Time Aspects of Transparent Intensional Logic in Communication and Decision-Making of Agents

Jakub Macek and Tomáš Frydrych

VŠB-Technical University Ostrava
17. listopadu 15, 708 33 Ostrava, Czech Republic
jakub.macek.st@vsb.cz frydrych.t@gmail.com

Abstract. This paper shows pitfalls in the formal representation of a state of a multiagent system that is changing in time and we propose a solution based on Transparent intensional logic. Moreover, we propose a method for the definition of facts and rules and communication between agents in such a way that preserves consistency of agents' knowledge base. Using an example we present a set of methods for information retrieval and nondeterministic decision-making proces. These methods can be used by an agent that is entering the system with limited knowledge in order to reach a specified goal.

Key words: TIL, multiagent system, agent, communication

1 Introduction

Transparent Intensional Logic (TIL) [1] is a logical system founded by Prof. Pavel Tichý [4,5]. It is a higher-order system primarily designed for the logical analysis of natural language. As an expressive semantic tool it has a great potential to be utilised in artificial intelligence and in general whenever and wherever people need to communicate with computers in a human-like way.

Due to its rich and transparent procedural semantics, all the semantically salient features are explicitly present in the language of TIL constructions. It includes explicit intensionalisation and temporalisation, as well as hyper-intensional level of algorithmically structured procedures (known as TIL constructions) which are of particular importance in the logic of attitudes, epistemic logic and the area of knowledge representation and acquisition. We use TIL constructions for agents communication and decision-making proces.

Imagine an agent agent in a multiagent system [2,3]. The agent is able to make independent decisions and these decisions together with the rules defined in the system can cause the system transition from one state to another. The goal of the agent is to achieve a predefined state of the system.

In this paper we use a demo example of a game in order to illustrate our method and the proposed solutions. The system is a game of dungeon exploration and contains one or more independent players who know the rules of the game at the very beginning. The game consists of several places connected

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2010, pp. 61–67, 2010.

with pathways (thus resembling undirected graph). Cumulative information about position of all players in game is its state.

Players is able to make only one kind of a decision, namely where to move next. As a consequence of these decisions the game gradually changes its state as the players are moving around.

The key part of the game that moves it closer to a real life situation is that players have limited knowledge of the system. Each decision helps them obtain new information along the way with some of their knowledge becoming obsolete over time.

Beside agent-players who are capable of decisions we have another type of agent. These represent local sources of information and player agents simulate perceiving their surroundings by communicating with them.

2 Solution without Time

To show insufficiency of the solution without using temporal aspect of the problem, let's define more simpler game with limited scope. We will use just one agent-player (denoted by Z) and a map consisting of two places (A and B). The player will be able to move between these two places without restriction. We will introduce appropriate rules for the player movements into the system. We also have to include a rule stating that the player can be at one place at a time.

At the beginning, the player will be at place A. Knowledge base of game thus contains these facts and rules:

$$\forall x((LocatedAt(x,A) \land MovesTo(x,B)) \supset LocatedAt(x,B))$$
$$\forall x((LocatedAt(x,B) \land MovesTo(x,A)) \supset LocatedAt(x,A))$$
$$\forall x(LocatedAt(x,A) \supset \neg LocatedAt(x,B))$$
$$\forall x(LocatedAt(x,B) \supset \neg LocatedAt(x,A))$$
$$LocatedAt(Z,A)$$

And as a result of the last fact, we can also deduce that:

$$\neg LocatedAt(Z,B)$$

Now the player makes a decision to move in the place B:

$$MovesTo(Z,B)$$

The system infers which is the new position of the player:

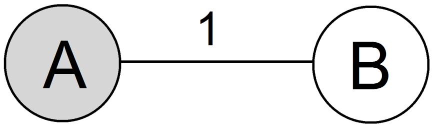
$$\forall x((LocatedAt(x,A) \land MovesTo(x,B)) \supset LocatedAt(x,B)) \land MovesTo(Z,B)$$
$$\models LocatedAt(Z,B)$$

We can now see that this conclusion contradicts the assumption that $\neg LocatedAt(Z,B)$. Obviously, the problem is due to the fact that we did not take into account temporal aspects of agents positions. This can be solved by removing invalid facts from knowledge base, but a more elegant solution is to introduce time to our formal representation of the system.

3 Introduction of Temporal Aspect to the Knowledge Base

Seemingly contradictory states can be avoided by introducing temporal aspect. Transparent intensional logic (TIL) uses real numbers for time variables, but for illustration we will limit ourselves in this paper to integers (representing individual seconds). For example the moment $t_2 = t_1 + 1$ is one second after the moment t_1. Additionally we define constant ${}^0Beginning \rightarrow \tau$, which will represent the first moment of the life cycle of our multiagent system.

First we define a define basic system with one agent Z and two places A, B that are linked through pathway 1 and are 30 seconds apart. Its obvious that the agent starting at the place A will move towards the place B and reach it at Beginning + 30.



The initial state of the system is constructed by the following construction:

$$\lambda w[[[{}^0LocatedAt\ w]{}^0Beginning]\ {}^0Z\ {}^0A]$$

Arguments of the function $LocatedAt/((o\iota)\tau)\omega)$ are an agent and a place, respectively.

Moreover, we introduce the rule for a movement of any agent $x \rightarrow \iota$ at time $t \rightarrow \tau$, which will represent moving through the only available pathway.

$$\lambda w \lambda t \lambda x[{}^0\supset\ [[[{}^0LocatedAt\ w]\ t]\ x\ {}^0A][[[{}^0LocatedAt\ w]\ [{}^0+\ t\ {}^030]\ x\ {}^0B]]$$

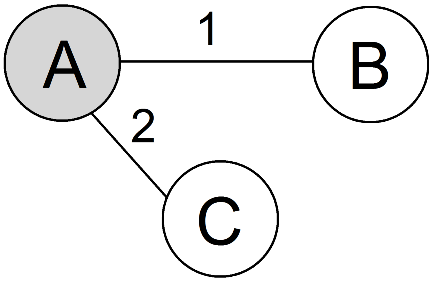
In natural language we can read this rule as follows: If at time t an agent x is at the place A then at time $t + 30$ the agent x will be at the place B.

Using this rule and information about its initial position, the agent can now deduce that after 30 seconds it will be located at place B.

$$\lambda w[[[{}^0LocatedAt\ w]\ [{}^0+\ {}^0Beginning\ {}^030]]\ {}^0Z\ {}^0B]$$

4 Decisions

In the previous section we considered a trivial situation in which the agent has no option and must move from the place A to the place B. Now let the rule that an agent cannot stay in the same place be valid and we add another possible destination C such that C is connected with A and the transition from A to C takes 20 seconds.



It is clear that we introduced some form of nondeterminism into the system. We need to compensate this by the concept of decision. This will be formalised by the following construction:

$$\lambda w[[[{}^0 Decision\ w]\ {}^0 Beginning]{}^0 Z\ {}^0 1]$$

Type $Decision/(((0\iota)\tau)\omega)$ where the first argument is an agent and the second argument is a pathway.

Now the rule describing a movement from one place to another can be expanded by adding a condition specifying that the agent decided to go through pathway 1. In the same way we create another rule for pathway 2.

$$\lambda w \lambda t \lambda x[{}^0\supset\ [{}^0\wedge[[[{}^0 LocatedAt\ w]\ t]\ x\ {}^0 A][[[{}^0 Decision\ w]\ t]{}^0 Z\ {}^0 1]]$$
$$[[[{}^0 LocatedAt\ w]\ [{}^0+\ t\ {}^0 30]\ x\ {}^0 B]]$$
$$\lambda w \lambda t \lambda x[{}^0\supset\ [{}^0\wedge[[[{}^0 LocatedAt\ w]\ t]\ x\ {}^0 A][[[{}^0 Decision\ w]\ t]{}^0 Z\ {}^0 2]]$$
$$[[[{}^0 LocatedAt\ w]\ [{}^0+\ t\ {}^0 20]\ x\ {}^0 C]]$$

We need to properly inform the agent about the need to make a decision and the possible options. For the place A that means to offer the agent to use either pathway 1 or 2. In order to comply with physical laws, we also specify that the agent can make only one of these decisions at a particular time t.

$$\lambda w[{}^0\supset\ [[[{}^0 LocatedAt\ w]{}^0 Beginning]\ {}^0 Z\ {}^0 A]$$
$$[{}^0\vee[[[{}^0 Decision\ w]\ {}^0 Beginning]{}^0 Z\ {}^0 1][[[{}^0 Decision\ w]\ {}^0 Beginning]{}^0 Z\ {}^0 2]]]$$

$$\lambda w \lambda t \lambda x[{}^0\supset\ [[[{}^0 Decision\ w]\ t]x\ {}^0 1][{}^0\neg[[[{}^0 Decision\ w]\ t]x\ {}^0 2]]$$
$$\lambda w \lambda t \lambda x[{}^0\supset\ [[[{}^0 Decision\ w]\ t]x\ {}^0 2][{}^0\neg[[[{}^0 Decision\ w]\ t]x\ {}^0 1]]$$

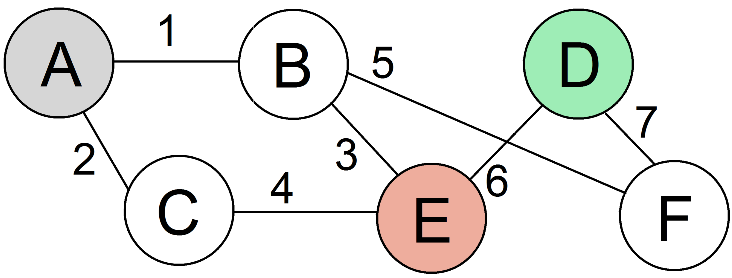
The decision made by the agent is a new fact in the knowledge base. Nevertheless we must ensure that agents do not pollute the system knowledge base with contradictory facts by specifying which facts can be introduced. For instance, we can introduce a constraint that only facts generated from a particular scheme not contradicting agent's knowledge are decisions. In our example the scheme can be: "any proposition containing only function *Decision* with parameters: current world as w, current time as t, me (agent-player) as x and any pathway individual as remaining argument".

For example if an agent is located at the place B at time t, then it cannot go through the pathway 2, since it is not available. In the same way we could introduce temporary unavailability of the pathway 1 from the place A (with regard to time t at which the decision is being made).

5 Simulation

So far we have defined agent-player that is capable of random decisions based on its knowledge base and blind exploration of the game system. Of course, such behavior is not very useful.

Let us extend the definition of our system so that the system behaviour is closer to the dungeon exploration. We will add more places and define positive and negative goals to motivate the agent. Positive goal (finding the tresure) will be defined as "agent is at place D". Negative goal (meeting a grue or falling into a chasm) will be defined as "agent is at place E".



$$\lambda w \lambda t \lambda t_1 \lambda x [^0\supset\ [^0\wedge[[[^0LocatedAt\ w]\ t]\ x\ ^0B][^0\geq\ t_1\ t]]$$

$$[[[^0PositiveGoal\ w]\ t_1]\ x]]$$

$$\lambda w \lambda t \lambda t_1 \lambda x [^0\supset\ [^0\wedge[[[^0LocatedAt\ w]\ t]\ x\ ^0C][^0\geq\ t_1\ t]]$$

$$[[[^0NegativeGoal\ w]\ t_1]\ x]]$$

These propositions representing goals have to affect the decision-making process. Decisions leading to positive goals will be favored, whereas decisions leading to negative goals will be avoided. However we must take into account that there is a possible situation in which every decision will lead to a negative goal. Therefore we cannot just dismiss such decisions, because the agent would run out of possible options and that would contradict previously defined rule.

In order that the agent knows the effects of its decisions these decisions must be simulated. To preserve consistency of information in knowledge base each simulation will have a separate simulated knowledge base. At the beginning of any simulation the simulated knowledge base will contain everything that is in real knowledge base (ie. information about agent's position, rules of movement, etc.), but all facts deduced during simulation will be stored only in the separate knowledge base.

The depth of simulation (meaning maximum number of simulated decisions in one simulation) will have direct influence on agent's intelligence. However the agent can make only one decision at time, because the state of the system in future is not guaranteed to be the same as in a simulation (and most likely will not be). To determine which possible decision is the best one, the agent needs to properly rank each one according to outcomes of simulations.

Decisions that ended with a positive goal in the first step (after the first decision) are considered to be optimal. If a positive goal has been reached later in the simulation, the decision should be ranked as slightly less optimal. Immediate decisions which didn't lead to a positive or negative goal are considered to be neutral, and finally decisions leading to a negative goal are ranked as the least optimal.

Immediate decision that ranked best will be done by the agent (meaning introduced both to its own and global knowledge base) and proper consequences will be deduced. In our example the consequence would be agent's position in following moments.

6 Agent-Informer and Communication

At the beginning the agent-player doesn't know all facts and rules describing the system. In fact, such a situation is not actually possible, because agent would be by definition omniscient. Instead the agent knows only those rules that apply now and here. It can obtain this information by communication with immobile agents-informers.

In reality this agent-informer would be agent-player's ability to get additional information by observing its surroundings. In our example the primary information is the list of possible decisions, in particular possible destinations. Agent-player will keep this information in its knowledge base.

To perform sufficiently extensive simulation, the agent needs also information about other places (in other words it needs to create a temporary map of nearby places). This information can be provided by agent-informer, too, but only for actual moment. It cannot provide reliable information about future state of the system. If agent obtained at place A information about pathway 5 that always leads from place B and when at place B obtained information that pathway 5 is temporarily unavailable, it would create a contradition in its knowledge base.

Information about future is always only assumed and solution to mentioned problem is to treat it as such. When agent-informer gives assumptions to agent-player, it will limit them to simulated environment and possibly also limit their usablity to a specific time span.

Simulation will be represented with special fact $Simulation/(o\iota)$, which will be true only if agent given as parameter is performing a simulation. For example, agent Z will introduce following fact to its simulated knowledge base:

$[{}^0Simulation\ {}^0Z]$. Using this fact as a condition we can transfer assumptions from agent-informer to agent-player the same way as real facts.

7 Conclusion

In this paper we demonstrated a method for the definition of facts, rules and communication between agents in the multiagent system that is changing in time and presents solution using Transparent intensional logic. Using an example we presented a set of methods for information retrieval and nondeterministic decision-making proces.

Acknowledgements

This research has been supported by the Grant Agency of the Czech Republic, project 401/09/H007 'Logical Foundations of Semantics', and also by the internal grant agency of VSB-TU of Ostrava – SP/2010214 Modeling, simulation and verification of software processes II.

References

1. Duží, M., Jespersen, B., Materna, P.: *Procedural Semantics for Hyperintensional Logic (Foundations and Applications of TIL)*. Springer. 2010. 17. Logic, Epistemology, and the Unity of Science.
2. Duží, M., Ciprich, N., Košinár, M., Kohut, O., Frydrych, T.: *The Architecture of an Intelligent Agent in MAS*. Waki Print Pia, Kanagawa, Japan. 2008.
3. Kohut, O., Košinár, M., Frydrych, T.: *TIL in Knowledge-Based Multi-Agent Systems*. Masarykova universita. 2008. 11.
4. Tichý, P. *The Foundations of Frege's Logic*. Walter de Gruyter, Berlin-New York, 1988.
5. Tichý, P. (2004): *Collected Papers in Logic and Philosophy*, V. Svoboda, B. Jespersen, C. Cheyne (eds.), Prague: Filosofia, Czech Academy of Sciences, and Dunedin: University of Otago Press.