The original publication is available at www.springerlink.com
http://link.springer.com/chapter/10.1007/978-3-642-34898-3_7#
AMBIENT INTELLIGENCE, Lecture Notes in Computer Science, 2012,
Volume 7683/2012, 97-112, DOI: 10.1007/978-3-642-34898-3_7

Automatic Behavior Understanding in Crisis Response Control Rooms

Joris IJsselmuiden[1], Ann-Kristin Grosselfinger[1], David Münch[1], Michael Arens[1], and Rainer Stiefelhagen[1,2]

[1] Fraunhofer IOSB, Karlsruhe, Germany
{joris.ijsselmuiden, ann-kristin.grosselfinger, david.muench, michael.arens}@iosb.fraunhofer.de
[2] Karlsruhe Institute of Technology, Karlsruhe, Germany
rainer.stiefelhagen@kit.edu

Abstract. This paper addresses the problem of automatic behavior understanding in smart environments. Automatic behavior understanding is defined as the generation of semantic event descriptions from machine perception. Outputs from available perception modalities can be fused into a world model with a single spatiotemporal reference frame. The fused world model can then be used as input by a reasoning engine that generates semantic event descriptions. We use a newly developed annotation tool to generate hypothetical machine perception outputs instead. The applied reasoning engine is based on fuzzy metric temporal logic (FMTL) and situation graph trees (SGTs), promising and universally applicable tools for automatic behavior understanding. The presented case study is automatic behavior report generation for staff training purposes in crisis response control rooms. Various group formations and interaction patterns are deduced from person tracks, object information, and information about gestures, body pose, and speech activity.

Keywords: automatic behavior understanding, smart environments, rule-based expert systems, fuzzy metric temporal logic, situation graph trees

1 Introduction

In recent years, there has been great progress in computer vision and other areas of machine perception, for example in person tracking and body pose estimation. However, high-level systems using multiple machine perception modalities and combining multiple objects have not progressed at the same pace. We are developing a toolkit for automatic behavior understanding that deploys multimodal machine perception for multiple objects, fuses everything into a world model with a single spatiotemporal reference frame, and generates semantic descriptions about the observed scene. The current system uses a dedicated annotation tool instead of multimodal machine perception as shown in Figure 1.

The presented case study is situated at the State Fire Service Institute (Institut der Feuerwehr) Nordrhein-Westfalen, during one of their staff exercises

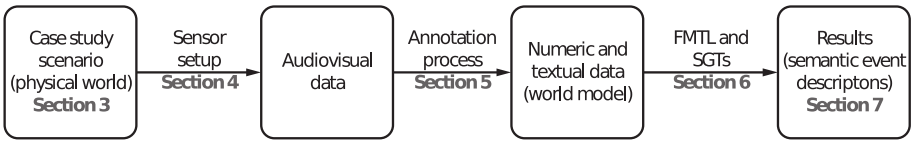

Fig. 1. System overview

for crisis response control room operations (see Figure 2). The task is to automatically generate behavior reports from multimodal machine perception during staff exercises and actual crisis management. These reports about staff behavior in the control room can be used for training purposes, evaluations, and audit trails. For instance, given the identity, position, orientation, and speech activity of the staff members over time, and information about objects in the room, these reports can contain descriptions and visualizations of group formations and interaction patterns, i.e. who was doing what with whom, using which support tools. This can be combined with audiovisual recordings and visualizations, and with the corresponding developments in cyberspace, i.e. field unit status, crisis dynamics, and other context information. Such a system would provide a rich information source, conveniently searchable for specific events.

The presented reasoning process is domain independent because it is separated from any machine perception it might use. The annotation process too is designed to be customizable for other application domains. Possible application domains include other behavior understanding applications, multimedia retrieval, robotics, ambient assisted living, intelligent work environments, intelligent user interfaces, indoor and outdoor surveillance, and situational awareness and decision support for military and civil security. Applied machine perception can range from video to radar, and from person and vessel tracking to body pose estimation, speech recognition, and activities in cyberspace. Other uses include camera control, sensor deployment planning, future event prediction, information exchange between system components, and top-down knowledge for machine perception to guide its search and improve outputs.

This paper is organised as follows. After discussing related work in Section 2, we explain the applied processing chain step by step as depicted in Figure 1. Section 3 describes the case study scenario: automatic behavior report generation for training purposes in crisis response control rooms. A staff exercise was recorded using multiple cameras and microphones with appropriate postprocessing as described in Section 4. The next step is turning the recorded audiovisual data into a world model consisting of numeric and textual data. Ultimately, this should be accomplished using machine perception and multimodal fusion, but we currently use a different approach. Section 5 describes how the postprocessed audiovisual data was manually analysed and annotated using a tool specifically developed for such purposes. The resulting world model forms the input for the reasoning engine based on fuzzy metric temporal logic (FMTL) and situation graph trees (SGTs) presented in Section 6. It delivers semantic descriptions about staff

Fig. 2. Case study scenario from the State Fire Service Institute (Institut der Feuerwehr) Nordrhein-Westfalen: automatic behavior report generation for training purposes in crisis response control rooms. Several events we aim to recognize are visible here: conversation, discussion with document, and editing a display.

behavior, which can be compared to ground-truth results annotated using the annotation tool. Section 7 presents some initial results, Section 8 explains how we can handle imperfect input data, and Section 9 concludes the paper.

The ultimate goal is an integrated system performing all these steps, using multiple machine perception components and multimodal fusion instead of manual annotation. Such a system should run in real time with synchronous visualizations of sensor data, machine perception, and resulting semantic descriptions. In application domains such as robotics and intelligent user interfaces, appropriate embodiment and action generation would be required. Our research is situated in a work environment that focuses on machine perception (especially computer vision) and human-machine interaction, which facilitates the progress toward such an online system. In the meantime, the presented approach improves high-level reasoning processes without the need for corresponding progress in machine perception. And even though behavior reports and visualizations cannot be generated fully automatically yet, our current and future data, observations, and reasoning results can improve understanding of control room operations. The novel contributions of this paper are as follows. Several steps toward an integrated development toolkit were completed, including a new dataset and a new tool for data analysis and annotation. The presented case study is of general interest because of its unique character and its large amount of perception modalities and objects. A newly developed FMTL/SGT knowledge base for this case study is contributed that is also applicable to other domains, along with corresponding experimental results. And we explain how to handle imperfect input data using FMTL and SGTs.

2 Related Work

Surveys on automatic behavior understanding are provided by [1–3]. In [1], a distinction is made between single-layered approaches operating directly on sensor data and hierarchical approaches applying machine perception first and using its output to generate semantic descriptions. In hierarchical systems, semantic event descriptions are usually generated from machine perception outputs using either the statistical approach, the syntactic approach, or the description-based approach. In statistical approaches, event likelihoods are computed by (derivations of) hidden Markov models, (dynamic) Bayesian networks, propagation networks, or similar models [4–6]. In syntactic approaches, atomic events are combined into complex events using formal (stochastic) grammars, mapping spatiotemporal changes in image sequences to events for instance [7–9]. And description-based approaches use formal languages such as logics and and-or graphs for representing and reasoning about spatiotemporal dynamics [10–14]. Statistical and description-based approaches are combined in Markov logic networks by [15–17]. Similarly, Bayesian compositional hierarchies are combined with rule generation from ontologies in [18]. Related studies on smart environments, surveillance, and other applications are found in [19–21]. And [22, 23] present two relevant studies from the field of crisis management.

Our own hierarchical description-based approach to automatic behavior understanding uses fuzzy metric temporal logic (FMTL) combined with situation graph trees (SGTs). In [24, 25], FMTL and SGTs are used to monitor road traffic scenes, [26–29] apply them to human behavior understanding and surveillance, and [30] uses them for intelligent robot control. Preliminary work on the case study presented in this paper is included in [31]. The models we use for representation and reasoning are based on expert knowledge rather than learned from training data. Compared to other approaches, expert-knowledge-based representation and reasoning in FMTL and SGTs is intuitive, convenient, flexible, and easily controllable. The clear boundary between machine perception and reasoning makes it easier to improve one without the other. Furthermore, deductions are understandable by humans and completely provable, and existing rules can be adapted to new settings with relatively little effort. Especially the ability to understand the reasoning process is essential to the presented case study. FMTL/SGT expert systems are suitable for knowledge intensive problems with heterogeneous search spaces such as the one presented here.

3 Case Study Scenario

The presented case study is situated at the State Fire Service Institute (Institut der Feuerwehr) Nordrhein-Westfalen during one of their staff exercises for crisis response control room operations (Figure 2). The exercise is a six hour role playing effort where the participants take on the roles of a full control room staff and others stage the outside world; simulating field units, crisis dynamics, distress calls, and radio communications. The simulated crisis for this exercise was

a collision between a passenger train and a cargo train carrying hazardous material. The staff inside the control room is organized as follows. Each first officer is responsible for a functional area: unit management (S1), situation assessment (S2), strategy (S3), and supplies (S4). The first officers answer to the director of operations, and each first officer as well as the director of operations have one or two additional staff members answering to them. Furthermore, there is some supporting staff for maintaining displays (e.g. maps and unit tables), editing documents, and managing incoming and outgoing messages. Several instructors are offering assistance, the director of operations being one of them.

What follows is a description of the typical workflow in such control rooms, corresponding to the recorded data. Once the control room is fully occupied, the director of operations introduces the staff to the current crisis situation. Everybody stops working and returns to their seats to listen. After the introduction, the director of operations tells his staff to continue their preparations and asks his first officers to join him at the table at the central table for strategic planning. When this is done, the director of operations addresses the whole room, announcing that everybody must attend to their tasks until the next briefing. This is when their behavior becomes highly dynamic. Director of operations, first officers, their subordinates, and supporting staff scatter across the room, attending to their displays, documents, and messages. Groups are constantly forming and breaking, and there is a lot of discussion going on. In due time, the director of operations calls the next briefing and everybody returns to their seats. After an introduction by the director of operations, each of the first officers stands in front of the appropriate wall display to give a status report on their own functional area. Everybody listens quietly, except for the director of operations who is occasionally asking the presenter questions, sometimes involving one of the other first officers in the discussion. The director of operations concludes the briefing by summarizing the current action plan and everybody gets back to performing dynamic control room operations.

We aim to model and recognize the different types of person-person interaction and person-object interaction in various group formations. Besides dynamic behavior, we also aim to recognize the more structured events during briefings. The recorded data consists of five briefing / dynamic behavior cycles, each lasting around 70 minutes. The first cycle containing the described introductory phase was analyzed thoroughly and two four minute fragments and two ten minute fragments were selected for the annotation process described in Section 5.

4 Sensor Setup

The staff exercise was recorded with four normal cameras and one fisheye, providing complete and redundant coverage from various angles as shown in Figure 3. Four microphones were installed across the room to provide complete audio coverage. To make the data analysis and annotation easier, we used the raw video data to generate one synchronized five-pane image per second. One of them is shown in Figure 3. A sampling rate of *1fps* is sufficient, because no machine

Fig. 3. Five-pane image showing the cameras' viewing angles. To simplify the annotation process, such images are generated at *1fps* from the raw video data recorded at the State Fire Service Institute (Institut der Feuerwehr) Nordrhein-Westfalen.

perception is performed on the data, and because the events we aim to recognize do not have fast dynamics. Higher sampling rates could be obtained with corresponding annotation effort, or an interpolation algorithm applied to the $1fps$ numeric and textual data (world model). The audio data is used to better understand what is going on in the control room (context information), and to annotate the participants' speech activity.

5 Annotation Process

Two four minute fragments and two ten minute fragments from the first 70 minutes of the exercise were selected for annotation with hypothetical outputs from machine perception and ground-truth for corresponding semantic event descriptions. A PyQt annotation tool was specifically developed for this purpose. Its main component is an interactive birdseye view allowing the user to manipulate the modeled objects. The tool is used to create a birdseye view and underlying XML data (hypothetical machine perception and semantic ground-truth results) for each second of recorded data. This is exemplified by the screenshot in Figure 4, displaying the same data as the five-pane image in Figure 3.

Before the annotation process can begin, the user has to edit an XML stage file using a custom XML scheme, specifying which dynamic objects can be added to the scene; in this case people, notepads, and messages. The stage file specifies their static attributes: name, type, subtype, and size. The file also determines which semantic event types should be recognized, so that the corresponding ground-truth results can be annotated using the provided interaction elements. Event types are specified in terms of their names, arities (number of arguments), and argument domains (allowed object types). Finally, the stage file describes

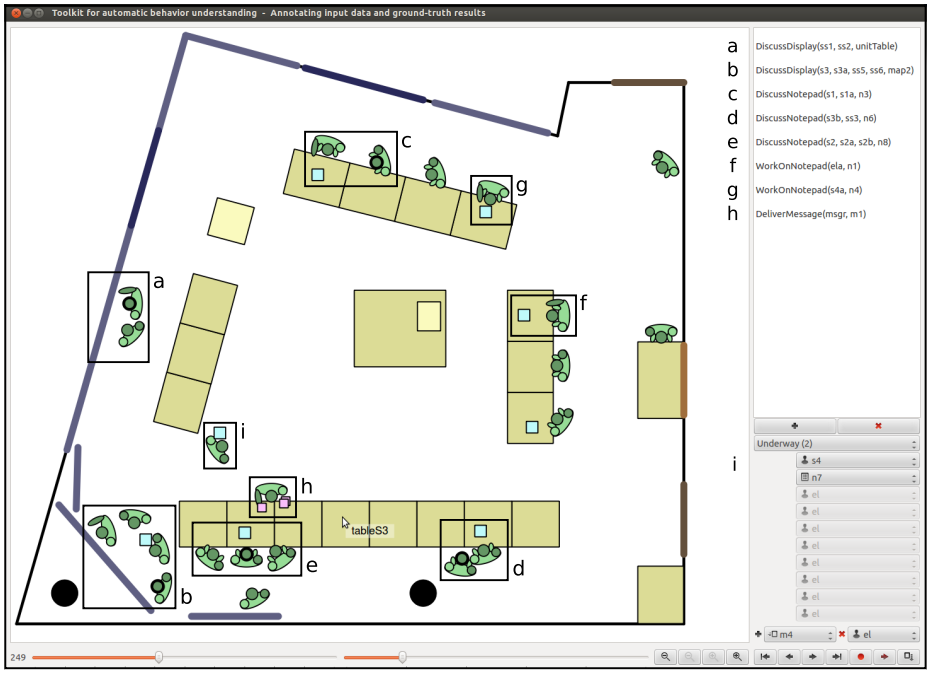

Fig. 4. Tool for annotating audiovisual data with hypothetical machine perception and semantic ground-truth. It provides an interactive birdseye view for manipulating modeled objects. The displayed data and ground-truth results correspond to Figure 3: *a)* two people discussing the field unit status table, *b)* similar *c)* two people discussing a notepad, *d,e)* similar, *f)* person working on a notepad, *g)* similar, *h)* delivering a message, and *i)* underway with notepad.

the static objects in the room in terms of name, type, subtype, location, and size. In this case, the static object types are: wall, table, display, door, hatch, and device. The static objects are visualized as in Figure 4. And the dynamic object specifications (in this case for people, notepads, and messages) and the ground-truth event types to choose from, are used to fill the corresponding interaction elements. Upon loading the stage file, no dynamic objects are present, they are added and removed through user interaction.

The attributes of the dynamic objects are manipulated using mouse interaction. Each person can be moved and rotated, and their body pose, gesture activity, and speech activity can be set. Speech is indicated by a rim around the head, speech-supporting gesticulation by a rim around the right hand. An extended and optionally rotated arm indicates pointing and interaction with displays, notepads, and messages, an extended head indicates looking down, and extended legs indicate sitting (see Figure 4). Notepads and messages can only be moved around. After the dynamic objects have been manipulated to reflect the audiovisual data, semantic event descriptions can be annotated by selecting

the required event-argument-combinations (see Figure 4, i). This process is repeated for each second of data, i.e. for each of the images exemplified by Figure 3. The interface includes elements for recording, playing back, and navigating through the data, and data files can be saved to be reloaded at a later time. The resulting XML data contains a description of the recorded dymanic objects and ground-truth events for each second. People, notepads, and messages possess the attributes name, type, subtype (strings), presence (boolean), x-coordinate, y-coordinate, width, and height (integers). In addition, people have orientation, gesture (integers), speech, looking down, and sitting (booleans). The recorded ground-truth events have their name and list of arguments specified in XML.

To further improve the annotation process, still images, video streams, and audio streams should be displayed in sync with the birdseye view visualizing the XML data. Furthermore, results and ground-truth (semantic event descriptions) should be visualized in the birdseye view, and ideally also in the audiovisual data. The bounding boxes in Figure 4 were added manually. Other ideas to improve the annotation tool include a more sophisticated set of data objects, 3D data functionality, and convenience/data-quality improvements through AI and physics laws that operate on the data model. The presented tool can also visualize real machine perception outputs, and it can be customized for any of the application domains described in Section 1.

6 Fuzzy Metric Temporal Logic (FMTL) and Situation Graph Trees (SGTs)

Once input data is available, the actual generation of semantic event descriptions can begin. The XML data from the annotation process and the handwritten XML stage file are fed into a reasoning engine based on fuzzy metric temporal logic (FMTL) and situation graph trees (SGTs) [24–31]. We use F-Limette: a reasoning engine for FMTL written in C, and the SGT-Editor: a Java application for editing and traversing SGTs. The FMTL language is a first order logic extended with fuzzy evaluation and temporal modality. Fuzzy evaluation allows for reasoning about inherently vague concepts such as distance categories (e.g. close, far) as well as reasoning about uncertainty in the input data. The latter will be addressed in Section 8. Temporal modality allows for reasoning about temporal developments using rule conditions grounded in points along the time axis corresponding to past, current, and future states of the world.

Each reasoning process starts at the root node of an SGT, which is then traversed as described in [28]. From each traversed node, FMTL rule conclusions are queried that initiate Prolog-like rule execution processes (i.e. F-Limette uses the logic programming paradigm). Each rule execution process returns a truth value between *0.0* and *1.0* depending on the rule conditions that were directly or indirectly evaluated after querying the rule conclusion in the SGT node, and ultimately on the atomic facts from the input data. The returned truth values are carried down to the next SGT node where they are used as base truth value

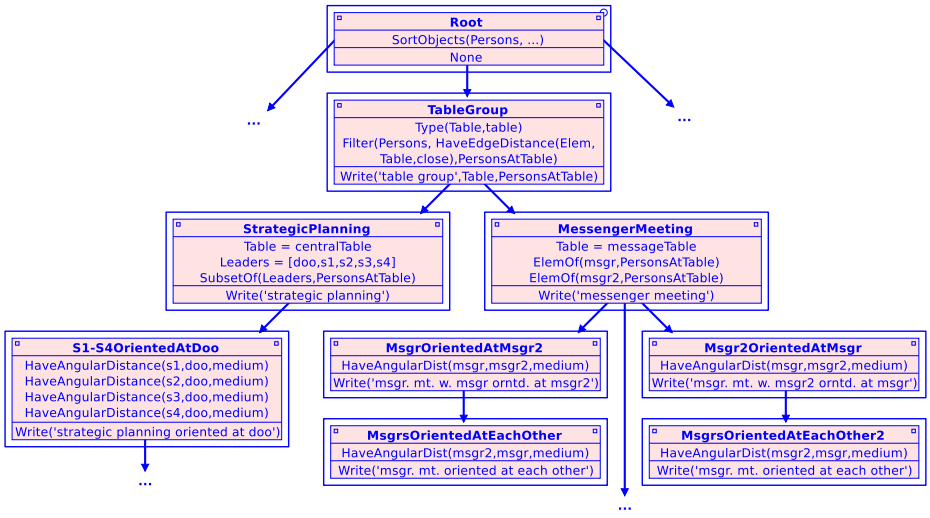

Fig. 5. Part of a situation graph tree (SGT) from the presented case study. It is used to detect groups around tables and conceptual refinements thereof.

(instead of *1.0*). Semantic event descriptions with corresponding truth values but also actuator commands can be generated from any SGT node.

SGTs are hypergraphs consisting of situation graphs (see Figure 5). Each situation graph contains one or more situation schemes, and each situation scheme possesses a name, one or more preconditions, and zero or more postconditions (i.e. semantic event descriptions and/or actuator commands). To model temporal dynamics and and events consisting of multiple phases, situation schemes can be interconnected through temporal edges within each situation graph. This feature is not used in Figure 5, as its situation graphs (visualized by thick boxes) contain only one situation scheme each. Its only temporal edge is the reflexive one on the *Root* situation scheme, causing the reasoning process to continue over time. Conceptual refinement is visualized by a thick edge between a situation scheme and a situation graph below it. FMTL rules are largely domain independent and typically about spatiotemporal relations, whereas SGTs are more domain specific as they usually constitute abstract relations between the FMTL rules they deploy. Once an FMTL rule base has been established it stays relatively fixed and it can be used by different SGTs within the same application domain or across different application domains. We now provide a detailed description of some of the formal knowledge that was developed for the presented case study. Figure 5 depicts an example SGT, Equations 1–10 and Figure 6 show some of the applied FMTL rules, and Table 1 lists the available atomic fact types.

$$EdgeDistanceIs(p,q,\delta) \wedge AssociateEdgeDistance(\delta, category) \rightarrow \textbf{\textit{HaveEdgeDistance}}(p,q,category) \quad (1)$$

$$Position(p,x_p,y_p) \wedge Position(q,x_q,y_q) \wedge Size(q,w_q,h_q) \wedge Orientation(q,\theta_q) \wedge DistancePointToPlane(x_p,y_p,x_q,y_q,w_q,h_q,\theta_q,\delta) \rightarrow \textbf{\textit{EdgeDistanceIs}}(p,q,\delta) \quad (2)$$

$$AngularDistanceIs(p,q,\delta) \wedge AssociateAngularDistance(\delta, category) \rightarrow \textbf{\textit{HaveAngularDistance}}(p,q,category) \quad (3)$$

$$Position(p,x_p,y_p) \wedge Orientation(p,\theta_p) \wedge Position(q,x_q,y_q) \wedge Angle(x_p,y_p,\theta_p,x_q,y_q,\delta) \rightarrow \textbf{\textit{AngularDistanceIs}}(p,q,\delta) \quad (4)$$

$$Position(p,x_p,y_p) \wedge \ AbsoluteArmAngle(p,\theta_{arm_{abs}}) \wedge Position(q,x_q,y_q) \wedge Angle(x_p,y_p,\theta_{arm_{abs}},x_q,y_q,\delta) \wedge AssociateAngularDistance(\delta, \textbf{\textit{close}}) \rightarrow \textbf{\textit{ExtendingArmToward}}(p,q) \quad (5)$$

$$Orientation(p,\theta_p) \wedge ExtendingArm(p,\theta_{arm}) \wedge AngularSum(\theta_p,\theta_{arm},\theta_{arm_{abs}}) \rightarrow \textbf{\textit{AbsoluteArmAngle}}(p,\theta_{arm_{abs}}) \quad (6)$$

$$\delta_y = y_q - y_p \wedge \delta_x = x_q - x_p \wedge Atan2(\delta_y,\delta_x,\theta_{yx}) \wedge AngularDifference(\theta_p,\theta_{yx},\delta) \rightarrow \textbf{\textit{Angle}}(x_p,y_p,\theta_p,x_q,y_q,\delta) \quad (7)$$

$$Speaking(p) \vee Gesticulating(p) \vee ExtendingArm(p,\theta) \rightarrow \textbf{\textit{Interacting}}(p) \quad (8)$$

$$\Diamond_{-1} Interacting(p) \vee Interacting(p) \vee \Diamond_1 Interacting(p) \rightarrow \textbf{\textit{InteractingInInterval}}(p) \quad (9)$$

$$\Diamond_{-1} InteractingInInterval(p) \wedge \Diamond_1 InteractingInInterval(q) \rightarrow \textbf{\textit{InteractingTogether}}(p,q) \quad (10)$$

Root in Figure 5 sorts the modeled objects into lists according to arbitrary FMTL sort criteria, in this case objects *p* with *Type(p, person)*. The situation scheme *TableGroup* selects objects *Table* with *Type(Table, table)*. For each of them, the list containing all persons is filtered into a list containing only persons that are close to that table. *HaveEdgeDistance(Elem, Table, close)* calculates the distance between a person's center and an object's closest edge and then associates this distance with fuzzy categories (see Equations 1 and 2 and Figure 6, left). *Filter(InputList, RuleToApply(Elem, ...), OutputList)* applies an arbitrary rule (in this case *HaveEdgeDistance(...)*) to each element *Elem* in *InputList* and adds each *Elem* with truth value *V[RuleToApply(Elem, ...)] > 0* to *OutputList*. *V[Filter(InputList, RuleToApply(Elem, ...), OutputList)]* is the average over all *V[RuleToApply(Elem, ...)]*.

The situation scheme *TableGroup* is refined into *StrategicPlanning* if *Table = centralTable* and the director of operations (*doo*) and S1 through S4 are close (determined by fuzzy evaluation in *HaveEdgeDistance(...)*). This can be further refined into *S1-S4OrientedAtDoo* if *HaveAngularDistance(sX, doo, medium)* applies to S1 through S4, i.e. if they have the director of operations in their fuzzy fields of vision (see Equations 3, 4, and 7, and Figure 6, right). The other

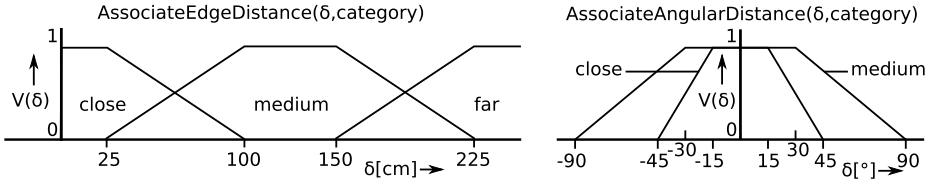

Fig. 6. Visualization of FMTL rules associating distances to distance categories

Table 1. Atomic facts from the input data (static and dynamic object attributes)

$Present(p)$	$Orientation(p, \theta_1)$	$Speaking(p)$	$Sitting(p)$
$Position(p, x, y)$	$Type(p, \tau_1)$	$Gesticulating(p)$	$LookingDown(p)$
$Size(p, w, h)$	$Subtype(p, \tau_2)$	$ExtendingArm(p, \theta_2)$	$OwnerOf(p, q)$

side of Figure 5 shows how *MessengerMeeting* and its refinements can be deduced once *TableGroup* has been established. *Table* needs to be instantiated as *messageTable* and the staff handling incoming and outgoing messages (*msgr* and *msgr2*) need to be close (*HaveEdgeDistance(...)*). Then, *HaveAngularDistance({msgr, msgr2}, {msgr, msgr2}, medium)* is used to deduce whether they are oriented at each other.

Figure 5 depicts one branch from the current SGTs. Other branches recognize events centered around persons, notepads, messages, and displays. Furthermore, all branches can contain further conceptual refinements for describing interaction patterns. Equations 5–7 and Figure 6 (right) for example can be used for display centric events, calculating a fuzzy truth value for *ExtendingArmToward(person, display)*. Equations 8–10 can be used in a conceptual refinement for Figure 5. Here, interactivity is checked at the previous, current, and next frame using $\Diamond_{-1}$ and $\Diamond_1$. And if two persons in a group are interacting simultaneously or in short succession, they are probably interacting together, provided that they are facing the same object or facing each other.

7 Results

Figure 7 shows some experimental results generated by the SGT in Figure 5 (black lines displaying truth values V as a function of time t in s), and the corresponding ground-truth that was annotated using the tool depicted in Figure 4 (black dots). The top-left graph shows that the system correctly recognizes when the director of operations and his first officers are gathering around the central table. The bottom-left graph shows that it also succeeds in detecting that the first officers are oriented at the director of operations. The system correctly drops this deduction when the director of operations is referring to a display and the first officers turn to look at it. The top-right graph shows the successful recognition of the supporting staff in charge of message handling (*msgr1* and *msgr2*) meeting at the message table, and *msgr2* briefly stepping away from

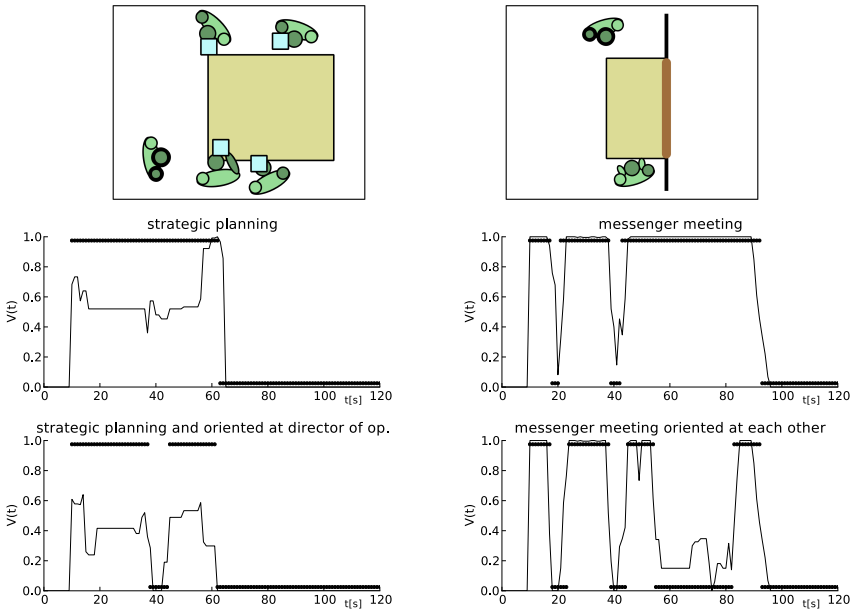

Fig. 7. Experimental results generated by the SGT in Figure 5

it twice. And the bottom-right graph shows that the system correctly classifies their orientations.

Table 2 provides an overview of the events that are recognized by the current system (top) and the ones that are still under development (bottom). Note that this list is by no means final and that each event can have multiple refinements where the staff members' roles and object names are taken into account for example. The presented results were obtained in real time, but as the number of involved objects, the predicates' arities, and the complexity of the FMTL rules and SGTs increase, runtime needs to be improved by applying better parallellization, more computer resources, and heuristics about which objects to consider. At the time of writing, a quantitative evaluation was not yet possible. We are currently performing one to evaluate the presented system.

8 Handling Imperfect Input Data

Algorithms for automatic behavior understanding must be able to handle gaps and uncertainty in their input data. Incomplete data handling is important because of possible occlusions in the sensor data, areas without sensor coverage, and technical problems with machine perception components. High-level events can not be detected if some of their rule conditions are not fulfilled due to missing data. Uncertainty handling is important because machine perception components often provide confidence values that should be incorporated into

Table 2. Events currently recognized (top) and still under development (bottom)

Person alone
Group around {person, notepad, message, table}
Person {joining, leaving} group
Person underway
Person underway with {notepad, message}
Group underway (similar speeds)
Group or person {observing, editing, discussing} {notepad, message} (uses $LookingDown(p)$)
Group or person {observing, editing, discussing} display
Person {talking, listening} to someone
Everybody at own seat (uses $Sitting(p)$ and $OwnerOf(p,q)$, truth value = $\frac{sitting}{all}$)
Briefing phases: introduction, S1, S2, S3, S4, conclusion
Discussions during briefing
Message handling and its phases
Fetching someone to join a group

the reasoning process so that uncertainty in perception outputs is reflected in the high-level results as well.

Related problems include various types of noise in the input data as well as wrong data, typically in the form of outliers. Our approach can inherently handle noisy data through FMTL rules applying temporal filtering and fuzzy evaluation. Such rules are also helpful against outliers. Additionaly, outlier detection can be applied to the input data during preprocessing. Outliers could also be detected by the reasoning process itself, using rules about the data's expected dynamics, potentially even providing machine perception with top-down knowledge about this to improve its outputs or guide sensor and resource deployment.

In logic reasoning, the effects of incomplete data can be countered to a certain degree using abduction, where intermediate conditions that can not be deduced are hallucinated instead so that reasoning can continue and certain events can be detected despite their missing conditions. Furthermore, interpolation can be applied to incomplete input data to counter such effects early in the processing chain. It can be applied as preprocessing, independently of the chosen high-level methods. This effectively turns the missing data problem into an uncertainty problem, because interpolated data should have appropriate confidence values associated with them that depend on the confidence in the surrounding data used for interpolation as well as on the temporal distances between new data points and the ones they were calculated from. Confidence should increase towards an interpolated gap's edge, and large gaps should cause ever lower confidence values as you move to the center. In [27], we describe how to apply abduction and interpolation to the FMTL/SGT framework.

Uncertainty in the input data (from interpolation or other causes) can be handled in the FMTL/SGT framework as follows. Each perception output i can have a confidence value $P[i]$ between 0.0 an 1.0. Let a and b be two points on a plane with confidence values $P[a]$ and $P[b]$. The Euclidian distance between a and b is calculated by an appropriate FMTL rule as $\delta_{ab} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$. Confidence values are usually combined through multiplication and δ_{ab} depends on a and b, so $P[AssociateDistance(\delta_{pq}, category)] = P[\delta_{pq}] = P[a]P[b]$. Vague truth values for $V[AssociateDistance(\delta_{pq}, category)]$ are calculated from δ_{pq} as

in Figure 6, regardless of these confidence values. This means that uncertainty and vagueness are represented separately. Using appropriate FMTL conjunction semantics, each $P(f)$ and $V(f)$ can be condensed into $V'(f)$, a truth value reflecting both uncertainty and vagueness.

9 Conclusion

The presented toolkit for automatic behavior understanding generates semantic event descriptions from machine perception using fuzzy metric temporal logic (FMTL) and situation graph trees (SGTs). It was applied to a case study on automatic behavior report generation for training purposes in crisis response control rooms. Instead of machine perception and multimodal fusion we used a newly developed annotation tool to provide the reasoning engine with input. The paper contains several novel contributions: a new dataset, a new tool for data analysis and annotation, a unique case study with a large amount of perception modalities and objects, a newly developed FMTL/SGT knowledge base for this case study (also applicable to other domains), corresponding experimental results, and an explanation on how to handle imperfect input data.

This forms the basis for our future work. First and foremost, exhaustive quantitative evaluations will be performed on the case study data, comparing the results to ground-truth in a precision/recall-fashion. Our annotation tool currently generates binary ground-truth, but we would like to expand this to n-valued or fuzzy ground-truth because of the fuzzy nature of the results. Second, we will keep improving the FMTL rules and SGTs to recognize more sophisticated events from the presented case study data, exploiting the full power of fuzzy evaluation and temporal modality. Third, such experiments will be performed on various types of imperfect input data as described in Section 8 to evaluate the system's robustness. Fourth, the annotation tool shall be developed further as described at the bottom of Section 5.

We are also starting to involve end-users, human science experts, and software developers. We currently focus on the physical attributes of the people and objects in the room (hypothetical machine perception outputs), but the system can be improved by taking into account more domain specific attributes, i.e. context information (unit status, crisis dynamics, staff roles, and more object information). This would allow us to model more sophisticated expert knowledge in FMTL and SGTs in order to deduce a richer set of semantic event descriptions that is of greater use to potential end-users. To achieve this, we plan to organize a seminar with participants from the State Fire Service Institute (Institut der Feuerwehr) Nordrhein-Westfalen and participants from the research group that was involved in the data recording. In addition to the audiovisual data, they gathered and analysed the messages, documents, and context-related developments of the staff exercise. We are also investigating the alternative application domains listed in Section 1. Our research is situated in an environment that focuses on computer vision and other forms of machine perception, which facilitates the progress toward an online system. The ultimate goal is an unsuper-

vised real-time system containing multiple machine perception components and multimodal fusion instead of manual annotation, with embodiment and action generation, and synchronous visualization of sensor data, machine perception, and semantic event descriptions.

Acknowledgements. This work is supported by the Fraunhofer-Gesellschaft Internal Programs under Grant 692 026. We thank the State Fire Service Institute (Institut der Feuerwehr) Nordrhein-Westfalen for providing the opportunity to record a staff exercise and for their valuable expert knowledge and feedback.

References

1. Aggarwal, J.K., Ryoo, M.S.: Human Activity Analysis: A Review. Computing Surveys, ACM. 43(3), 16:1–16:43 (2011)
2. Ye, J., Dobson, S., McKeever, S.: Situation Identification Techniques in Pervasive Computing: A Review. Pervasive and Mobile Computing, Elsevier. 8(1), 36–66 (2011)
3. Turaga, P., Chellappa, R., Subrahmanian, V., Udrea, O.: Machine Recognition of Human Activities: A Survey. Circuits and Systems for Video Technology, IEEE Transactions on. 18(11), 1473–1488 (2008)
4. Kosmopoulos, D.I., Doulamis, N.D., Voulodimos, A.S.: Bayesian Filter Based Behavior Recognition in Workflows Allowing for User Feedback. Computer Vision and Image Understanding, Elsevier. 116(3), 422–434 (2012)
5. Fischer, Y., Beyerer, J.: Defining Dynamic Bayesian Networks for Probabilistic Situation Assessment. Information Fusion, Internat. Conference on (FUSION). (2012)
6. Shi, Y., Huang, Y., Minnen, D., Bobick, A., Essa, I.: Propagation Networks for Recognition of Partially Ordered Sequential Action. Computer Vision and Pattern Recognition, IEEE Conference on (CVPR). 862–869 (2004)
7. Aloimonos, Y., Guerra-Filho, G., Ogale, A. The Language of Action: A New Tool for Human-Centric Interfaces. In: Human Centric Interfaces for Ambient Intelligence. 95–131, Elsevier (2009)
8. Kitani, K.M., Sato, Y., Sugimoto, A.: Recovering the Basic Structure of Human Activities from Noisy Video-Based Symbol Strings. Pattern Recognition and Artificial Intelligence, International Journal of, World Scientific. 22, 1621–1646 (2008)
9. Ivanov, Y., Bobick, A.: Recognition of Visual Activities and Interactions by Stochastic Parsing. Pattern Analysis and Machine Intell., IEEE Tr. on. 22(8), 852–872 (2000)
10. Van Kasteren, T.L.M., Englebienne, G., Kröse, B.J.A.: Hierarchical Activity Recognition Using Automatically Clustered Actions. Ambient Intelligence, International Conference on (AmI). 82–91 (2011)
11. Filippaki, C., Antoniou, G., Tsamardinos, I.: Using Constraint Optimization for Conflict Resolution and Detail Control in Activity Recognition. Ambient Intelligence, International Conference on (AmI). 51–60 (2011)
12. Yao, B.Z., Yang, X., Lin, L., Lee, M.W., Zhu, S.C.: I2T: Image Parsing to Text Description. Proceedings of the IEEE. 98(8), 1485–1508 (2010)
13. Ryoo, M., Aggarwal, J.: Semantic Representation and Recognition of Continued and Recursive Human Activities. Computer Vision, International Journal of, Springer. 82, 1–24 (2009)
14. Gupta, A., Srinivasan, P., Shi, J., Davis, L.: Understanding Videos, Constructing Plots, Learning a Visually Grounded Storyline Model from Annotated Videos. Computer Vision and Pattern Recognition, IEEE Conf. on (CVPR). 2012–2019 (2009)

15. Sadilek, A., Kautz, H.: Location-Based Reasoning about Complex Multi-Agent Behavior. Artificial Intelligence Research, Journal of, AAAI Press. 43, 87–133 (2012)
16. Morariu, V., Davis, L.: Multi-Agent Event Recognition in Structured Scenarios. Computer Vision and Pattern Recognition, IEEE Con. on (CVPR). 3289–3296 (2011)
17. Kembhavi, A., Yeh, T., Davis, L.: Why Did the Person Cross the Road (There)? Scene Understanding Using Probabilistic Logic Models and Common Sense Reasoning. Computer Vision, European Conference on (ECCV). Part II, 693–706 (2010)
18. Bohlken, W., Neumann, B.: Generation of Rules from Ontologies for High-Level Scene Interpretation. Rule Interchange and Applications, International Symposium on (RuleML). 93–107 (2009)
19. Augusto, J.C., Nugent, C.D. (editors): Designing Smart Homes, The Role of Artificial Intelligence. Springer (2006)
20. Gottfried, B., Aghajan, H. (editors): Behaviour Monitoring and Interpretation, Smart Environments. IOS Press (2009)
21. Gong, S., Xiang, T.: Visual Analysis of Behaviour, From Pixels to Semantics. Springer (2011)
22. Ley, B., Pipek, V., Reuter, C., Wiedenhoefer, T.: Supporting Improvisation Work in Inter-Organizational Crisis Management. Human Factors in Computing Systems, ACM Annual Conference on (CHI). 1529–1538 (2012)
23. Toups, Z.O., Kerne, A., Hamilton, W.A., Shahzad, N.: Zero-Fidelity Simulation of Fire Emergency Response: Improving Team Coordination Learning. Human Factors in Computing Systems, ACM Annual Conference on (CHI). 1959–1968 (2011)
24. Nagel, H.H.: Steps Toward a Cognitive Vision System. AI Magazine, AAAI Press. 25(2), 31–50 (2004)
25. Arens, M., Gerber, R., Nagel, H.H.: Conceptual Representations between Video Signals and Natural Language Descriptions. Image and Vision Computing, Elsevier. 26(1), 53–66 (2008)
26. Bellotto, N., Benfold, B., Harland, H., Nagel, H.H., Pirlo, N., Reid, I., Sommerlade, E., Zhao, C.: Cognitive Visual Tracking and Camera Control. Computer Vision and Image Understanding, Elsevier. 116(3), 457–471, Special Issue on Semantic Understanding of Human Behaviors in Image Sequences (2012)
27. Münch, D., IJsselmuiden, J., Grosselfinger, A.K., Arens, M., Stiefelhagen, R.: Rule-Based High-Level Situation Recognition from Incomplete Tracking Data. Rules, Research Based and Industry Focused, International Symposium on (RuleML). (2012)
28. Münch, D., Jüngling, K., Arens, M.: Towards a Multi-Purpose Monocular Vision-based High-Level Situation Awareness System. International Workshop on Behaviour Analysis and Video Understanding @ ICVS. (2011)
29. González, J., Rowe, D., Varona, J., Roca, F.X.: Understanding Dynamic Scenes Based on Human Sequence Evaluation. Image and Vision Computing, Elsevier. 27(10), 1433–1444, Special Sec. on Comp. Vis. Meth. for Ambient Intelligence (2009)
30. Bellotto, N.: Robot Control Based on Qualitative Representation of Human Trajectories. Designing Intelligent Robots: Reintegrating AI, AAAI Symposium on. (2012)
31. Münch, D., IJsselmuiden, J., Arens, M., Stiefelhagen, R.: High-Level Situation Recognition Using Fuzzy Metric Temporal Logic, Case Studies in Surveillance and Smart Environments. Analysis and Retrieval of Tracked Events and Motion in Imagery Streams, 2nd IEEE Workshop on (ARTEMIS) @ ICCV. (2011)