FP6-IST-002020

COGNIRON

The Cognitive Robot Companion

Integrated Project

Information Society Technologies Priority

D5.4.1 Report on Requirements for a Modelling System Dealing with Temporal Processes

Due date of deliverable: 31/03/2004
Actual submission date: 31/01/2005

Start date of project: January 1st, 2004 **Duration** : 48 months

Lead contractor for this deliverable: LAAS-CNRS

Revision: final
Dissemination level: PU

Executive Summary

Time is a fundamental component of the knowledge we manipulate, and use. We not only care about objects and artifacts, their position, their shape, but also when were they at this location or this other one, how old are they, for how long have they been in a particular state, etc. Similarly, when one has to analyze a dynamic process (fixing dinner, walking the dog, etc), one must take time into account. For example, if one put some water to boil, one knows that after a more or less predictable duration (related to the size of the pane, and the quantity of water), it should be boiling. A cognitive robot will also need such capabilities to be able to fully understand, interact with humans and other robots and act on the environment.
In this document we study and analyze what are the time representations one can use to endow our cognitive robot with such capabilities. By time representation, we do not only look at how should information be "tagged" with time, but also what are the temporally related knowledge to use for activity such as situation recognition, or action recognition.
At this stage, we mostly present here a state of the art, and point out what are the approaches which seems to be the most promising from our point of view.

Role of Temporal Relation Modelling in Cogniron

In order to address fundamental questions of scene understanding we not only have to include object recognition and the spatial relationships between objects but also their temporal properties. For example: if the robot removes an object at a certain location at some time, this object will not be at that location after that time. Furthermore, time can compensate for lost metric information due to an abstraction process, e.g. the travel time between two topological nodes can be very useful for localization. To model these temporal relations in a formal way various formalism have been considered. In real world situations the information about the event (the action and consequences) will be uncertain and subject to noise.
In this workpackage we study and develop models that are well suited to deal with temporal relations. We have experience with various temporal models (some dealing with uncertainties) and we will consider if they are well suited in the context of the home tour scenario.

Relation to the Key Experiments

Making use of a temporal model and a situation recognition system using a temporal chronicle would certainly improve the various Key Experiments. As we will see in the following sections, there exists a number of systems which could be used for that purpose. Some of these systems are available and use techniques which have already been demonstrated on large scale applications (e.g. situation recognition with temporal chronicles). Other techniques are somewhat less mature or more complex to setup or run on a real system, as a result, we need to further evaluate their adequacy for the COGNIRON Key Experiments.

1 Models of Temporal Relations

A congitive robot has to deal with geometrical and topological information about its environment (here we consider the environment to include other robots and humans present or known to the robot). Nevertheless, such an information is usually no sufficient to be able to get the full understanding of the current situation, and of its evolution. Indeed, events and sensory information often need to be temporally related to be fully correlated. Observing A then B may carry a completely different meaning than observing B then A... (e.g. if a human while giving an order to move an object point to a location X and then to another one Y, it may mean than it is instructing the robot to take the object from X to Y, and not from Y to X). Similarly, observing A and then B in a "given" time frame or time window may imply a particular situation while the meaning could be quite different if the window is smaller or larger. Thus temporal relations about objects and agents is probably as important than the geometrical (location or position) information. Last, such observation, or analysis is bound to have some uncertainty, either on the exact date of occurrence of a particular events (e.g. due to the sensor uncertainty, or to communication delay), or on on the models used to process these informations (e.g. we may have models that recongize a particular situation with a given probability).
So the designer of a cognitive robot is faced with a number of problems, with respect to temporal representations and models.

- What kind of representation to use to "tag" or "stamp" events and data?
- What should be the horizon over which a particular information (and its temporal tag) should be kept (in case it needs to be referenced later on for situation recognition). Should the information be explicitly stored, or "remembered" in a hypothesis tree?
- How do we acquire the temporal models which are then used by the system?
- What type of processing and reasoning one want to implement: situation recognition, situation forecasting, etc?
- What type of uncertainty management will one need for the temporal information (on the data itself, and/or on the models)?

The issue of information storage is one which need to be adressed. For example, should a robot store all the subsequent positions of a human present in the scene for later use? Obviously, if this is done for every piece of information gathered by the robot, a large chunk of the memory will have to be devoted to this function. However, there are also recognition mechanism which only hold the information needed to confirm or infirm the current hypothesis.
The temporal representation of information is also critical. For events, usually the date of occurrence is enough. But for temporal "relations" various format can be considered, symbolic or numerical, discrete or continuous, exact or interval.
Various temporal models can be considered and are presented in our state of the art: temporal chronicles, hidden Markov models and dynamic Bayesian networks. Some are more adapted to handling uncertainties, and provide some interesting "the model can be learned" capabilities, while others are more dedicated to a more explicit (i.e. quantitative) temporal information handling.

2 State of the Art

If one consider the problem of modeling temporal relations to recognize situations, there is a large body of work on this subject which has been studied in the diagnostic field, to recognize alarms, intrusions or in general events resulting from a temporally related events [3]. We are not quite interested in recognizing alarms or problems, still these techniques are not limited to the diagnostic field and appear to be somewhat applicable to our problem. After all, monitoring a situation or a process is independent of the type of processing one wants to perform afterward (it can be diagnostic, or robot human interaction). In any case, this field has come up with an interesting method to represent temporal information, and to recognize situations.

2.1 Temporal Chronicles

IxTeT-Reco is a temporal chronicle recognition system. A chronicle model is a description of generic scenario (normal or abnormal evolution) to be surveyed [11, 16]. It is represented as a set of events and a set of temporal constraints, between these events and with respect to the context. A chronicle model may also specify events to be generated and actions to be triggered as a result of the chronicle occurrence. Such an action or event could be passed to another component of the architecture in charge of processing/executing it. Deduced events can also in turn be taken as input by other chronicles, hence enabling a recursive chaining.
On-line, the chronicle recognition system receives as input a stream of time-stamped instanciated events. A sensory stimulus, once detected by a signal processing tool, may become an event. If some observed events match the model events of a chronicle, and if their occurrence dates meet the specified constraints, then an instance of this chronicle occurs. Its recognition may generate in turn new events, it can permit the focus of attention enabling the detection of forthcoming events, or it can trigger alarms, messages, data logging or other actions. This recognition system recognizes all instances of occurring chronicles, on the fly, as they develop. It does not account for all observed events, but only for those matching available models. The system generates as output deduced events and triggered actions associated to recognized chronicle models. It is mainly a temporal reasoning system. It is predictive since it predicts forthcoming events that are relevant to partial instances of chronicles currently hypothesized as taking place; it focuses on them and it maintains their temporal windows of relevance.
As a result, such approach focus on the recognition process, and does not “store” explicitly any data for later retrieval nor analysis. Of course, this is done at a price (maintaining all the possible hypothesis under consideration). The advantage of such approach over a basic temporal pattern matching of an event database obviously depend of the number of events, the number of chronicles and the horizon over which each of them may perform a full recognition. Note that the real-time properties of the recognition are well defined and predictable.
IxTeT-reco has been extended in [9] to deal with optional events and also to allow different politics of chronicle recognition (greedy or lazy).
The CRS [26] system, which was developed subsequently to IxTeT-reco is very similar.

2.2 Temporal Chronicles with Uncertainties

The WITAS project [17] has a chronicle recognition system based on IxTeT-reco. Still, in [9] the author introduces an extension of this system to handle uncertainties, which he developed while working on the WITAS project during a visit at Linköping University.

The extension itself was motivated by two problems:

- The confidence one has in a particular sensor data can vary (the quality of the images, the accuracy of the sensor itself, etc).
- The processing one apply to these data can itself lead to deduction with various confidence.

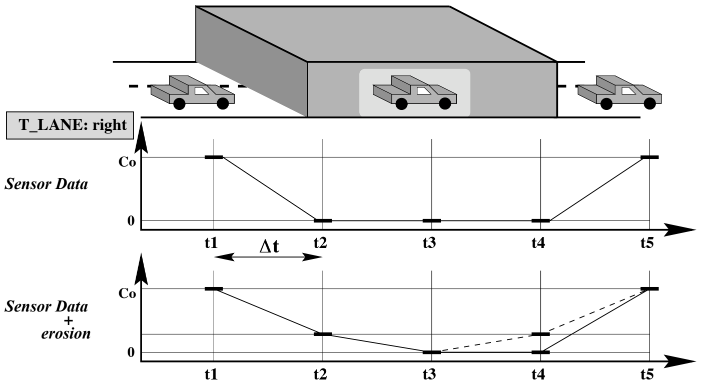

Figure 1: Example of disappearing data (from [9]).

Thus, the author decided to relax one of IxTeT-reco hypothesis (the one which assume that all sensory data are "certain" and always present). He studied various formalism to handle this (Bayesian Network, Belief Theory and Possibility Theory).
The resulting system was applied to situation recognition such as recognizing the maneuver of a passing car while it was temporary occulted (by a bridge or an overpass). An interesting property of the system is that most of the confidence computation can be done off line while compiling the recognition model, thus preserving the real-time properties of the recognition itself.

2.3 Hidden Markov Models and Dynamic Bayesian Networks

In many real life situations, we are faced with the task of inducing the dynamics of a complex stochastic process from limited observations about its state over time. Until now Hidden Markov Models (HMMs)[23] have played the largest role as a representation for learning models of stochastic processes. Recently, however, there has been increasing use of more structured models of stochastic processes, such as factorial HMMs [15] or Dynamic Bayesian Networks (DBNs)[5]. Such structured decomposed representations allow complex processes over a large number of states to be encoded using a much smaller number of parameters, thereby allowing better generalization from limited data [15, 14, 28]. Furthermore, the natural structure of such processes makes it easier for a human expert to incorporate prior knowledge about the domain structure into the model, thereby improving its inductive bias.
Typically, consider an intelligent agent whose task is to monitor a complex dynamic system such as the movements of humans in an environment. Tracking the state of such systems is a difficult task: their dynamics are noisy and unpredictable, and their state is only partially observable. Stochastic processes provide a coherent framework for modelling such systems. In many cases, the state of the system is prepresented using a set of state variables, where individual states are assignments of values to these variables. DBNs allow complex systems to be represented compactly by exploiting the fact

```
chronicle Overtake () () {
 timepoint before, t0, t1, t2, after;
 variable ?v, ?dir;
 ?v in VEHICLES;
 ?dir in DIRECTIONS;

 hold (T_DIRECTION() : ?dir, (before, t0));
 hold (DIRECTION(?v) : ?dir, (before, t0));
 hold (POSITION(?v, ?dir) : in_front, (before, t0));
 hold (T_LANE() : right, (before, t0));
 hold (LANE() : right, (before, t0));

 optional event (T_LIGHT() : (off, right), t0);
 event (T_LANE() : (right, left), t1);

 hold (POSITION(?v, ?dir) : behind, (t2, after));

 t0 $>$ before;
 (t1 - t0) in [MIN_CHANGE_LANE, MAX_CHANGE_LANE];
 (t2 - t1) in [MIN_OVERTAKE, MAX_OVERTAKE];
 after $>$ t2;

 when recognized { call report_overtake(?v, after);}
 }
```

Figure 2: Passing scenario

that each variable typically interacts only with few others. Learning a DBN from observations is based on the well-known Expectation-Maximization algorithm [8, 24], extended recently to learn the structure of the network itself [2, 13]. Once the model is learned, efficient approximate inference is done with the particle filtering family of algorithms, as in [18].
Still, HMM and DBN have some difficulties to represent time "values". For example, one need to fold in the model, time related properties (such as the speed, or the time since a particular event) to properly handle recognition of situations with a variable possible state recognition depending on the elapsed time between the occurence of two similar events.

2.4 Other Approaches

Several areas of research investigate problems close to ours. In plan recognition for example, one is interested in recognizing that a sequence of observed actions matches a previously described plan. In that area, a top-down approach guided by the plan to be recognized, is proposed in [25]; in addition to several restrictive assumptions (mainly the synchronization between observations and action occurrences) this work, as most others in plan recognition, does not take into account explicit temporal and real-time constraints.
Time is explicit in the event calculus of Kowalski and Sergot. This representation, mainly developed as a question-answering system for temporal data-bases [19], has been extended for planning or scheduling by several authors (e.g. [1]), does not address the recognition issue, while [7] does). A variant of it based on the interval calculus [20] is closer to our problem, but with a more restricted representation and without algorithmic concerns. In diagnosis, several authors consider dynamic sys-

tems; among which [4] and [22] explore available knowledge on temporal constraints between states (called modes) mainly as a way to focus an abductive diagnosis. On-line recognition is not their main focus, they do not provide complexity analysis, nor performance results.
The approach presented in [21] use finite state automaton to perform recognition, but has some limit to properly handle quantitative temporal information.
In [27], the authors a scenario recognition system for Video Interpretation. The temporal aspect of the recognition is inspired by the one presented in [11, 16], but applied to information extracted from a video stream (See Figure 3 for an example).

```
Scenario(Attack,
    Characters((cashier : Person), (robber : Person))
    SubScenarios(
      (cas_at_pos, inside_zone, cashier, "Back_Counter")
      (rob_enters,changes_zone,robber, "Entrance_zone","Infront_Counter")
      (cas_at_safe, inside_zone, cashier, "Safe")
      (rob_at_safe, inside_zone, robber, "Safe") )
    Constraints((rob_enters during cas_at_pos)
         (rob_enters before cas_at_safe)
         (cas_at_pos before cas_at_safe)
         (rob_enters before rob_at_safe)
         (rob_at_safe during cas_at_safe) ) )
```

Figure 3: Example of a bank attack sceanrio (from [27])

3 Proposed approach

The temporal chronicle representations as well as the situation recognition algorithm seems to be the most appropriate approach to the type of problems we want to address in COGNIRON. In this section, we further present this approach and the mechanism used to efficiently recognize a situation.

3.1 Knowledge Representation For Chronicles

The knowledge representation relies on temporally qualified domain attributes. One consider time as linearly ordered discrete set of instants, whose resolution is sufficient for the dynamics of the environment. One can handle the usual symbolic constraints of the time-point algebra (i.e. before, equal, after and their disjunctions), as well as numerical constraints. The later are expressed as pairs of real numbers $(e2 - e1) = [I-, I+]$ corresponding to lower bounds and upper bounds on the temporal distance between two points $e1$ and $e2$. For complexity reasons, one do not allow disjunctions of numerical constraints, since the constraint satisfaction problem becomes NP-Complete [6].
The representation use a propositionnal reified logic formalism, where a set of multi-valued domain attributes (fluents) are temporally qualified by two predicates: *hold* and *event*. The persistence of the value v of a domain attribute p during the interval $[t, t'[$ is expressed by the assertion $hold(p : v, (t, t'))$. An *event* is an instantaneous change at time t of the value (from $v1$ to $v2$) of a domain attribute $event(p : (v1, v2), t,)$
The representation allows multi-valued attributes with variable arguments, each instance of the arguments corresponds to an independent fluent (e.g. $p(?x)$ gives $p(a), p(b)...$).

A chronicle model is a conjunction of event predicates and constraints. It may also involve the description of some context that is required to hold over some interval independently of when the corresponding fluents became true. An example of a chronicle followas:

```
chronicle Situation
{       timepoint t1,t2,t3,t4,t5,t6;
  //- forthcoming events and context assertion
event (ATTRA:(stable, increasing),t1);
event (ATTRB:(stable, increasing),t2);
event (ATTRC:(stable, increasing),t3);
event (ATTRD:(stable, increasing),t4);
event (ATTRD:(stable, increasing),t5);
event (ATTRD:(increasing, decreasing),t6);
hold (ATTRE:None, (t1, t4));
  //- temporal constraints
(t2-t1) in [1.00,3.00];
(t3-t2) in [0.00,1.00]; (t4-t3) in [0.00,1.00];
(t5-t2) in [0.00,1.00]; (t6-t5) in [0.00,2.00];}
```

The system receives and processes only events. The chronicle recognition is complete as long as the observed event stream is complete, i.e. any change of the value of a domain attribute produces an observed event. This hypothesis enables to manage context assertions quite naturally through occurrences and non-occurrences of events. To process assertion $hold(p : v, (t, t'))$, the system checks that there has been an event $(p : (v', v), t'')$ with $t'' < t$ and such that no event $p : (v, v'')$ occurs within $[t'', t'[$. To initialize the system, a set of events corresponding to the initial state of the world must be sent with an occurrence date equals to $-\infty$. Consequently a chronicle model corresponds to a network of temporal constraints where each node is an event. The network for the previous chronicle is the one presented on Figure 4 with the condition that no event $ATTRE(NONE, ?t)$ occurs during $[t'1, t4[$.

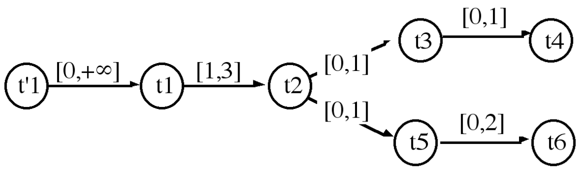

Figure 4: Chronicle Network.

3.2 On-Line Chronicle Recognition

The chronicle recognition system has to detect, on the fly, any subset of the stream of observed events which matches the set of model events in a chronicle and meets all the constraints and context assertions. A match between some observed events with a subset of model events is a partial instance of a chronicle. A complete match requires the observation of all model events[1]. However, not all observed events have to be accounted for: an event is meaningless unless it happens within a context and a sequence of events matching a chronicle

[1][9] has relaxed this limitation with optional events.

Let us illustrate the recognition process with a simple scenario corresponding to the chronicle model given above: initially $ATTRE$ is set to None, the $ATTRA$ increases at time 10', the $ATTRB$ increase at 12' and so on. On receiving a $ATTRA$ increase event at 10', the system detects a possible instance of this chronicle, it predicts what are the expected events for this instance and what should be their respective temporal windows compatible with this partial instance (See top figure 5 on the right). Reception of an event on $ATTRB$ increase at 12' is propagated to the windows constraining the remaining events (bottom figure). If, on some alternate scenario, after receiving these two events, either $ATTRC$ or $ATTRD$ events do not occur at time 13', or if $ATTRE$ changes its value before occurrence time of $ATTRD$ event, then the corresponding instance will not meet the specified constraints and will be disregarded.
A compilation stage is useful for testing the consistency of constraints in a chronicle model and for coding it into data structures efficient for the recognition process. It mainly consists into propagating all the constraints into a complete and minimal network. At preprocessing time, propagations are local to each chronicle model. For each chronicle model, we propagate constraints with an incremental path consistency algorithm. The result of this algorithm is (for each chronicle model) the least constrained complete graph equivalent to the user constraints, or a subset of inconsistent constraints if any.

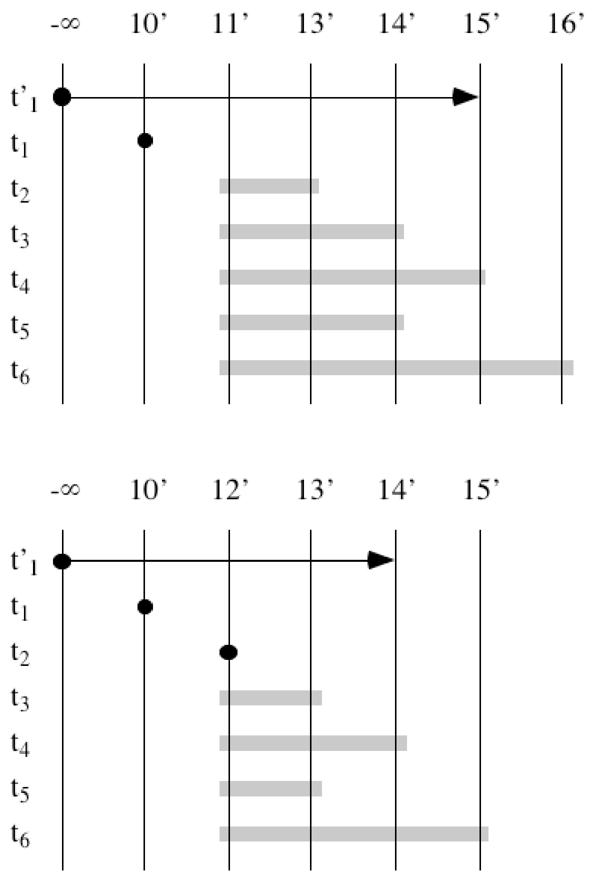

Figure 5: Matching Events and Propagating Windows.

An observed event may trigger a hypothesis of a chronicle instance. Such a partial instance corresponds to a network, grounded in time, predicting expected events for this hypothesis. Events that

may trigger a recognition hypothesis are those at the root nodes of the model network as well as at any further nodes compatible with the bounds on detection delays Δ of the predecessor nodes and the constraints.
The recognition process relies on a complete forecasting of expected events predicted by chronicle hypothesis. We define an interval, called window of relevance D(e), which contains all possible occurrence dates for a predicted event e of a partial instance S, in order for e to be consistent with constraints and known dates of observed events in S. Furthermore, for each chronicle instance, we maintain some time bounds. The deadline of an instance S is the latest date for the nearest non-occurred event. Similarly, we compute the holdline of an instance which is the latest date for ensuring that the nearest assertion holds, i.e. when we are sure that an occurrence of a forbidden event cannot violate the assertion (for the previous example, at time 10' deadline=16' and a holdline=15'). We can compute the recognition interval for each chronicle instance: the user can know when a recognition is expected and which events will be necessary to achieve it. This is used for performing some actions such as a focus of attention on what is expected or what needs to be avoided.
A hypothesis of chronicle instance may progress in two ways: (i) a new event may be detected, it can be either integrated into the instance and make the remaining predictions more precises, or it may violate a constraint for an assertion and make the corresponding hypothesis invalid; or (ii) time passes without nothing happening and, perhaps, may make some deadline violated or some assertions constraints obsolete. The system manages its own internal clock by receiving clock updates. Let now denoted the value of this clock. When an observed event e matches of model event e_k, we always have the reception date $r(e) = now$, and either

- $d(e) \in D(e_k)$: e doesn't meet the temporal constraints of the expected event e_k of S,
- $d(e) \ni D(e_k)$: $D(e_k)$ is reduced to the single point $d(e)$.

More generally, if the window of relevance of some expected event e_k in S has been further constrained (that is $D(e_k)$ reduced), we are sure that this constraint remains consistent with what is already known. We need however to propagate the reduction to other expected events, which in turn are further constrained.

```
$propagate(e_k, S)$
for all forthcoming event $e_i \neq e_k$ of $S$
$D(e_i) \leftarrow D(e_i) \cap [D(e_k) + I(e_i - e_k)]$
```

This produces a new set of non-empty and consistent $D(e_i)$. We never need to verify the consistency of events that fall into their windows of relevance. We just need to propagate further their new value of $D(e)$.
When the system updates its internal clock, this new value of now can reduce some windows of relevance $D(e_i)$ and, in this case, we need to propagate it over all expected events of an instance S (by $D(e_i) \leftarrow D(e_i) \cap ([t, +\infty[- D(e_i)))$. This instance remains consistent as long as all $D(e_i)$ remain non empty.
Notice that a clock update may not require propagation: it is necessary to propagate a date only when a time bound is reached (before that we are sure that constraints in a chronicle instance remain consistent). Time bounds enable an efficient pruning.
Checking hold assertions requires a special care. Let us consider a chronicle instance S which requires that an event e should not occur during $[e1, e2]$. If e occurs before $D(e1)$ or after $D(e2)$, S isn't

concerned. Otherwise, there are two basic cases (depending on the overlap between $D(e1)$ and $D(e2)$) and each one leads to a processing according to the localization of the occurrence date of e.
Before propagating a second observation into a hypothesis, the original chronicle instance must be duplicated and only the copy is updated. For each chronicle model, the system manages a tree of hypotheses of current instances. When a hypothesis of chronicle instance is completed or killed (because of a violated constraint), it is removed from this tree. Duplication is needed to warranty the recognition of a chronicle as often as it may arise, even if its instances are temporally overlapping.
We have briefly presented the temporal chronicle representation and the recognition algorithm used in the IxTeT-reco system. This work as inspired some interesting extensions (for Witas and with CRS) and appear to be one of the most promising one in this domain.

4 Future Work

In the current stage we plan to further study the integration of a temporal chronicle system in the COGNIRON architecture. This could either be CRS [26] or IxTeT-Reco [10]. The two systems are very close (they have been developed by the same person), still there are some pros and cons which we need to consider to make a final choice.

- IxTeT-reco has been somewhat extended to handle uncertainties (see Section 2.2).
- IxTeT-reco has been integrated in the LAAS architecture and connected to the procedural reasoning system (PRS, now Open-PRS) we use as a supervision and procedural executive system.
- CRS was rewritten from scratch, and may have a cleaner design and may offer a better support.

Another aspect we may pursue is the use of HMM and DBN to recognize particular interactions with humans and actions from a human or a from another robot. One of the nice property of these approaches is that they can rely on learned model, and deal with uncertainty. We already have conducted some experiments to learn an HMM of a robot “complex” navigation actions [12], and plan to extend this study to learn human “intentions”.

References

[1] D. Bernard, M. Borillo, and B. Gaume. From event calculus to the scheduling problem. *Journal of Applied Intelligence*, 1:195–221, 1991.

[2] X. Boyen, N. Friedman, and Daphne Koller. Discovering the hidden structure of complex dynamic systems. In *UAI*, August 99.

[3] S. Cauvin, M.-O. Cordier, C. Dousson, P. Laborie, F. Lévy, J. Montmain, M. Porcheron, I. Servet, and L. Trave-Massuyes. Monitoring and alarm interpretation in industrial environments. *AI Communication*, 11(3-4):139–173, 1998.

[4] L. Console, L. Portinale, D. Theseider, and P. Torasson. Diagnostic reasoning across different time points. In *Proc. 10th ECAI*, pages 369–373, 1992.

[5] T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Artificial Intelligence*, 93:1–27, 1989.

[6] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. In *KR'89: Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 1989.

[7] R. Demolombe and E. Hamon. What does it mean that an agent is performing a typical procedure? a formal definition in the situation calculus. In *AAMAS*, pages 905–911, 2002.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[9] O. Despouys. *An Architecture for Planning and Execution Control in Dynamic Environments*. PhD thesis, INPT, LASS-CNRS, France, December 2000. In French.

[10] C. Dousson. *Suivi d'évolutions et reconnaissance de chroniques.* Intelligence artificielle, Université Paul Sabatier, Toulouse, France, September 1994.

[11] C. Dousson, Paul Gaborit, and M. Ghallab. Situation recognition: Representation and algorithms. *Proc. of the* 13^{th} *IJCAI*, pages 166–172, August 1993. Chambry, France.

[12] M. Fox, M. Ghallab, G. Infantes, and D. Long. Robot introspection through learned hidden markov models. *Artificial Intelligence*, 2005. submitted.

[13] N. Friedman. The bayesian structural em algorithm. In *UAI*, pages 129–138, 98.

[14] N. Friedman, K. Murphy, and S.J. Russell. Learning the structure of dynamic probabilistic networks. In *UAI*, pages 139–147, 1998.

[15] Z. Ghahramani and M.I. Jordan. Factorial hidden markov models. *NIPS 8*, 1996.

[16] M. Ghallab. On chronicles: representation, on-line recognition and learning. In *Proceedings of the International Conference on Knowledge Representation and Reasoning (KR)*, pages 597–606, 1996.

[17] F. Heintz. Chronicle recognition in the witas uav project. In *the Swedish AI Society's annual conference 2001 (SAIS 2001), Skövde, Sweden*, 2001.

[18] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In Springer-Verlag, editor, *Fourth European Conference on Computer Vision*, pages 343–356, Cambridge, UK, 1996.

[19] R. Kowalski. Database update in the event calculus. *Journal of Logic Prgramming*, 12:121–146, 1992.

[20] K. Kumar and A. Mukerjee. Temporal event concep-tualization. In *Proc. 9th IJCAI*, pages 472–475, 1987.

[21] F. Lévy. Recognising scenarios: a study. *Fifth international workshop of diagnosis (DX '94)*, pages 174–178, September 1994. New Paltz.

[22] W. Nejdl and J. Gamper. Harnassing the power of temporal ctoins in modelbased diagnosis of dyna:ic systems. In *Proc. 11th ECAI*, pages 667–671, 1994.

[23] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE Acoustics, Speech & Signal Processing*, 1986.

[24] L.R. Rabiner. A turorial on hidden markov models and selected applications in speech recognition. *IEEE*, 77(2), February 1989.

[25] A.S. Rao. Means-end plan recognition - towards a theory of reactive recognition. In *Proc. 4th KR*, pages 497–508, 1994.

[26] Web Site. Crs: a chronicle recognition system. Technical report, France Telecom, http://crs.elibel.tm.fr, 1999.

[27] V.T. Vu, F. Bremond, and M. Thonnat. Automatic video interpretation: A recognition algorithm for temporal scenarios based on pre-compiled scenario models. In *In ICVS03*, page 523, 2003.

[28] G. Zweig and S.J. Russell. Speech recognition with dynamic bayesian networks. In *AAAI*, pages 173–180, 1998.