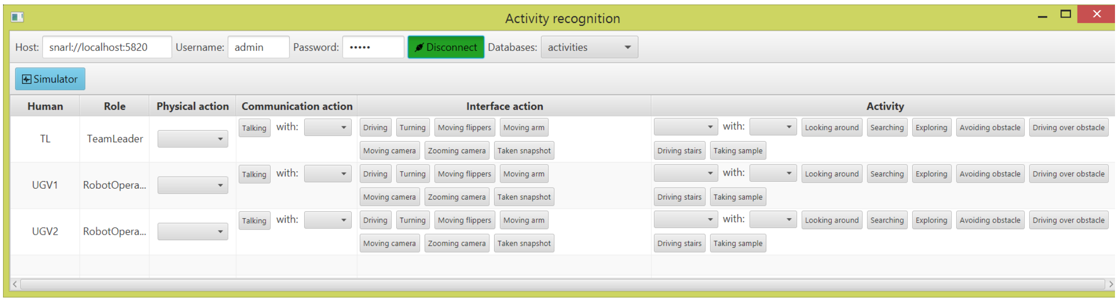
Figure 5.1.: Impression of the location in Dortmund during the TRADR Joint-Exercise 2015

Scenario for a complete overview), but it is important to note that the actual scenario is less important for the evaluation of the activity recognition system. It is preferable that all activities, which are recognized by the activity recognition system, are performed multiple times during the scenario. However, the evaluation of the activity recognition system should not impose any restrictions in the way the end-users behave. End-users should be able to perform the scenario the way they see fit, and the activity recognition systems tries to recognize activities based on their behaviour.

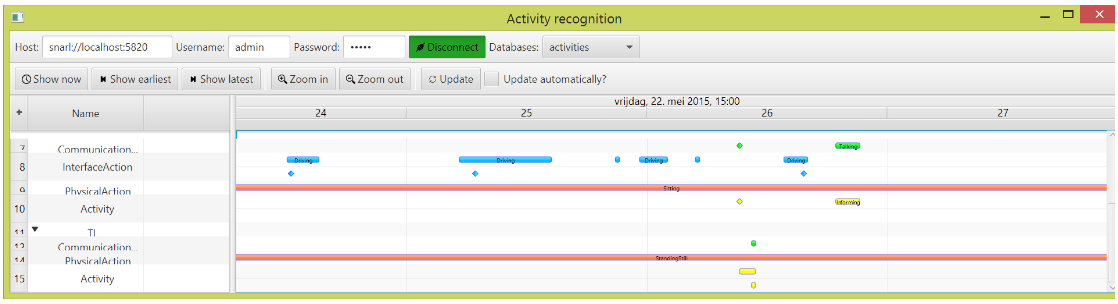
The scenario consists of two sorties. The objectives of the first sortie are to search several locations for victims, and potential hazards (e.g., fire) using the UGV's and UGV's. After the first sortie all victims have been rescued, and the fire is extinguished, but some parts of the building have collapsed obstructing parts of the scenario. The objectives of the second sortie is to perform a building assessment. Before the evaluation the end-users were provided with some background information (e.g., a map, and a background story) regarding the scenario. The background story involved an incident at an industrial complex. Initial responders have assessed the situation, cleared several areas of the complex, and have set up a base camp. Other areas (particularly the area around the furnaces) have not been cleared and the status of the areas is unknown. Those areas that have to be explored for potential victims are hazards (e.g., chemical leaks, danger of collapse). Also, smoke has been seen at the furnace on the left.

5.2. Activity Recognition System

The activity recognition system was evaluated using an observer. The observer was positioned behind both robot operators and was tasked to label the behaviour of the end-users. A list of all activities and descriptions was provided to the observer. The observer only had to label actions and activities of the end-users in the role of team leader (*TL*), and in the role of robot operator for UGV 1 (*UGV1*). This was due to the fact that during a test run it turned out to be difficult to keep track of the other robot operator as well, which resulted in missing actions, and activities. An overview of the software tools which were used to label and monitor the actions and activities is seen in Figure 5.2a.

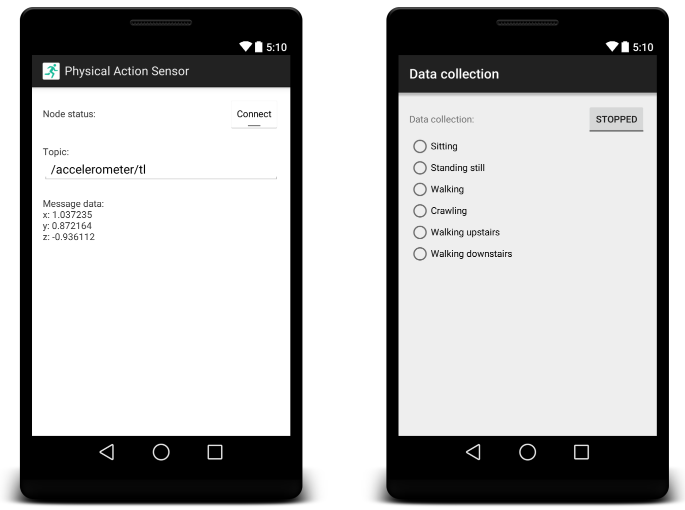


(a) Simulator



(b) Viewer

Figure 5.2.: Simulator and viewer tools



(a) Application for transmitting accelerometer data in real time

(b) Application for gathering training data to build the model

Figure 5.3.: Smart phone applications

6. Results

As described in Chapter 5 Evaluation the activity recognition system was evaluated during the T-JEx 2015 in Dortmund. Since the results of the activity recognition system are self-contained, a comparison is made with two other approaches, namely *most common activity*, and *random activities*. Since all of the results are based on several performance measures a brief overview is presented (for a detailed explanation see Sokolova and Lapalme (2009)). Assuming a binary prediction problem in which you have labelled data (ground truth), and a model predicting classes (predictions) it is possible to define a confusion matrix (see Figure 6.1). Based on the confusion matrix it is possible to define various performance measures:

- **Accuracy** describes the fraction of correct predictions: $\frac{TP+TN}{TP+FP+FN+TN}$.
- **Balanced accuracy** describes the fraction of correct predictions averaged per class: $\frac{0.5*TP}{TP+FN} + \frac{0.5*TN}{TN+FP}$.
- **Precision** describes the fraction of actual positive predictions among all positive predictions: $\frac{TP}{TP+FP}$. Intuitively precision describes the certainty about the correctness of the predictions.
- **Recall** describes the fraction of actual positive predictions among all predictions: $\frac{TP}{TP+FN}$. Intuitively recall the describes the fraction of predictions detected.

		Ground truth	
		True	False
Prediction	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

Table 6.1.: Example confusion matrix

In the following chapter an overview is given of the results of the activity recognition system:

- Section 6.1 Sensing provides the results of the physical sensing component.
- Section 6.2 Activity Recognition provides the results of the activity recognition, and the results of two other approaches.
- Section 6.3 Feedback on Requirements and Claims describes the feedback on the requirements and claims based on an interview with two end-users.

6.1. Sensing

The model of the physical sensing component was evaluated using the observed physical actions for both the team leader and the robot operator of UGV 1. Before the evaluation data was gathered for training the model of the physical sensing component. Data was gathered by three interns using self-annotated physical actions with a smart phone application (see Figure 5.3b).

In Table 6.2 an overview of the results for both of humans. For robot operator of UGV 1 the accuracy is 0.93, with a precision is 1.00, and a recall of 0.93. For the team leader the accuracy is 0.64, with a precision of 0.94, and a recall of 0.62. The overall accuracy is 0.78, with a precision of 0.98, and a recall of 0.78.

		Observed	
		Still	Walking
Recognized	Still	972	0
	Walking	77	0

(a) Robot operator of UGV 1

		Observed	
		Still	Walking
Recognized	Still	552	34
	Walking	341	106

(b) Team leader

Table 6.2.: Confusion matrix for the physical actions

6.2. Activity Recognition

During the evaluation actions and activities were observed. The observed actions serve as input for the activity recognition layer, since not all components of the sensing layer have been integrated. The activity recognition layer produces *recognized* activities, which can be compared with the *observed* activities. The idea behind the comparison is to determine whether the recognized and observed activities match in the temporal domain, and in other properties. The other properties include the human which performs the activity and possibly the human with which the activity is performed. For example, the observer labelled the activity *informing* with the following triplets:

⟨ `Informing :performedBy TL` ⟩
⟨ `Informing :performedWith UGV1` ⟩
⟨ `Informing temporal:hasStartTime "2015-02-04T14:12:53"` ⟩
⟨ `Informing temporal:hasFinishTime "2015-02-04T14:13:07"` ⟩

In order for the recognized activity to match the properties *performedBy* and *performdWith* have to be the same. While in the general case these properties are important since activities have to be recognized for all humans; in this case only activities of the team leader and a single robot operator were observed. Therefore, only the temporal domain is being considered in comparing the activities. By viewing the activity recognition as a model classifying whether an activity has or has not occurred for a given time granularity, it is possible to use well-defined performance measures. Example output of the activity recognition model under the assumption of a multi-class classification problem is given in Table 6.3, in which for each activity is shown whether it is recognized or not at each point in time. Also, while multiple activities can occur at the same time, this rarely happens. Therefore, the performance measures are provided per activity.

The results of the activity recognition are discussed per day due to the substantial differences between the evaluations. The evaluation on day 1 went according to the scenario, in which both sorties were performed. However, during sortie 2 the network was not disrupted intentionally (as should have been according to the requirements). On day 2 only sortie 1 was performed with various intentional network interruptions based on the requirements. The interruptions had no effect on the data, because the activity recognition system ran on a separate laptop. Also, it turned out that on day 2 roughly 12 minutes of all the data (i.e., all actions and all activities for all humans) was missing due to an unknown reason. This means only data from the beginning and the end of the sortie on day 2 are available.

In Table 6.4 the results of various evaluation measures are provided for day 1 and day 2. Figure 6.1 provides a graphical representation of the observed and recognized activities, and the overlap between those as a percentage of the total time.

AvoidingObstacle	0	0	0	0	0	1	...	1	1	0	0	0	0	0
Discussing	0	0	0	0	0	0	...	0	0	0	0	0	0	0
Informing	0	0	0	0	0	0	...	0	1	1	1	1	1	0
Instructing	0	0	1	1	1	1	...	0	0	0	0	0	0	0
LookingAround	0	0	0	0	0	0	...	0	0	0	0	0	0	0
Searching	0	0	0	0	0	0	...	0	0	0	0	0	0	0
	time (granularity in seconds)													

Table 6.3.: Example output of the activity recognition model under the assumption of a multi-class classification problem

Activity	Accuracy	Balanced accuracy	Precision	Recall
AvoidingObstacle	0.98	0.50	-	0.00
Discussing	0.99	-	0.00	-
Informing	0.94	0.79	0.43	0.62
Instructing	0.95	0.58	0.77	0.17
LookingAround	0.86	0.70	0.27	0.53
Searching	1.00	0.50	-	0.00
Average:	0.95	0.61	0.37	0.26

(a) Day 1

Activity	Accuracy	Balanced accuracy	Precision	Recall
AvoidingObstacle	0.99	0.50	0.00	0.00
Discussing	1.00	-	0.00	-
Informing	0.98	0.84	0.59	0.70
Instructing	0.98	0.66	0.36	0.33
LookingAround	0.70	-	0.00	-
Searching	1.00	-	0.00	-
Average:	0.94	0.66	0.15	0.34

(b) Day 2

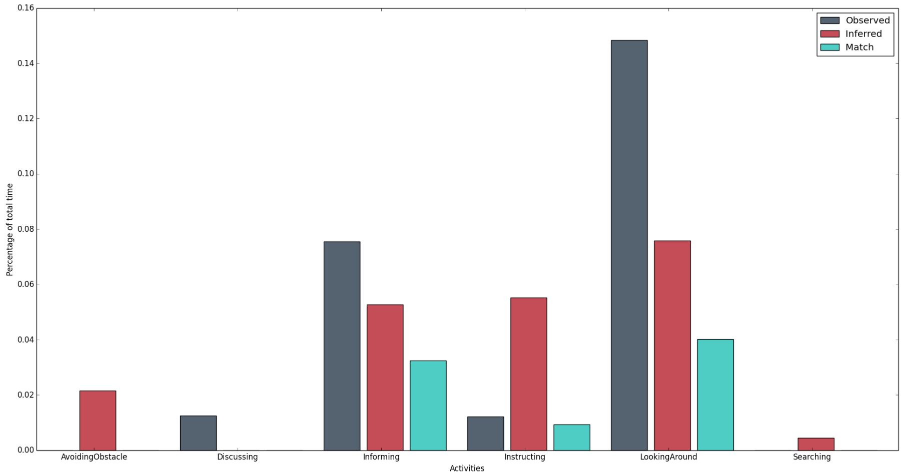
Table 6.4.: Results for each activity for day 1 and day 2, containing the accuracy, balanced accuracy, precision, and recall

6.2.1. Comparison

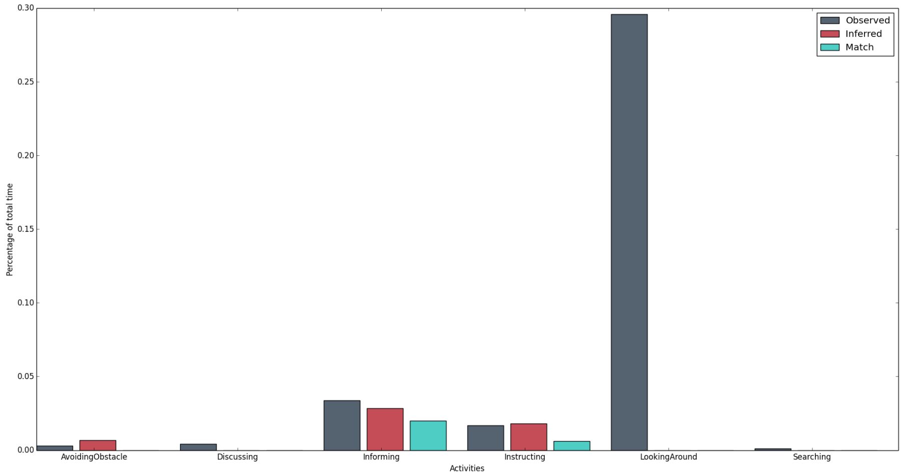
In Section 6.2 Activity Recognition an overview is presented of the results of the activity recognition system. However, the results are self-contained and difficult to generalize. Therefore, in order to put the results into perspective a comparison is made with two other approaches.

Most Common Activity

A basic approach is to always recognize the most common activity. Most common can be either defined as the most frequent activity, or the activity with the longest total duration. For day 1 the most occurring activity is *informing* with 36 occurrences (50.00%), and *looking around* with a total duration of 14.84% of the total time. For day 2 the most occurring activity is *informing* with 23 (51.11%) occurrences, and *looking around* a total duration of 29.56% of the total time. The results for both approaches are presented in Table 6.5.



(a) Activities day 1



(b) Activities day 2

Figure 6.1.: Activities for day 1 and day 2, depicting the percentage observed, inferred, and overlap of the total time per activity

Random Activities

Another approach to compare the results of the activity recognition system with is with random activities. Using a naive approach to random, i.e., simply predicting true or false for each activity at each time point, yields the results as shown in Table 6.6 (averaged over 10 runs).

	Day	Accuracy	Balanced accuracy	Precision	Recall
Most occurring	Day 1	0.81	0.57	0.51	0.15
(*informing*)	Day 2	0.87	0.58	0.23	0.12
Longest total duration	Day 1	0.83	0.59	0.55	0.19
(*looking around*)	Day 2	0.87	0.67	0.32	0.33

Table 6.5.: Results for both approaches as the most common activity

Day	Accuracy	Balanced accuracy	Precision	Recall
Day 1	0.50	0.50	0.50	0.04
Day 2	0.50	0.50	0.53	0.06

Table 6.6.: Results for the random activities approach

6.2.2. Summary

The results (see Figure 6.2) show that the balanced accuracy for the activity recognition system outperforms all but one approach, namely the longest total duration on day 2. Using the performance measures precision and recall, the results show that the precision of the activity recognition system is *lower* in all cases when compared to the other approaches, but the recall of the activity recognition system is *higher* in all cases when compared to the other approaches.

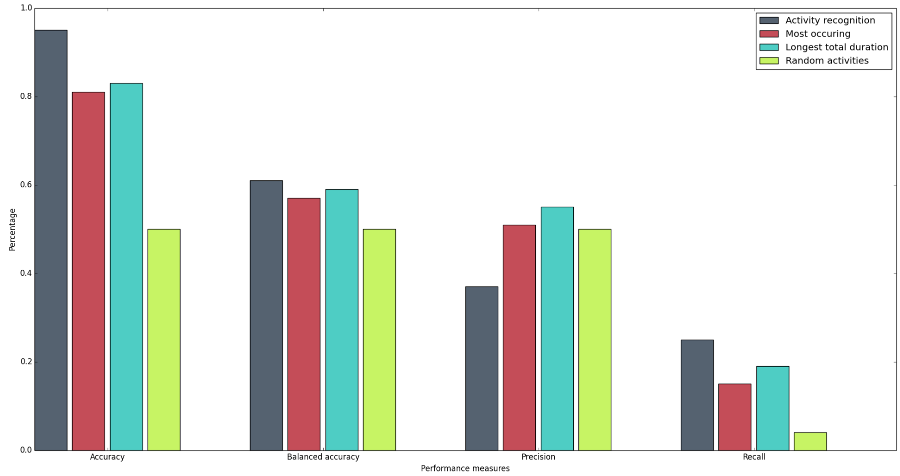
6.3. Feedback on Requirements and Claims

With no formal evaluation of the requirements a subjective evaluation in the form of an interview was conducted with two end-users. During a phone conversation of ~1.5 hours with the end-users several questions related to the requirements were asked. Since no interface (see Figure 5.2b) was present during the evaluation, they were explained of the capabilities of the activity recognition system via an example relating to the scenario. In the following sections a summary of the answers to each of the questions is presented (the full contents of the interview are listed in Appendix D End-Users Interview).

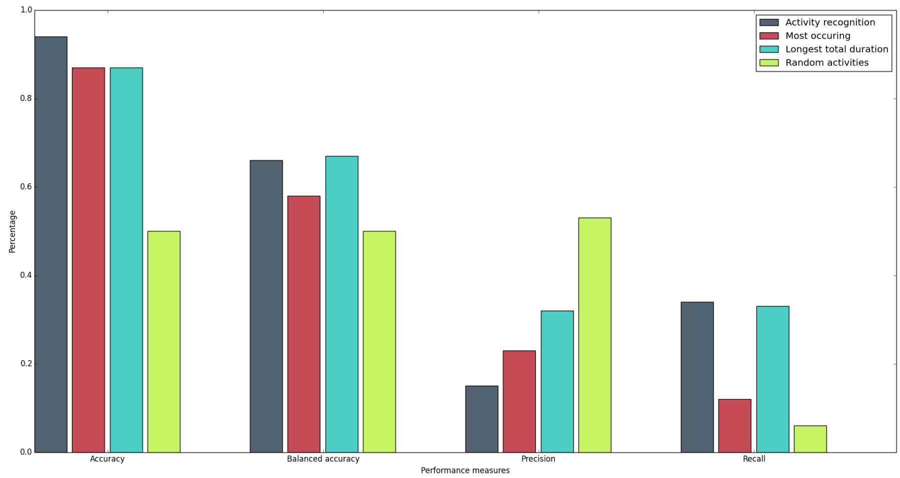
1. As a team leader, do you think the activity recognition system changes your awareness of the team members? In general, as a team leader, they trust that the team members carry out their assigned tasks. If team members are unable to carry out their tasks, or encounter any problems they will report to the team leader. In this sense the activity recognition system has no added value. During the mission it might even distract the team leader from what is going on.

2. Would you trust the activities recognized by the activity recognition system? In principle they do, but it requires insight into how activities are inferred from actions. Also, the actions and activities would need to relate to particular events (e.g., the detection of smoke, or finding a victim) to help understand what is happening. Furthermore, it will take time to trust these kind of technologies, especially for older fire fighters.

3. As any member of the team, how comfortable are you with your activities being monitored? In general, many missions occur in a public place with other persons around who are aware of what you are doing. However, these kind of technologies would monitor and log everything that happens. This might make it easier for others to hold a rescue worker liable when a mistake is made.



(a) Results day 1



(b) Results day 2

Figure 6.2.: Results for day 1 and day 2, containing the accuracy, balanced accuracy, precision, and recall for each approach

4. As a robot operator or an in-field rescuer, how comfortable are you with your activity being monitored by the team leader? In most cases other rescue workers are already closely paying attention to what others are doing, so it would not change a lot.

5. Would you use the output of the activity recognition system in the current state? In the current state the information would be useful for analysis after a mission. It could provide a useful tool to support rescue workers becoming more proficient at their work.

6.3.1. Summary

At the moment the end-users do not think that the activity recognition system has any added value during a mission, since rescue workers trust in the abilities of each other. However, they do think the information is useful for evaluation of the mission afterwards in becoming more proficient in their work.

7. Discussion

In the following chapter a reflection is given on the results, the hybrid approach to activity recognition of humans in a human-robot rescue team is discussed, and a conclusion with further warranted research is provided.

7.1. Evaluation

The activity recognition system was evaluated during the T-JEx 2015 in Dortmund at an old blast furnace, which provided a high-fidelity environment with actual fire fighters. Of the activity recognition system the physical sensing component, and the activity recognition were evaluated, discussed below.

7.1.1. Sensing

Of the sensing layer only the results of the model of the physical sensing component are discussed, because other components were not integrated within the TRADR system. The model had an overall accuracy of 0.78, with an accuracy of 0.94 for the robot operator of UGV 1, and an accuracy of 0.64 for the team leader (as shown in Section 6.1 Sensing). A substantial difference between the results of the robot operator for UGV 1, and the team leader exists. An explanation for this fact lies in the lack of observed data for the team leader. Since the observer was facing away towards the robot operators and away from the team leader, many changes in physical actions of the team leader went unnoticed by the observer.

Similar research shows overall accuracies between 0.73 and 0.99 by Ravi et al. (2009), overall accuracies between 0.78 and 0.90 by Kwapisz, Weiss, and Moore (2011), and overall accuracies between 0.86 and 0.88 by Wyss and Mäder (2010). While the results appear similar it is difficult to provide a good comparison, because various factors influence the results, namely the number of classes, the type of models, the features, the amount of training data, etc. Since I have no intention to provide an in-depth comparison between the results of similar studies, it is better to determine whether the results are sufficient in the context of urban search and rescue. At the moment the model is able to predict two physical actions, namely *walking*, and *still*. For these physical actions the current activity recognition system is sufficient, since the reasoning process only uses rules which depend on those physical actions. However, in the future the model will need to predict additional physical actions to cope with the more varied behaviour (e.g., climbing, crouching, etc.) of in-field rescuers.

Robustness, and reliability are important properties of technology in the context of urban search and rescue. An overall accuracy of 0.78 means that in 1 out of 5 cases the model predicts a wrong physical action, which prevents the reasoning process from inferring activities. Therefore, while the results are comparable to the lower end of the spectrum of the results of similar research, the accuracy of the model is not reliable enough, nor is the model able to provide necessary robustness. Thus, in the future the model of the physical sensing component needs to be improved, not only in performance, but also in the number of physical actions the model predicts.

7.1.2. Activity Recognition

In the following sections the results are discussed per related evaluation measures, namely accuracy and balanced accuracy, and precision and recall. Also, a meta-analysis of the evaluations measures is provided.

Accuracy and Balanced Accuracy as Evaluation Measures

Under the assumption that the activity recognition model is a multi-class classification model various performance measures are computed for each activity as listed in Table 6.4). When considering the average accuracy for day 1 (0.95) and day 2 (0.94) the results look promising. However, since only 4.14% for day 1, and 5.91% for day 2 of the total time an activity is observed the results are biased towards the true negatives (see Table 6.1). This means it is possible for a model to get a high accuracy score if no activities are predicted at all. Therefore, the balanced accuracy provides a better performance measure, because it avoids inflated accuracy scores on datasets by correcting for the imbalanced class. This means that a balanced accuracy of 0.5 is considered chance level. As can be seen from the results the balanced accuracy is 0.61 for day 1 and 0.66 for day 2.

In order to put the results into perspective a comparison is made with two approaches. The most common activity approach always predicts the most common activity, in which 'most common' is defined as the most occurring activity, and as the activity with the longest total duration. The results for both days are presented in Table 6.5. When only considering the average balanced accuracy it is evident that the activity recognition slightly outperforms the most common activity, except for the activity with the longest duration for day 2. This can be explained by the fact that the activity *looking around* was observed 29.56% of the time for day 2. This means that the most common activity approach is correct 4.93% of the total time, while the activity recognition model is only correct 2.60% of the total time. However, this does not explain why the activity recognition model predicted the activity *looking around* 0.00% of the time for day 2. The obvious explanation for this fact lies in the lies in the inability of the activity recognition system to recognize the looking around behaviour from the observed actions. This means that the looking around rules were not suitable for detecting the activity. An additional explanation is the fact that the activities the observer was observing were open for interpretation. While a list of activities and descriptions was provided, the descriptions were not as well-defined as the rules.

Another approach is selecting a random activities at each point in time. From the results presented in Table 6.6 can be seen that the balanced accuracy is 0.50 for both days. This is to be expected due to the imbalanced data set. In both cases the activity recognition model outperforms the random activities approach by 0.61 to 0.50 for day 1, and 0.66 to 0.50 for day 2 respectively. This means the activity recognition model has a higher performance than simply selecting a set of random activities at each point in time.

Precision and Recall as Evaluation Measures

The activity recognition model can also be compared with the other approaches using the performance measures precision, and recall. In Chapter 6 Results a formal definition is provided of precision, and recall, but intuitively precision provides of a measure of how many of the predictions are correct, and recall provides a measure of how many of the predictions are detected. In the case of the activity recognition model the average precision is 0.37, and the average recall is 0.26 for day 1, and the average precision is 0.15, and the average recall is 0.34 for day 2 as seen in Table 6.4. For example, when considering day 1 a positive prediction by the model is likely to be correct in 37% of the time, and the model will detect an activity in 26% of the time. Since there is a trade-off between precision and recall, this raises the question about what is more important: the activities the model recognizes are correct, or that the model detects activities. In the context of urban search and rescue it less important the activity recognition model misses activities, than if the activities predicted by the activity recognition system are incorrect. However, if the activity recognition model is unable to detect activities other systems are unable to provide support. Thus, while precision is favoured over recall, the recall score has to be sufficient in order for activities to be recognized. While it is not possible to determine what 'sufficient' is, since that depends on how the information about activities is used by other systems, the precision and recall scores are unacceptably low. It seems unwise for any support system to base their decisions on the activities recognized by the activity recognition system, maybe with the exception for a few particular activities, such as *informing*, and *looking around*.

I will now compare the results with the most common activity approach, and random activities approach. The most common activity approach consists of the most occurring activity, and the activity with the longest duration. For the most occurring activity the precision is 0.51, and the recall is 0.15 for day 1, and the precision is 0.23, and the recall is 0.12 for day 2. For the activity with the longest duration the precision is 0.55, and the recall is 0.19 for day 1, and the precision is 0.32, and the recall is 0.33 for day 2. In all cases the activity recognition model has a lower precision, and a higher recall when compared to both types of the most common activity approach. This means for both types of the common activity approach if a prediction is made it is more likely to be correct than if the activity recognition model makes a prediction. But, the activity recognition model detects more activities than both types of the most common activity approach.

For the random activities approach the precision is 0.51, and the recall is 0.04 for day 1, and the precision is 0.53, and the recall is 0.06 for day. This means for the random activities approach if a prediction is made it is more likely to be correct than if the activity recognition model makes a prediction. But, the activity recognition model detects more activities than the random activities approach.

Based on the discussion between precision, and recall in the context of urban search and rescue, and the results, the most common activity approach, and the random activities approach outperform the activity recognition model. This means the activity recognition model will need to increase the precision, while still maintaining a sufficient recall score.

Meta-Analysis of Evaluation Measures

The results of the activity recognition model have been evaluated using the performance measures: balanced accuracy, precision and recall. The basis for the evaluation was to assume the activity recognition model to be a multi-class classification model. While this assumption allows the compare the activity recognition model with other approaches, and is used in many papers regarding activity recognition, a more sophisticated evaluation measure is preferred. Also, another reason for using the chosen evaluation measures is the fact that, while a more sophisticated evaluation measure might provide a better representation of the performance of the activity recognition system, at the moment the activity recognition system can still be improved in many ways without the need for a sophisticated evaluation measure. However, there are some aspects which I consider important to be taken into account in the case a more sophisticated evaluation measure is warranted.

Weighted Activities At the moment all activities are considered equal. However, in the context of urban search and rescue communication is an important aspect during missions. The chain of command relies on information, and without communication information is less easily exchanged. So, in the context of urban search and rescue, activities related to communication (i.e., *instructing*, *informing*, and *discussing*) should be weighted more than other activities. However, it also raises the question as to what the weights for each activity would be, which would also be context dependent.

Incorrectly Recognized Activities As discussed in Section 7.1.2 Precision and Recall as Evaluation Measures it generally more important that the recognized activities by the activity recognition model are correct, over the detection of activities. So, while the model should try to reduce the chance of incorrectly recognizing activities, certain errors should be penalized more than others. For example, if the robot operator of an UGV is trying to avoid an obstacle, but the activity recognition system recognizes the activity looking around, and therefore no support is provided in time and the robot crashes. In this case, the activity recognition system incorrectly recognizes the activity avoiding obstacle for the activity looking around, which should be penalized more than if the activity recognition system incorrectly recognizes informing for discussing. Thus, various incorrectly recognized activities should be penalized more than others. However, determining which activities should be penalized, and by how much is context-dependent and non-trivial, which makes this aspects difficult to realize.

7.1.3. Feedback on the Requirements and Claims

The results of the interview with two end-users are provided in Section 6.3 Feedback on Requirements and Claims. The interview was intended to receive feedback on the requirements and claims as defined in Section 3.3 Specification of Requirements and Claims. It is not scientifically responsible to verify the requirements and claims based on the results of the interview, since the requirements and claims were not formally evaluated. Therefore, the discussion is necessarily somewhat open and speculative.

CL1: By showing the activities of all team members to the team leader, the awareness of the team leader will increase

Based on the answers to question 1, 2 and 5, the awareness of the team leader is unlikely to increase. The team leader trusts the team members to carry out their assigned tasks, which means the team leader has no incentive to use the output activity recognition system. However, the evaluations were performed in a small-scale setting in which the team leader was managing only two robot operators. Also, since the pace in these missions is relatively low this often lead to the micro-managing of the robot operators by the team leader (based on personal observations). In the case a team leader has to keep track of multiple team members who all require attention, the information provided by the activity recognition system might provide additional awareness.

CL2: The efficiency of the cooperation of team members is increased, because the team leader is more aware of the activities of the team members

Based on the answers to question 1, 2 and 5, the efficiency of the cooperation between team members is unlikely to increase. The explanation is similar to reasons provided in CL1. While some cooperation between the team members existed during the valuation, the limited number of team members makes it difficult to extensively evaluate cooperation.

7.2. Hybrid Approach to Activity Recognition of Humans in a Human-Robot Rescue Team

In Chapter 2 Background an overview is presented of various approaches to activity recognition based on the work of L. Chen, Nugent, and Wang (2012). The primary distinction is made between data-driven (e.g., generative, discriminative, etc.) and knowledge-driven models (e.g., mining, logic, etc.). An hybrid approach combining data- and knowledge-driven models was applied to a human-robot rescue team. For the data-driven aspect a discriminative approach was applied to transforming physical motion perceived using accelerometer data into events. The knowledge-driven aspects entails an ontology-based for describing events, actions, and activities related to humans in a human-robot rescue team. Based on L. Chen, Nugent, and Wang (2012) ontology-based approaches have several advantages compared to other approaches, namely less ‘cold start’ problems, easier re-usability, access to inference, and the ability of combining multiple data sources explicitly. Also, ontology-based approaches ensure the domain is explicitly modelled, which means all aspects related to activity recognition of humans in a human-robot rescue team has to be specified. Modelling the domain is non-trivial and often requires expert knowledge to specify all the relevant aspects. The disadvantages of ontology-based approaches, namely weak in handling uncertainty and time, are relieved by applying a logic-like formalism to handle temporal relations. However, the logic-like formalism is limited in expressiveness, and verbose. An option to be explored is to employ another logic-like formalism, such as Prolog, which provide a more expressive language.

7.2.1. Behaviours

The hybrid approach uses information perceived form three different behaviours, namely: physical motion, communication, and interface actions. Physical motion is perceived using accelerometers worn by the rescue workers of which physical actions are recognized using an data-driven model. Meta information about communication is perceived of the usage of communicative devices of the rescue workers. Interface actions are perceived based on the interactions between the humans and the technology assisted interfaces, such as the Operator Control Unit, and Tactical Display System. The selected behaviours are the minimal set of behaviours necessarily present in humans of a human-robot rescue team. Most of the activities humans of a human-robot rescue team perform involve a combination of the behaviours.

Other types of behaviour are able to provide additional information, for example: object-usage (e.g., usage of fire extinguishers, or cutting tools), or gaze behaviour. However, object-usage behaviour provide primarily information about in-field rescuers, which were not part of the focus in this research. Also, while gaze behaviour potentially provides information about what a human has or has not ‘seen’, it is still a topic of ongoing research by other partners and therefore outside the scope of this project. Besides other types of behaviours, additional information can also be provided via sensor information. For example, spatial information provides knowledge about the location of the rescue workers, robots, and other assets, and physiological information provides knowledge about the physical and mental well-being of the rescue workers.

An important question aspect of the behaviours has not been mentioned yet. While the selected behaviours are the minimal set of behaviours present in humans of a human-robot rescue team, the question remains whether the selected behaviours are sufficient for recognizing the desired activities. This could namely be part of the explanation as to why the results of the activity recognition system are unacceptably low. Only meta information of the communication behaviour is perceived, which might contain insufficient information to recognize the activity *informing*, *instructing*, and *discussing* with. A possible solution is to perform natural language processing to extract the necessary information. However, this might be problematic in the context of urban search and rescue due to the noisy environment. Also, the activities *avoiding obstacle*, *looking around*, *searching* are unlikely to be recognized using only interface actions. This becomes apparent when looking at the results of these activities; in particular the activity *looking around*. Gaze behaviour might not only provide sufficient behaviour to recognize the activity *looking around*, it might also provide useful information about what the robot operator is looking at. This makes it easier to provide more accurate support to the robot operator.

To summarize, at the moment three behaviours, physical motion, communication, and interface actions, are perceived. The inclusion of other types of behaviours, such as gaze or object-usage behaviour, and information, such as spatial information, might not only be useful to refine the current activities, it also allows to define new activities.

7.2.2. Rules

The knowledge-driven aspect of the hybrid approach allows to represent entities, and relations related to human activities in a human-robot rescue team. Rules are defined according to a logic-like formalism with which a reasoner is able to infer activities based on the specification of the rules. The knowledge-driven aspect is limited in expressiveness, which is due to an inherent balance between computational complexity and language expressiveness (for an overview of SPARQL complexity see Pérez, Arenas, and Gutierrez (2009)). Also, combined with the fact the logic-like formalism is verbose, it is complex process defining rules. One way to relieve the complexity of defining rules, is by defining additional entities and relations in the ontology. At the moment temporal relations only apply to temporal entities, but it is possible to define generic relationships between actions. For example, with the property `adjacentTo` it is possible to define whether two actions are temporally adjacent for a given temporal period. Also, the logic-like formalism does not allow to model uncertainty. By being able to model uncertainty it is possible to more accurately specify the activities. One advantage of using a logic-like formalism that has not been mentioned yet is the ability to make the reasoning process explicitly. This means it is possible to show which actions have led to recognizing an activity. This is a useful way for proving insight in the system, for example, during the evaluation of a mission. Also, by employing a logic-like formalism it is possible to easily adapt, and parametrize the rules. This makes it possible to, for example, have personalized rules for each rescue worker, or to have specific rules for different types of missions.

At the moment the rules are defined manually using domain and expert knowledge. However, certain algorithms might be able to learn the rules from the observed data (as briefly mentioned in Section 2.2.2 Mining-Based). In most cases this does mean that enough data has to be gathered in order for those algorithms to learn the rules. This is problematic in the context of urban search and rescue it is difficult to gather large amounts of data. However, it may be interesting to combine learning from rules with domain and expert knowledge. For example, the rules are initially learned from data, and later refined using domain and expert knowledge.

To summarize, at the moment activities are being recognized based on manually defined rules, which is a complex process due to the verbosity and the limited expressiveness of the language of the logic-like formalism. The verbosity can be relaxed by defining additional entities and relations in the ontology, but another logic-like formalism might be the only way to increase the expressiveness of the language. Possible improvements include the addition of uncertainty, or learning the rules automatically from data.

7.3. Conclusion

A hybrid approach to activity recognition of humans in a human-robot team in the context of urban search and rescue is provided in this thesis. The hybrid approach employs data-driven and knowledge-driven aspects in order to recognize activities from three human behaviours, namely physical motion, communication, and interface actions, and role-based information. Both the behaviours and the rules have various aspects that can be improved upon. The research question whether activities can be recognized in real time for a human-robot rescue team, and if both individual and team activities can be recognized, is not trivially answered. Compared to other approaches, such as most common activity, and random activities, the activity recognition system outperforms all or none, depending on the evaluation measure. Also, while performing activity recognition just for the sake of recognizing activities is scientifically relevant, without knowing what the activities are going to be used for it is difficult to determine which evaluation measures are important. Activity recognition as a research area is mature enough to place a larger focus on the development of useful applications using recognized activities. Also, knowing what the recognized activities are going to be used for, makes it easier to define what sources of information are relevant. However, in any case, at the moment the performance of the activity recognition system is not sufficient for the standards in the context of urban search and rescue.

7.4. Future Research

The idea behind the activity recognition system was to design and develop a generic framework in which various sources of information are combined in order to recognize activities. Both data-driven and knowledge-driven approaches are applied in order to get the most out of the activity recognition system. With the recognized activities other automated and intelligent systems are able to provide support to humans in a human-robot rescue team. In the previous sections a complete overview of most aspects related to the activity recognition system are discussed, however, a clear direction beyond this research is still lacking. Therefore, for the interested reader, several possible future research directions are provided if one is continuing research in this field.

The main purpose behind recognizing activities is to lessen the need for some assumptions of automated support systems. What has been lacking in this research is the definition of relevant information those support systems actually need. While all activities recognized by the activity recognition system are relevant in the context of urban search and rescue, some activities might not provide any additional information for a specific type of support. For example, if an automated support system aims to reduce the physical fatigue of rescue workers, physical activities are relevant than, for example, activities related to robot operation. This means a more explicit focus on actually supporting rescue workers would have provided a better sense of direction in terms of relevant activities, but also in knowing what types of information are useful. The activity recognition system in the current form uses three types of behaviour, and role-based information in order to infer activities. Even though the reasons for using these types of information are well motivated, a particular type of information, namely spatial information, was not used. While technical limitations made it difficult to receive reliable spatial information, in my opinion, spatial information is able to provide important information about the whereabouts of the agents (both humans, and robots), and the location of objects, and areas. Much of this information is directly applicable to human-robot teams in order to support the rescue work. Also, instead of trying to indirectly support rescue workers using recognized activities, the framework is flexible enough to define rules to directly provide support to rescue workers. This makes it possible to directly go to supporting rescue workers instead of using recognized activities as a intermediate step. Though, in a sense activities are still recognized, but the focus is placed on supporting rescue workers, and not on recognizing activities.

Acknowledgments

This research is supported by the EU FP7 ICT Programme Project 609763 (TRADR; www.tradr-project.eu). We would like to thank the fire fighters of FDDo (Dortmund, Germany), GB (Rotterdam, Netherlands) and of Prato, Italy, for their support.

Bibliography

Aggarwal, J. and M. Ryoo (Apr. 2011). “Human activity analysis”. In: *ACM Computing Surveys* 43.3, pp. 1–43. ISSN: 03600300. DOI: 10.1145/1922649.1922653.

Allen, J. (Nov. 1983). “Maintaining knowledge about temporal intervals”. In: *Communications of the ACM* 26.11, pp. 832–843. ISSN: 00010782. DOI: 10.1145/182.358434.

AscTec. *Pelican*. URL: http://www.asctec.de/en/uav-uas-drone-products/asctec-pelican/ (visited on 03/27/2015).

Bao, L. and S. Intille (2004). *Activity Recognition from User-Annotated Acceleration Data*, pp. 1–17. ISBN: 3540218351. DOI: 10.1007/b96922.

BlueBotics. *Absolem*. URL: http://www.bluebotics.com/mobile-robotics/absolem/ (visited on 03/27/2015).

Burke, J. and R. Murphy (2004). “Situation awareness and task performance in robot-assisted technical search: Bujold goes to Bridgeport”. In:

Burke, J., R. Murphy, et al. (2004). “Moonlight in Miami: Field Study of Human-Robot Interaction in the Context of an Urban Search and Rescue Disaster Response Training Exercise”. In: *Human–Computer Interaction* 19.1-2, pp. 85–116. DOI: 10.1080/07370024.2004.9667341.

Casper, J. and R. Murphy (Jan. 2003). “Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center.” English. In: *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society* 33.3, pp. 367–85. ISSN: 1083-4419. DOI: 10.1109/TSMCB.2003.811794.

Chen, D., J. Yang, and H. Wactlar (2004). “Towards automatic analysis of social interaction patterns in a nursing home environment from video”. In: *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval - MIR ’04*. New York, New York, USA: ACM Press, p. 283. ISBN: 1581139403. DOI: 10.1145/1026711.1026757.

Chen, L., C. Nugent, and H. Wang (June 2012). “A Knowledge-Driven Approach to Activity Recognition in Smart Homes”. In: *IEEE Transactions on Knowledge and Data Engineering* 24.6, pp. 961–974. ISSN: 1041-4347. DOI: 10.1109/TKDE.2011.51.

Clark & Parsia (2014). *Stardog*. URL: http://stardog.com/ (visited on 02/19/2015).

Courtemanche, F. et al. (May 2011). “Activity recognition using eye-gaze movements and traditional interactions”. In: *Interacting with Computers* 23.3, pp. 202–213. ISSN: 09535438. DOI: 10.1016/j.intcom.2011.02.008.

Diggelen, J. van, M. Grootjen, and E. Ubink (2013). “Ambient Intelligence - Software and Applications”. In: Advances in Intelligent Systems and Computing 219. Ed. by Ad van Berlo et al., pp. 1–8. DOI: 10.1007/978-3-319-00566-9.

Federal Emergency Management Agency (2015). *Urban Search & Rescue*. URL: http://www.fema.gov/urban-search-rescue (visited on 03/06/2015).

Franz Inc. (2015). *AllegroGraph*. URL: http://franz.com/agraph/allegrograph/ (visited on 08/01/2015).

Giele, T. et al. (2015). “Dynamic Task Allocation for Human-Robot Teams”. In: *ICAART* 7.

Hakeem, A. and M. Shah (2004). “Ontology and taxonomy collaborated framework for meeting classification”. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*. IEEE, 219–222 Vol.4. ISBN: 0-7695-2128-2. DOI: 10.1109/ICPR.2004.1333743.

Herman, I., S. Fernández, and C. Tejo (2015). *SPARQL Endpoint interface to Python*. URL: https://rdflib.github.io/sparqlwrapper/ (visited on 07/13/2015).

Horrocks, I. et al. (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. URL: http://www.w3.org/Submission/SWRL/ (visited on 07/24/2015).

Jordan, A. (2002). "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes". In: *Advances in neural information processing systems*.

Kaber, D. and M. Endsley (Mar. 2004). "The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task". In: *Theoretical Issues in Ergonomics Science* 5.2, pp. 113–153. ISSN: 1463-922X. DOI: 10.1080/1463922021000054335.

Kinova Robotics. *JACO*. URL: http://kinovarobotics.com/products/jaco-robotics/ (visited on 03/27/2015).

Kruijff, G. et al. (2014). "Designing , developing , and deploying systems to support human – robot teams in disaster response". In: 28.23, pp. 1547–1570.

Kwapisz, J., G. Weiss, and S. Moore (Mar. 2011). "Activity recognition using cell phone accelerometers". In: *ACM SIGKDD Explorations Newsletter* 12.2, p. 74. ISSN: 19310145. DOI: 10.1145/1964897.1964918.

Lara, O. and M. Labrador (Jan. 2013). "A Survey on Human Activity Recognition using Wearable Sensors". In: *IEEE Communications Surveys & Tutorials* 15.3, pp. 1192–1209. ISSN: 1553-877X. DOI: 10.1109/SURV.2012.110112.00192.

Lerman, K. (Mar. 2006). "Analysis of Dynamic Task Allocation in Multi-Robot Systems". In: *The International Journal of Robotics Research* 25.3, pp. 225–241. ISSN: 0278-3649. DOI: 10.1177/0278364906063426.

Lester, J., T. Choudhury, and N. Kern (2005). "A Hybrid Discriminative/Generative Approach for Modeling Human Activities". In: *IJCAI*.

Meditskos, G. et al. (Mar. 2013). "SP-ACT: A hybrid framework for complex activity recognition combining OWL and SPARQL rules". In: *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. IEEE, pp. 25–30. ISBN: 978-1-4673-5077-8. DOI: 10.1109/PerComW.2013.6529451.

Messina, E. and A. Jacoff (May 2006). "Performance Standards for Urban Search and Rescue Robots". In: ed. by Grant R. Gerhart, Charles M. Shoemaker, and Douglas W. Gage, pages. DOI: 10.1117/12.663320.

Murphy, R. (May 2004). "Human–Robot Interaction in Rescue Robotics". In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 34.2, pp. 138–153. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2004.826267.

Murphy, R. and J. Burke (Sept. 2005). "Up from the Rubble: Lessons Learned about HRI from Search and Rescue". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 49.3, pp. 437–441. ISSN: 1071-1813. DOI: 10.1177/154193120504900347.

National Institute of Standards and Technology (2015). "National Institute of Standards and Technology". In: URL: http://nist.gov/.

Natvig, T. (2015). *Mumble*. URL: http://www.mumble.info (visited on 07/13/2015).

Neerincx, M. and J. Lindenberg (2008). "Situated cognitive engineering for complex task environments". In: *Naturalistic decision making and macrocognition*, pp. 373–390.

O'Connor, M. and A. Das (2011). "A Method for Representing and Querying Temporal Information in OWL". In: *Communications in Computer and Information Science*. Communications in Computer and Information Science 127. Ed. by Ana Fred, Joaquim Filipe, and Hugo Gamboa, pp. 97–110. ISSN: 18650929. DOI: 10.1007/978-3-642-18472-7.

Open Source Robotics Foundation (2015). *Robot Operating System*. URL: http://www.ros.org/ (visited on 08/01/2015).

Parasuraman, R., M. Barnes, et al. (2007). "Adaptive Automation for Human-Robot Teaming in Future Command and Control Systems". In:

Parasuraman, R., K. Cosenzo, and E. de Visser (2009). "Adaptive automation for human supervision of multiple uninhabited vehicles: Effects on change detection, situation awareness, and mental workload." In: *Military Psychology* 21.2, pp. 270–297. ISSN: 1532-7876. DOI: 10.1080/08995600902768800.

Patterson, D. et al. (2005). "Fine-grained activity recognition by aggregating abstract object usage". In: *Proceedings - International Symposium on Wearable Computers, ISWC*. Vol. 2005, pp. 44–51. ISBN: 0769524192. DOI: 10.1109/ISWC.2005.22.

Pérez, J., M. Arenas, and C. Gutierrez (Aug. 2009). "Semantics and complexity of SPARQL". In: *ACM Transactions on Database Systems* 34.3, pp. 1–45. ISSN: 03625915. DOI: 10.1145/1567274.1567278.

Ravi, N. et al. (2009). *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*. Ed. by Sigeru Omatu et al. Vol. 5518. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1541–1546. ISBN: 978-3-642-02480-1. DOI: 10.1007/978-3-642-02481-8.

Riboni, D. and C. Bettini (Aug. 2010). "COSAR: hybrid reasoning for context-aware activity recognition". In: *Personal and Ubiquitous Computing* 15.3, pp. 271–289. ISSN: 1617-4909. DOI: 10.1007/s00779-010-0331-7.

Riboni, D., L. Pareschi, et al. (Mar. 2011). "Is ontology-based activity recognition really effective?" In: *2011 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 427–431. DOI: 10.1109/PERCOMW.2011.5766927.

Samsung (2012). *Samsung Galaxy SIII*. URL: http://www.samsung.com/global/galaxys3/ (visited on 07/12/2015).

Semantic Web Company (2015). *Virtuoso Universal Server*. URL: https://www.semantic-web.at/virtuoso-universal-server (visited on 08/01/2015).

Shet, V., D. Harwood, and L. Davis (2005). "VidMAP: video monitoring of activity with prolog". In: *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.* Pp. 224–229. DOI: 10.1109/AVSS.2005.1577271.

Shoaib, M. et al. (Jan. 2015). "A survey of online activity recognition using mobile phones." In: *Sensors (Basel, Switzerland)* 15.1, pp. 2059–85. ISSN: 1424-8220. DOI: 10.3390/s150102059.

Sokolova, M. and G. Lapalme (July 2009). "A systematic analysis of performance measures for classification tasks". In: *Information Processing & Management* 45.4, pp. 427–437. ISSN: 03064573. DOI: 10.1016/j.ipm.2009.03.002.

Stanford Center for Biomedical Informatics Research (2014). *Protégé*. URL: http://protege.stanford.edu/ (visited on 02/19/2015).

Turaga, P. et al. (Nov. 2008). "Machine Recognition of Human Activities: A Survey". In: *IEEE Transactions on Circuits and Systems for Video Technology* 18.11, pp. 1473–1488. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2008.2005594.

Visser, E. de and R. Parasuraman (June 2011). "Adaptive Aiding of Human-Robot Teaming: Effects of Imperfect Automation on Performance, Trust, and Workload". In: *Journal of Cognitive Engineering and Decision Making* 5.2, pp. 209–231. ISSN: 1555-3434. DOI: 10.1177/1555343411410160.

W3C (2013). *SPARQL 1.1 Overview*. URL: http://www.w3.org/TR/sparql11-overview/ (visited on 07/24/2015).

Waugh, W. and G. Streib (Dec. 2006). "Collaboration and Leadership for Effective Emergency Management". In: *Public Administration Review* 66.s1, pp. 131–140. ISSN: 0033-3352. DOI: 10.1111/j.1540-6210.2006.00673.x.

Wikipedia (2015). *Universally unique identifier*. URL: https://en.wikipedia.org/wiki/Universally%5C_unique%5C_identifier (visited on 07/20/2015).

Wyss, T. and U. Mäder (Nov. 2010). "Recognition of Military-Specific Physical Activities With Body-Fixed Sensors". In: *Military Medicine* 175.11, pp. 858–864. ISSN: 0026-4075. DOI: 10.7205/MILMED-D-10-00023.

Yanco, H. et al. (2015). "Analysis of Human-robot Interaction at the DARPA Robotics Challenge Trials". In: 32.June, pp. 420–444. DOI: 10.1002/rob.

Ye, J., G. Stevenson, and S. Dobson (June 2011). "A top-level ontology for smart environments". In: *Pervasive and Mobile Computing* 7.3, pp. 359–378. ISSN: 15741192. DOI: 10.1016/j.pmcj.2011.02.002.

Zhu, C. and W. Sheng (May 2009). "Human daily activity recognition in robot-assisted living using multi-sensor fusion". In: *2009 IEEE International Conference on Robotics and Automation*. IEEE, pp. 2154–2159. ISBN: 978-1-4244-2788-8. DOI: 10.1109/ROBOT.2009.5152756.

A. Activity Recognition Rules

In the following chapter all rules used by the reasoning process are listed per activity.

A.1. Instructing

```
CONSTRUCT
{
    ?activity a :Instructing ;
              :performedBy ?human2 ;
              :performedWith ?human1 ;
              temporal:hasStartTime ?startTime1 ;
              temporal:hasFinishTime ?finishTime2 .
}
WHERE
{
    # Request.
    ?action1 a :Talking ;
        :performedBy ?human1 ;
        :performedWith ?human2 ;
        temporal:hasStartTime ?startTime1 ;
        temporal:hasFinishTime ?finishTime1 .

    # Response.
    ?action2 a :Talking ;
        :performedBy ?human2 ;
        :performedWith ?human1 ;
        temporal:hasStartTime ?startTime2 ;
        temporal:hasFinishTime ?finishTime2 .
    ?human2 :hasRole :TeamLeader .

    # Check if the request is finished before the start of the response.
    FILTER(?finishTime1 < ?startTime2) .

    # Check if there is a short pause (of maximum 5 seconds) in between the request and the response.
    FILTER(temporal:durationGreaterThan(5, ?finishTime1, ?startTime2, temporal:Seconds)) .

    # Check if the response is longer than the request.
    BIND(temporal:duration(?startTime1, ?finishTime1, temporal:Seconds) AS ?duration1) .
    BIND(temporal:duration(?startTime2, ?finishTime2, temporal:Seconds) AS ?duration2) .

    FILTER(?duration2 > ?duration1) .

    BIND(IRI(CONCAT(STR(:Instructioning), "+", STR(?action1))) AS ?activity) .
}
```

Listing A.1: Rule for the activity *instructing* (1).

```
CONSTRUCT
{
    ?activity a :Instructing ;
              :performedBy ?human1 ;
              :performedWith ?human2 ;
              temporal:hasStartTime ?startTime ;
              temporal:hasFinishTime ?finishTime .
}
WHERE
{
    # Request.
    ?action a :Talking ;
        :performedBy ?human1 ;
        :performedWith ?human2 ;
        temporal:hasStartTime ?startTime ;
        temporal:hasFinishTime ?finishTime .
    ?human1 :hasRole :TeamLeader .

    BIND(IRI(CONCAT(STR(:Instructioning), "+", STR(?action))) AS ?activity) .
}
```

Listing A.2: Rule for the activity *instructing* (2).

A.2. Informing

```
CONSTRUCT
{
    ?activity a :Informing ;
              :performedBy ?human1 ;
              :performedWith ?human2 ;
              temporal:hasStartTime ?startTime1 ;
              temporal:hasFinishTime ?finishTime2 .
}
WHERE
{
    # Report.
    ?action1 a :Talking ;
        :performedBy ?human1 ;
        :performedWith ?human2 ;
        temporal:hasStartTime ?startTime1 ;
        temporal:hasFinishTime ?finishTime1 .

    # Response.
    ?action2 a :Talking ;
        :performedBy ?human2 ;
        :performedWith ?human1 ;
        temporal:hasStartTime ?startTime2 ;
        temporal:hasFinishTime ?finishTime2 .
    ?human2 :hasRole :TeamLeader .

    # Check if the report is finished before the start of the response.
    FILTER(?finishTime1 < ?startTime2) .

    # Check if there is a short pause (of maximum 5 seconds) in between the report and the response.
    FILTER(temporal:durationGreaterThan(5, ?finishTime1, ?startTime2, temporal:Seconds)) .

    # Check if the report is longer than the request.
    BIND(temporal:duration(?startTime1, ?finishTime1, temporal:Seconds) AS ?duration1) .
    BIND(temporal:duration(?startTime2, ?finishTime2, temporal:Seconds) AS ?duration2) .

    FILTER(?duration1 > ?duration2) .

    BIND(IRI(CONCAT(STR(:Informing), "+", STR(?action1))) AS ?activity) .
}
```

Listing A.3: Rule for the activity *informing* (1).

```
CONSTRUCT
{
    ?activity a :Informing ;
              :performedBy ?human1 ;
              :performedWith ?human2 ;
              temporal:hasStartTime ?startTime ;
              temporal:hasFinishTime ?finishTime .
}
WHERE
{
    # Report.
    ?action a :Talking ;
        :performedBy ?human1 ;
        :performedWith ?human2 ;
        temporal:hasStartTime ?startTime ;
        temporal:hasFinishTime ?finishTime .
    ?human2 :hasRole :TeamLeader .

    BIND(IRI(CONCAT(STR(:Informing), "+", STR(?action))) AS ?activity) .
}
```

Listing A.4: Rule for the activity *informing* (2).

A.3. Discussing

```
CONSTRUCT
{
    ?activity a :Discussing ;
              :performedBy ?human2 ;
              :performedWith ?human1 ;
              temporal:hasStartTime ?startTime1 ;
              temporal:hasFinishTime ?finishTime5 .
}
WHERE
{
    # Part 1.
    ?action1 a :Talking ;
            :performedBy ?human1 ;
            :performedWith ?human2 ;
            temporal:hasStartTime ?startTime1 ;
```

```
            temporal:hasFinishTime ?finishTime1 .

    # Part 2.
    ?action2 a :Talking ;
            :performedBy ?human2 ;
            :performedWith ?human1 ;
            temporal:hasStartTime ?startTime2 ;
            temporal:hasFinishTime ?finishTime2 .

    # Part 3.
    ?action3 a :Talking ;
            :performedBy ?human1 ;
            :performedWith ?human2 ;
            temporal:hasStartTime ?startTime3 ;
            temporal:hasFinishTime ?finishTime3 .

    # Part 4.
    ?action4 a :Talking ;
            :performedBy ?human2 ;
            :performedWith ?human1 ;
            temporal:hasStartTime ?startTime4 ;
            temporal:hasFinishTime ?finishTime4 .

    # Part 5.
    ?action5 a :Talking ;
            :performedBy ?human1 ;
            :performedWith ?human2 ;
            temporal:hasStartTime ?startTime5 ;
            temporal:hasFinishTime ?finishTime5 .

    # Part 1 < Part 2 < Part 3 < Part 4 < Part 5.
    FILTER(?finishTime1 < ?startTime2) .
    FILTER(?finishTime2 < ?startTime3) .
    FILTER(?finishTime3 < ?startTime4) .
    FILTER(?finishTime4 < ?startTime5) .

    # Check if there is a short pause between Part 1 - Part 2, Par 2 - Part 3, Part 3 - Part 4, and Part 4 - Part 5.
    FILTER(temporal:durationGreaterThan(5, ?finishTime1, ?startTime2, temporal:Seconds)) .
    FILTER(temporal:durationGreaterThan(5, ?finishTime2, ?startTime3, temporal:Seconds)) .
    FILTER(temporal:durationGreaterThan(5, ?finishTime3, ?startTime4, temporal:Seconds)) .
    FILTER(temporal:durationGreaterThan(5, ?finishTime4, ?startTime5, temporal:Seconds)) .

    BIND(IRI(CONCAT(STR(:Discussing), "+", STR(?action1), "+", STR(?action2), "+", STR(?action3), "+", STR(?action4))) AS ?activity) .
}
```

Listing A.5: Rule for the activity *discussing* (1).

```
CONSTRUCT
{
    ?activity a :Discussing ;
              :performedBy ?human2 ;
              :performedWith ?human1 ;
              temporal:hasStartTime ?startTime1 ;
              temporal:hasFinishTime ?finishTime4 .
}
WHERE
{
    # Part 1.
    ?action1 a :Talking ;
            :performedBy ?human1 ;
            :performedWith ?human2 ;
            temporal:hasStartTime ?startTime1 ;
            temporal:hasFinishTime ?finishTime1 .

    # Part 2.
    ?action2 a :Talking ;
            :performedBy ?human2 ;
            :performedWith ?human1 ;
            temporal:hasStartTime ?startTime2 ;
            temporal:hasFinishTime ?finishTime2 .

    # Part 3.
    ?action3 a :Talking ;
            :performedBy ?human1 ;
            :performedWith ?human2 ;
            temporal:hasStartTime ?startTime3 ;
            temporal:hasFinishTime ?finishTime3 .

    # Part 4.
    ?action4 a :Talking ;
            :performedBy ?human2 ;
            :performedWith ?human1 ;
            temporal:hasStartTime ?startTime4 ;
            temporal:hasFinishTime ?finishTime4 .

    # Part 1 < Part 2 < Part 3 < Part 4.
    FILTER(?finishTime1 < ?startTime2) .
    FILTER(?finishTime2 < ?startTime3) .
    FILTER(?finishTime3 < ?startTime4) .

    # Check if there is a short pause between Part 1 - Part 2, Par 2 - Part 3, and Part 3 - Part 4.
    FILTER(temporal:durationGreaterThan(5, ?finishTime1, ?startTime2, temporal:Seconds)) .
    FILTER(temporal:durationGreaterThan(5, ?finishTime2, ?startTime3, temporal:Seconds)) .
    FILTER(temporal:durationGreaterThan(5, ?finishTime3, ?startTime4, temporal:Seconds)) .

```

```
    BIND(IRI(CONCAT(STR(:Discussing), "+", STR(?action1), "+", STR(?action2), "+", STR(?action3), "+", STR(?action4))) AS ?activity) .
}
```

Listing A.6: Rule for the activity *discussing* (2).

A.4. Avoiding Obstacle

```
CONSTRUCT
{
    ?activity a :AvoidingObstacle ;
              :performedBy ?human ;
              temporal:hasStartTime ?startTime1 ;
              temporal:hasFinishTime ?finishTime3 .
}
WHERE
{
    # MovingFlippers.
    ?action1 a :MovingFlippers ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime1 ;
            temporal:hasFinishTime ?finishTime1 .

    # Driving.
    ?action2 a :Driving ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime2 ;
            temporal:hasFinishTime ?finishTime2 .

    # MovingFlippers.
    ?action3 a :MovingFlippers ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime3 ;
            temporal:hasFinishTime ?finishTime3 .

    FILTER(?finishTime1 < ?startTime2) .
    FILTER(?finishTime2 < ?startTime3) .

    # Check if there is a short pause between MovingFlippers - Driving - MovingFlippers.
    FILTER(temporal:durationGreaterThan(10, ?finishTime1, ?startTime2, temporal:Seconds)) .
    FILTER(temporal:durationGreaterThan(10, ?finishTime2, ?startTime3, temporal:Seconds)) .

    BIND(IRI(CONCAT(STR(:AvoidingObstacle), "+", STR(?action1), "+", STR(?action2))) AS ?activity) .
}
```

Listing A.7: Rule for the activity *avoiding obstacle* (1).

```
CONSTRUCT
{
    ?activity a :AvoidingObstacle ;
              :performedBy ?human ;
              temporal:hasStartTime ?startTime1 ;
              temporal:hasFinishTime ?finishTime2 .
}
WHERE
{
    # MovingFlippers.
    ?action1 a :MovingFlippers ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime1 ;
            temporal:hasFinishTime ?finishTime1 .

    # Driving.
    ?action2 a :Driving ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime2 ;
            temporal:hasFinishTime ?finishTime2 .

    # Check if MovingFlippers occurs during Driving.
    FILTER(temporal:during(?startTime1, ?finishTime1, ?startTime2, ?finishTime2, temporal:Seconds)) .

    BIND(IRI(CONCAT(STR(:AvoidingObstacle), "+", STR(?action1), "+", STR(?action2))) AS ?activity) .
}
```

Listing A.8: Rule for the activity *avoiding-obstacle* (2).

A.5. Looking Around

```
CONSTRUCT
{
    ?activity a :LookingAround ;
              :performedBy ?human ;
              temporal:hasStartTime ?startTime2 ;
              temporal:hasFinishTime ?finishTime2 .
}
WHERE
{
    # Driving.
    ?action1 a :Driving ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime1 ;
            temporal:hasFinishTime ?finishTime1 .

    # Turning.
    ?action2 a :Turning ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime2 ;
            temporal:hasFinishTime ?finishTime2 .

    # Driving.
    ?action3 a :Driving ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime3 ;
            temporal:hasFinishTime ?finishTime3 .

    FILTER(?finishTime1 < ?startTime2) .
    FILTER(?finishTime2 < ?startTime3) .

    # Check if there is a long pause (between 15 - 30 seconds) between Driving - Turning - Driving.
    FILTER(temporal:durationLessThan(15, ?finishTime1, ?startTime2, temporal:Seconds)) .
    FILTER(temporal:durationGreaterThan(30, ?finishTime1, ?startTime2, temporal:Seconds)) .

    FILTER(temporal:durationLessThan(15, ?finishTime2, ?startTime3, temporal:Seconds)) .
    FILTER(temporal:durationGreaterThan(30, ?finishTime2, ?startTime3, temporal:Seconds)) .

    # Check if the duration of Turning is long (more than 6 seconds).
    FILTER(temporal:durationLessThan(6.0, ?startTime2, ?finishTime2, temporal:Seconds)) .

    BIND(IRI(CONCAT(STR(:LookingAround), "+", STR(?action2))) AS ?activity) .
}
```

Listing A.9: Rule for the activity *looking-around* (1).

```
CONSTRUCT
{
    ?activity a :LookingAround ;
              :performedBy ?human ;
              temporal:hasStartTime ?startTime2 ;
              temporal:hasFinishTime ?finishTime2 .
}
WHERE
{
    # Driving.
    ?action1 a :Driving ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime1 ;
            temporal:hasFinishTime ?finishTime1 .

    # Turning.
    ?action2 a :Turning ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime2 ;
            temporal:hasFinishTime ?finishTime2 .

    FILTER(?finishTime1 < ?startTime2) .

    # Check if there is a long pause (between 15 - 30 seconds) between Driving - Turning.
    FILTER(temporal:durationLessThan(15, ?finishTime1, ?startTime2, temporal:Seconds)) .
    FILTER(temporal:durationGreaterThan(30, ?finishTime1, ?startTime2, temporal:Seconds)) .

    # Check if the duration of Turning is long (more than 6 seconds).
    FILTER(temporal:durationLessThan(6.0, ?startTime2, ?finishTime2, temporal:Seconds)) .

    BIND(IRI(CONCAT(STR(:LookingAround), "+", STR(?action2))) AS ?activity) .
}
```

Listing A.10: Rule for the activity *looking-around* (2).

```
CONSTRUCT
{
    ?activity a :LookingAround ;
              :performedBy ?human ;
              temporal:hasStartTime ?startTime1 ;
              temporal:hasFinishTime ?finishTime1 .
}
WHERE
{
    # Turning.
    ?action1 a :Turning ;
```

```
            :performedBy ?human ;
            temporal:hasStartTime ?startTime1 ;
            temporal:hasFinishTime ?finishTime1 .

    # Driving.
    ?action2 a :Driving ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime2 ;
            temporal:hasFinishTime ?finishTime2 .

    FILTER(?finishTime1 < ?startTime2) .

    # Check if there is a long pause (between 15 - 30 seconds) between Turning - Driving.
    FILTER(temporal:durationLessThan(15, ?finishTime1, ?startTime2, temporal:Seconds)) .
    FILTER(temporal:durationGreaterThan(30, ?finishTime1, ?startTime2, temporal:Seconds)) .

    # Check if the duration of Turning is long (more than 6 seconds).
    FILTER(temporal:durationLessThan(6.0, ?startTime1, ?finishTime1, temporal:Seconds)) .

    BIND(IRI(CONCAT(STR(:LookingAround), "+", STR(?action1))) AS ?activity) .
}
```

Listing A.11: Rule for the activity *looking-around* (3).

A.6. Searching

```
CONSTRUCT
{
    ?activity a :Searching ;
              :performedBy ?human ;
              temporal:hasStartTime ?startTime1 ;
              temporal:hasFinishTime ?finishTime3 .
}
WHERE
{
    # MovingCamera.
    ?action1 a :MovingCamera ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime1 ;
            temporal:hasFinishTime ?finishTime1 .

    # ZoomingCamera.
    ?action2 a :ZoomingCamera ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime2 ;
            temporal:hasFinishTime ?finishTime2 .

    # MovingCamera.
    ?action3 a :MovingCamera ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime3 ;
            temporal:hasFinishTime ?finishTime3 .

    FILTER(?finishTime1 < ?startTime2) .
    FILTER(?finishTime2 < ?startTime3) .

    # Check if there is a short pause between MovingCamera - ZoomingCamera - MovingCamera.
    FILTER(temporal:durationGreaterThan(10, ?finishTime1, ?startTime2, temporal:Seconds)) .
    FILTER(temporal:durationGreaterThan(10, ?finishTime2, ?startTime3, temporal:Seconds)) .

    BIND(IRI(CONCAT(STR(:Searching), "+", STR(?action1), "+", STR(?action2))) AS ?activity) .
}
```

Listing A.12: Rule for the activity *searching* (1).

```
CONSTRUCT
{
    ?activity a :Searching ;
              :performedBy ?human ;
              temporal:hasStartTime ?startTime1 ;
              temporal:hasFinishTime ?finishTime2 .
}
WHERE
{
    # MovingCamera.
    ?action1 a :MovingCamera ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime1 ;
            temporal:hasFinishTime ?finishTime1 .

    # ZoomingCamera.
    ?action2 a :ZoomingCamera ;
            :performedBy ?human ;
            temporal:hasStartTime ?startTime2 ;
            temporal:hasFinishTime ?finishTime2 .

```

```
    FILTER(?finishTime1 < ?startTime2) .

    # Check if there is a short pause between MovingCamera - ZoomingCamera.
    FILTER(temporal:durationGreaterThan(10, ?finishTime1, ?startTime2, temporal:Seconds)) .

    BIND(IRI(CONCAT(STR(:Searching), "+", STR(?action1), "+", STR(?action2))) AS ?activity) .
}
```

Listing A.13: Rule for the activity *searching* (2).

```
CONSTRUCT
{
    ?activity a :Searching ;
               :performedBy ?human ;
               temporal:hasStartTime ?startTime1 ;
               temporal:hasFinishTime ?finishTime2 .
}
WHERE
{
    # ZoomingCamera.
    ?action1 a :ZoomingCamera ;
             :performedBy ?human ;
             temporal:hasStartTime ?startTime1 ;
             temporal:hasFinishTime ?finishTime1 .

    # MovingCamera.
    ?action2 a :MovingCamera ;
             :performedBy ?human ;
             temporal:hasStartTime ?startTime2 ;
             temporal:hasFinishTime ?finishTime2 .

    FILTER(?finishTime1 < ?startTime2) .

    # Check if there is a short pause between ZoomingCamera - MovingCamera.
    FILTER(temporal:durationGreaterThan(10, ?finishTime1, ?startTime2, temporal:Seconds)) .

    BIND(IRI(CONCAT(STR(:Searching), "+", STR(?action2), "+", STR(?action1))) AS ?activity) .
}
```

Listing A.14: Rule for the activity *searching* (3).

B. Semantic Web Rule Language (SWRL)

The initial approach for inferring activities from actions was based on the Semantic Web Rule Language (SWRL) (Horrocks et al. 2004). SWRL is based on a combination of OWL DL, OWL Lite, and a subset of the Rule Markup Language. With SWRL it is possible to define rules in the form of an implication, consisting on an antecedent (head), and a consequent (body). All conditions in either the antecedent or the consequent are in conjunctive form. Whenever all conditions in the antecedent hold, the conditions in the consequent hold as well. Listing B.1 shows an example of the syntax of a rule in SWRL.

The reason why the implementation changed from SWRL to SPARQL is due to a technical issue. Stardog does not allow the creation of new individuals with multiple properties. Since each activity has at least a type, a performer, a start time, and a finish time, it was not possible to use SWRL to define the mappings from actions to activities.

```
1  Talking(?activity1) /\
2    performedBy(?activity1, ?human1) /\
3    performedWith(?activity1, ?human2) /\
4    temporal:hasStartTime(?activity1, ?startTime1) /\
5    temporal:hasFinishTime(?activity1, ?finishTime1) /\
6  Talking(?activity2) /\
7    performedBy(?activity2, ?human2) /\
8    performedWith(?activity2, ?human1) /\
9    temporal:hasStartTime(?activity2, ?startTime1) /\
10   temporal:hasFinishTime(?activity2, ?finishTime1) /\
11 temporal:before(?finishTime1, ?startTime2) /\
12 temporal:durationGreaterThan(5, ?finishTime1, ?startTime2, temporal:Seconds) /\
13 temporal:duration(?duration1, ?startTime1, ?finishTime1, temporal:Seconds) /\
14 temporal:duration(?duration2, ?startTime2, ?finishTime2, temporal:Seconds) /\
15 swrlb:greaterThan(?duration1, ?duration2)
16 ->
17 Informing(?activity1) /\
18 performedBy(?activity1, ?human1) /\
19 performedWith(?activity1, ?human2) /\
20 temporal:hasStartTime(?activity1, ?startTime1) /\
21 temporal:hasFinishTime(?activity2, ?finishTime2)
```

Listing B.1: The rule is a realization of the activity *informing* in the SWRL language. The syntax is correct except at line 17 when a new individual has to be created.

C. Evaluation Scenario

The following chapter provides an overview of the complete scenario used during the evaluation during T-JEx. The scenario consisted of two sorties with a short break in between during which a briefing was held. Figure C.1 provides a schematic overview of the location. The actual furnaces are displayed by two inaccessible circles, surrounded by a dark blue square.

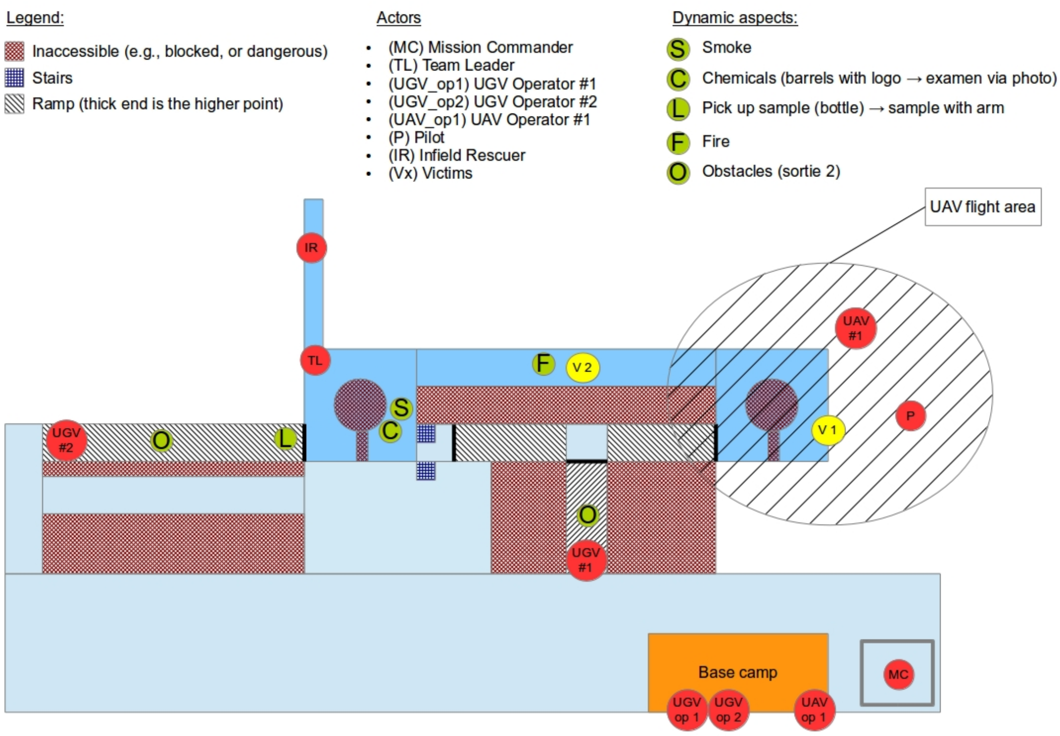


Figure C.1.: Schematic map of the scenario

C.1. Sortie 1

The objective of the first sortie is to use the UAV to search one furnace (on the right) for victims, and to use the UGV's to thoroughly search the other furnace (on the left) for victims, and potential hazards.

UGV1

1. UGV1 drives up the ramp

2. UGV1 takes a left and traverse the stairs
3. UGV1 drives on (visibility is limited due to smoke)
4. UGV1 detects chemical barrels. Puts information into the TRADR system. UGV2 is notified to take a sample.
5. UGV1 searches for victims via camera and heat camera and takes an audio sample
6. UGV1 finds victim and communicates via audio
7. UGV1 searches further, goes right into the direction of the fire
8. Go-home call is received, UGV1 drives to the starting location (autonomously if possible)

UGV2

1. UGV2 drives up ramp
2. UGV2 detects chemicals and takes a sample (pick up bottle)
3. UGV2 continues to drive to the bridge and requests hand-off of sample with in-field rescuer
4. UGV2 gets notification that help is needed to take a camera picture of chemical barrels
5. UGV2 drives past the fire to chemical barrels, takes picture, and goes to the starting location (autonomously if possible)

UAV

1. UAV provides an overview of the situation
2. UAV finds victim using pictures of camera

In-field rescuer

1. In-field rescuer walks up the stairs
2. In-field rescuer receives UGV2 request for the hand-off of the sample
3. In-field rescuer walks to the hand-off location and takes the sample

C.2. Sortie 2

Between the first and second sortie victims have been rescued, and the fire is extinguished. Some parts of the building have collapsed obstructing parts of the scenario (i.e., obstacles such as pallets and iron bars). This relates to the tasks of the sortie, namely building assessment. However, from an evaluation point-of-view the motivation behind a first and a second sortie is to gain insights into what and how information is communicated between sorties. Also, at some point during the second sortie the network will be intentionally disrupted to investigate the robustness of the TRADR system. For example, to investigate what happens with the robots, and what kind of issues to the team members experience.

UGV1

1. UGV1 drives autonomously towards furnace
2. UGV1 finds objects on the path
3. UGV1 either autonomously drives around or over the objects, or the robot operator takes over control

UGV2

1. UGV2 drives autonomously towards furnace
2. UGV2 finds objects on the path
3. UGV2 either autonomously drives around or over the objects, or the robot operator takes over control

UAV

1. UAV is tasked to look at some parts of the building from the outside

In-field rescuer

1. In-field rescuer receives request from UGV2 to remove some obstacles
2. In-field rescuer removes obstacles

D. End-Users Interview

A week after T-JEx an interview with two end-users, myself, and Nanja was held via a phone group conversation. The end-users are Robert and Norbert, both professional fire fighters of the Dortmund fire brigade. Below are the contents of the interview.

1. As a team leader, do you think the activity recognition system changes your awareness of the team members?

Robert
: A benefit would be that I would know what my team members are doing. I know that they are sitting there, and I simply trust what they are doing. I might forget what they are doing, and the yellow-schedule [1] might be interesting, but the actions provide too much detail. It is important to know whether they are working on the task or not.

Norbert
: I agree in a similar fashion, but the information might be more useful for the robot operators. For example, having information in relation to spatial information. The team leader only wants to have facts, and only relevant information for the task of the team leader.

Robert
: Also, if there are multiple robots (e.g., 4 or more) it might be more useful than it is now. Since, the team leader is able to overview all team members (and is actually rather bored).

Norbert
: The activity recognition system could be a good tool for analysing the mission afterwards. Maybe the information can be used to becoming better robot operators.

2. Would you trust the activities recognized by the activity recognition system?

Norbert
: In principle yes. The information can be useful for path recognition, for example to compute whether there is sufficient power to return the robot home.

Robert
: It depends on how it is done. Is it possible there are high-level activities without low-level activities?

Bas No, since the high-level activities are inferred from the low-level activities.

Robert
: Ok, but it is important to understand how they related to each other.

[1] The activities in the viewer.

3. As any member of the team, how comfortable are you with your activities being monitored?

Robert
During T-Jex I was perfectly comfortable, and quite relaxed. Any fire fighter should have the motivation to improve their actions afterwards, but I think that most fire fighters would have problems being monitored all the time.

Norbert
Everyone is different, it depends on the situation.

Bas What about monitoring heart rate? Would you have any privacy concerns?

Robert
Could be problematic for some people. It is not clear how or why the system makes the predictions. Is it because of the behaviour of the fire fighter or more related to the sortie?

Norbert
I don't know, I don't believe that it is possible to make use of such information in a useful way.

Nanja
What about having the robots knowing what the humans are doing?

Robert
If the robot encounters a problem it cannot resolve on it's own, should it report immediately or wait? It depends on whether the problem is important enough or whether the robot operator is busy. Let's say you have 20 icons popping up, then you might click them all away.

Robert
You can reduce the amount of information using a good information flow.

4. As a robot operator or an in-field rescuer, how comfortable are you with your activity being monitored by the team leader?

Robert
Usually the in-field rescuers are closely being watched by the team leader. Having your actions on a monitor is actually further away than when the team leader is physically present on the site.

Norbert
The team leader trusts the team members, and does not wants to observe them. If a team member has a problem then they will ask the team leader again. The team leader does not have the time to monitor them all the time.

Robert
It is related to the personality of a team leader. Some team leaders prefer to the monitor other team members more closely then others, but when the team leader is busy he trusts the team members anyway.

5. Would you use the output of the activity recognition system in the current state?

Robert
The activity recognition system is useful for evaluation afterwards. It is probably difficult to use during a mission, but it could be given a try.

Norbert

It is useful for analysing the mission. The additional information is not needed in general.