Communication of Intelligent Agents Integrated to GIS

Ondřej Kohut[1] , Michal Košinár [2]

Katedra Informatiky, FEI, VŠB – Technical University Ostrava, 17. listopadu 15/2172, 708 33, Ostrava - Poruba, Czech Republic

[1]ondrej.kohut.fei@vsb.cz
[2]michal.kosinar@vsb.cz

Abstrakt. Transparentní intensionální logika (TIL) je vysoce expresivní logický systém. Vzhledem k její síle se nabízí její široké užití v oborech umělé inteligence a multiagentových systémech. Představíme jazyk TIL-Script jako počítačovou reprezentaci jazyka konstrukcí TIL se sémantikou definovanou v jejich pojmech. TIL-Script je primárně navržen jako alternativa k jazyku FIPA SL (Semantic Language) a kdykoliv je třeba využít přirozeného jazyka (např. interakce člověk/počítač) anebo když jsou jiné systémy nedostatečné (například znalosti zdrojově omezených agentů). Ukážeme příklad komunikace agentů v GIS užívající TIL-Script a předvedeme implementaci úlohy demonstrující výměnu znalostí v závislosti na vidění agentů.

Klíčová slova: MAS, GIS, rozhodování, komunikace, Transparentní Intensionální Logika, TIL-Script.

Abstract. Transparent Intensional Logic (TIL) is a highly expressive logic system. Its potential applications in artificial intelligence and multi-agent systems are wide. We propose language TIL-Script as a string representable language of TIL constructions, with semantic defined in terms of TIL constructions. TIL-Script could be used as an alternative to FIPA SL language, whenever natural language is in need (e. g. human/computer interaction) and/or another systems are insufficient (for example knowledge of resource bounded agents). We will discuss usage of TIL-Script in the communication of agents in GIS and also present simple demonstration of knowledge exchange according to agents' visibility what we have implemented.

Keywords: MAS, GIS, reasoning, Communication, Transparent Intensional Logic, TIL-Script.

1 Introduction

Multiagent systems is a relatively new technology which is still fast developing. One of the main topics of it is a communication and communication languages. Actual content languages are based on mathematical logics and they are extending them to fit their purpose. But because of those extensions are made at a syntactical level their semantics is not very clear.
We propose a language TIL-Script based on Transparent Intensional Logic (TIL) which is a general (philosophical) logic primarily designed for a logical analysis of natural languages. This language is well suitable to be used as a content language for agent messages and it has a well elaborated semantics. It can be also used as a general semantic framework for specifying semantics of known formal languages.
In the first section we briefly introduce the Transparent Intensional Logic. The second section discus the FIPA's content language SL and present some problems related to it. Then we introduce the TIL-Script language. At first by discussing the differences of TIL-Script and TIL, and then a main part of its grammar is presented. There are some examples of analysis using both standard TIL notation and the TIL-Script language. An example of communication and knowledge exchange in a multiagent system according to agents' visibility is presented in the last section.

2 Transparent Intensional Logic

Transparent Intensional Logic (TIL) is a logical system founded by Prof. Pavel Tichý. It was designed to fill a gap in the logical analysis of natural language (see [7]). This gives TIL a great expressive power. TIL is concerned to be a philosophical logic, that means it is motivated by non-mathematical domains. As an expressive semantic tool it has a great potential in artificial intelligence and any other area where both computers and humans are to be dealt together. More about the role of logic in artificial intelligence can be found in [6].
TIL is capable of capturing (nearly) any semantic feature of natural language. It includes temporal and modal attitudes, epistemic logic and knowledge representation (even modeling resource bounded knowledge), change of the state of the world, etc.
We will introduce TIL just briefly. For detailed explanation see [1, 2].

2.1 Basic Notions

The fundamental concept of TIL is a notion of construction. Constructions are analogical to formulas and terms of traditional logics, but there are several fundamental differences. Most logics make a strict border between semantics and syntax. Formulas are used to be defined as well formed sentences of some formal language. That means they are mere strings and an interpretation is needed to reveal their meaning (semantics).
Constructions are not language expressions (strings). They are abstract procedures – object that are algorithmically structured. Because constructions are themselves semantic objects they don't need to be interpreted and they contain both the semantic and the syntactic components.

2.2 Types

Constructions are coded (and presented) in a language, which is formally derived from the language of typed lambda-calculus (but these strings are not constructions themselves, they are just forms of presentation of the constructions). Types are just collections of member objects. For a type α we call its members 'α-objects'.

Types in TIL arise from a type base. Type base is a (finite) set of basic (atomic) types. Standard type base of TIL is an epistemic base (but is not the only one possible) containing types:

- ο – Truth values. Type consisting of two values: True and False.
- ι – Individuals. Simple objects: the 'lowest-level bearers of properties'.
- τ – Time points. This type is just a set of real numbers.
- ω – Possible worlds. The collection of all logically possible states of the world.

Other types are defined as functional closures of atomic types. The collection of all (even partial) functions from types $\beta_1, \ldots, \beta_n$ to type α is a type. We denote this type $(\alpha\beta_1, \ldots, \beta_n)$. These types are called types of order 1. As we'll see later, there are types of higher orders too.

2.3 Constructions

Constructions are the elementary building stones of TIL. Depending on valuation v, any construction *v*-constructs an object of some type, or is v-improper (doesn't v-construct anything). TIL is a logic of partial functions.

We recognize four kinds of constructions:

Trivialization ^0a is an elementary construction constructing the given object a in the simplest (unstructured) way.

Variable *x* is a construction ("x" is just a name). What it constructs depends on valuation *v* we
say it *v*-constructs (in valuation *v*).

Application $[F\ C_1 \ldots C_n]$ is an application of function to its arguments. If *F* *v*-constructs *f* of type $(\alpha\beta_1, \ldots, \beta_n)$ and each C_i *v*-constructs c_i of type β_i, then the whole application *v*-constructs the result of applying *f* on $c_1 \ldots c_n$ which is $f(c_1, \ldots, c_n)$.

Abstraction $\lambda x_1 \ldots x_n\ C$ Like in lambda calculus: if variables x_i range over β_i and *C* *v*-constructs object of type α, then the abstraction *v*-constructs following function *f*: let v_0 be a valuation that associates x_i with b_i and is identical *v* otherwise. Then *f* is undefined on $b_1, \ldots, b_n$ if *C* is *v'* improper, otherwise the value of *f* on $b_1, \ldots, b_n$ is what is *v'*-constructed by *C*.

Each construction is of some order. The order of construction is the highest order of type involved in the construction.

2.4 Higher order types

Basic type of higher order is the type $*_i$ – the collection of all constructions of order i. Any type of order i is also type of order i + 1. Other types of order i are functional closures of defined types as shown in section 2.2.

3 Multiagent Systems and Communication

Technologies based on agents are relatively new and very promising. A big amount of their applications can be found in artificial intelligence and large computer systems. A road-map of this approach is presented in [5].
We do not intend to talk about multiagent systems (MAS) in general. In this paper we focus on communication in MAS and particularly on content languages.
Basic standards for MAS are given by FIPA (The Foundation for Intelligent Physical Agents, see [4, 3]). According to it basic unit of communication is a message. It can take arbitrary form, but it is supposed to have a structure containing several attributes. Content of the message is one of these attributes.

From the point of view of communication logic the most important attributes are:

Performative denotes a type of the message – its communicative act. Basic performatives are: *Query, Inform and Request*.

Content carries the semantic of the message. It can be encoded in any suitable language.

Ontology is a vocabulary of domain specific terms. These (and only these) terms can be used in the content of the message.

4 FIPA SL

One of objectives of this paper is to propose a new content language for multiagent systems. First we will briefly discuss existing standard – FIPA SL.
FIPA SL (Semantic Language) is the only FIPA candidate content language marked as "standard". It is based on language of First Order Logic (FOL), but it extends its capabilities. One of advantages of this approach is that FOL is a well known logic and it is well elaborated. But there are disadvantages too. First, FOL is a mathematical logic. As it, its development was motivated mainly by mathematics and FOL is really good for describing algebraic structures. But this is usually not the way it is used in multiagent systems.
For utilization in multiagent systems FOL needed to be extended. Formulas of FOL express only assertions. But also queries and requests are valid messages. So SL defines so called identifying expressions. Furthermore SL is capable of propositional attitudes – that means assertions about another assertions, i.e. "John believes that it is raining.".
Syntactically is the SL well defined but problems occur when one wants to know its semantics. There is no proper specification of semantics in the standardization document but the section "Notes on FIPA SL Semantics", which is (as it says) just notes. The standard counts upon well known semantics of FOL, but because of the numerous extensions it is not applicable. This lack of semantics can have very unpleasant consequences. Two agents relying completely on the standard can understand the same message differently and that can

lead to serious misunderstandings between the agents. This is in a direct conflict with the name – “Semantic Language”.

5 The TIL-Script Language

We claim that Transparent Intensional Logic is very suitable logic system for utilization as a content language in multiagent systems. For this purpose main advantages arise from its:

Semantic nature Constructions of TIL are themselves semantics objects. So the semantics is naturally well defined. There is no danger of misunderstandings as with the SL language.

High expressibility The expressive power of TIL is really high. TIL is capable of analyzing nearly any semantic feature of natural languages.

Original purpose TIL unlike mathematical logics was intended to be a tool for logical analysis of communication. Primarily it was designed for natural languages, but this gives it a great potential even in other areas.

5.1 Type base

The type base of TIL-Script is an extended epistemic type base. To make it more convenient we add some types common in informatics. The type of actions is added too.

Bool – Type of truth values.
Indiv – Type of individuals.
Time – Type of times.
World – Type of possible worlds.
Real – Real numbers.

5.2 Composite Types

Like TIL, composite TIL-Script types are collection of all (even partial) functions on defined types.
In the following we will show the notation of composite types in TIL-Script in examples.

(Bool Indiv) – Type of functions from individuals to truth values.
(Bool Time Indiv) – Type of functions from time points and individuals to truth values.
((Indiv)Time)World – Type of individuals in intension (individual offices).
*1 – Type of constructions of order 1.

5.3 Examples

At this point we will present an example of sentence of natural language (English) analyzed in TIL and its encoding using the TIL-Script language.

We won't go to details of natural language analysis using TIL, but we state some necessary information. Empirical claims, which truth value depends on actual state of world and/or time, are analyzed in TIL as constructions constructing a function mapping a state of world and a time point to the truth value of the claim in this state of world and time.

The highest mountain is in Asia.
Highest / ((Indiv(Bool Indiv))Time)World – Selects the highest element from a set (according to time and state of the world).
Mountain / ((Bool Indiv)Time)World – A property of being a mountain. That means a set of mountains in given state of world and time.
In / ((Bool Indiv Indiv)Time)World – A test if an individual is 'in' another individual (according to time and state of the world).
Asia / Indiv – The individual 'Asia'.

TIL-Script Code:
```
\w\t['In@w,t [[\w\t['Highest@w,t 'Mountain@w,t] w] t] 'Asia]
```

6 Ontologies for TIL-Script

Any content language is tightly related to ontologies. All concepts used or mentioned by a content language must be defined in a ontology. And vice versa, the content language must be able to use any concept from the ontology.
FIPA's definition of ontology is relatively vague. It just says that ontology provides a vocabulary of domain specific concepts and relations between them. This leads to diversity in implementations. Actually ontology takes a frame-like structure, which is well suitable for the FIPA SL language and developer frameworks like Jade support it.
Latest trend is to use well-proven technologies of semantic web. Namely the OWL language for defining ontologies. But actual implementation tools for multiagent systems doesn't support OWL very well. The way we have chosen for TIL-Script is to connect it to frame-like ontologies, because of an implementation in Jade. Integration of OWL into TIL-Script is a subject of our latest research.
Concepts (or classes) of ontologies are sets of so called individuals. Do not confuse these individuals and member of TIL-Script's type Indiv. Ontology individuals can be objects of any TIL-Script's type. For TIL-Script that means that any ontology concept (class), which members are of type , is an object of type (οα) a set of α-objects. Ontology individuals (members of classes) are directly α-objects.
Connection of TIL-Script to an ontology is done by the trivialization. You may trivialize any concept or individual defined by the ontology. It is forbidden to trivialize an object not defined by the ontology in use. The only extra request for an ontology to be used together with TIL-Script is that any class must have defined the TIL-Script type of its members.

7 An Example

In this section we present a simple example scenario of communication of two agents in a multiagent system using the TIL-Script language.

Scenario

The situation is simple. There is a large car parking, train station and two agents:

- Driver – An agent driving a car, who wants to park at the car parking rather near the train station.
- Dispatcher – A dispatcher of the car parking managing the pull-ins.

A sketch of their communication is as following:

1. Driver: I want you to park me somewhere near to the train station.
2. Dispatcher: OK, I can park you at this pull-in (concrete id).
3. Driver: I accept.

The Ontology

In order to analyze the communication using TIL-Script we need an ontology of used concepts.

TheDriver / Indiv – The driver agent.
TheDispatcher / Indiv – The dispatcher agent.
TheTrainStation / Indiv – The train station the driver wants to park near.
Pull-in / (Bool Indiv) – A class of pull-ins at a car park. It has a slot Id, which is a number indentifying a concrete pull-in.

Near / ((Bool Indiv Indiv)Time)World – A binary predicate saying that two places are near (at given time and state of the world).

Arrange / (Bool (Indiv ((Bool)Time)World)) – A function returning an action to be performed by an agent (the first parameter) to arrange that a proposition (the second argument) become true.

Park / ((Bool Indiv Indiv)Time)World– A binary predicate saying that given object parks at the car park at given pull-in.

Communication
Now we can reconstruct the communication between the driver and the dispatcher precisely.

Driver: Call for proposal:

```
\x:Indiv['Arrange 'TheDispatcher \w\t['And['Park@w,t 'TheDriver x]['Near@w,t
    'TheTrainStation x]]]
```

Note: A content of call for proposal communicative act is a function taking one argument. It returns an action to be proposed and a responding agent will fill the parameter according to his proposal.

Dispatcher: Propose:

```
[\x:Indiv['Arrange 'TheDispatcher \w\t['And ['Park@w,t 'TheDriver x]['Near@w,t
    'TheTrainStation x]]]
        ['Pull-in [Id_36]]]
```

Driver: Accept proposal: (the content is the same as in proposal)

8 Communication according to visibility

Often we count on the fact that agent's are equipped with some GPS device and thus it can by fully informed about its environment. But also we need to reason about the agents which are really autonomous, i.e. without any special device; they should be entirely dependent only on its sensors. For illustration see fig. 1. If the agent A receives its visibility (from some GIS agent for example) then it knows locations of both agents B and C. But if no GIS agent (or other information resource) is at disposition then the agent C is not visible for the observe agent. The agents can use only their "perceptions", its knowledge and communication skill. Although the agent A cannot see the agent C, it can establish contact with the agent B and obtain information about objects and agents behind the obstacle.

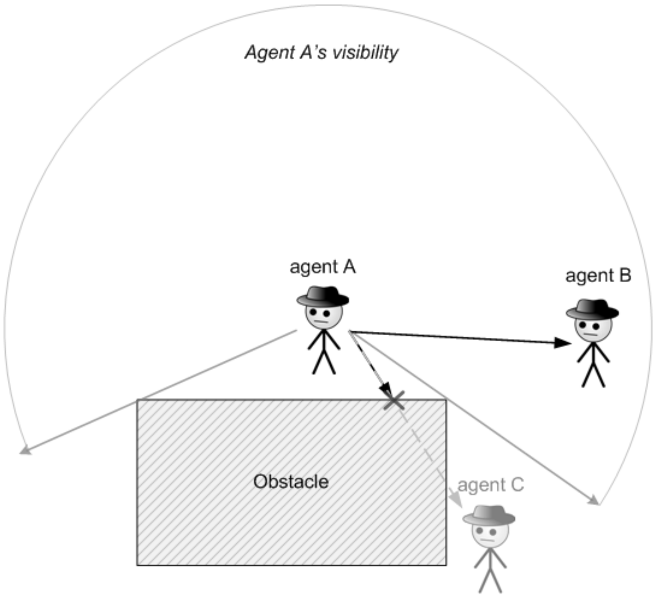

Fig. 1: Real visibility

We have implemented a simple demonstration of knowledge exchange by use of the communication. Agents' knowledge is represented by prolog rules and facts. Each agent has essential abilities to manage with rules and facts. Agents can require other agents for needed knowledge and also afford the requested knowledge. In the demonstration, several instances of the agent were created. Each of them had different set of special rules and facts and it had its goal. The goal of our agent was to solve some problem represented by prolog query. But no agent was able to solve the problem alone. The only way to solve it was the communication with other agents and exchange knowledge. Each agent was walking on its circuit and if it met another agent (i.e. the agent is in its circular visibility) then it started communication with the seen agent.

The algorithm for knowledge acquisition can be briefly described as follow. At first agent ask for the goal query. If it obtains a rule or fact that isn't contained in its knowledge base then the agent add it into its knowledge base. Afterwards it selects recursively the predicates from the start query and asked for the rules with the same head. For illustration see following.

Let's take these rules:

```
A:-B,C.
C:-D.
E:-F,G.
```

where A is the goal. We need to generate S – the set of all important predicates. At the beginning S is empty. At first A is added to S, then B and C and afterwards D. Unreachable

predicates (E,F,G) are not interesting, thus they are not contained in S. While the asking, the asked predicates are removing from S and after another agent's answer S is also dynamically extending. The 'interview' of the agents ends if S is empty or if query is solved. The agent remember that another agent is 'totally asked' and if it will be needed it can ask it again in the future after a timeout – the asked agent can receive another knowledge from other agents.
In this simple demonstration a very plain content language was used instead of TIL-Script. It had only three types of messages:
UNKNOWN *predicate_head* – request for prolog rules and facts with given *head*
REFINE *rules_and_facts* – answer containing desired prolog *rules and facts*
SORRY *head* – answer which means that answer doesn't carry any rule or facts for given *head*.

We have simulated a situation when agents are able to communicate only if they can see each other. Still no obstacles were implemented, but we showed how it could look if agents are dependent only on their perception. Preview of our simulation is shown in fig. 2.

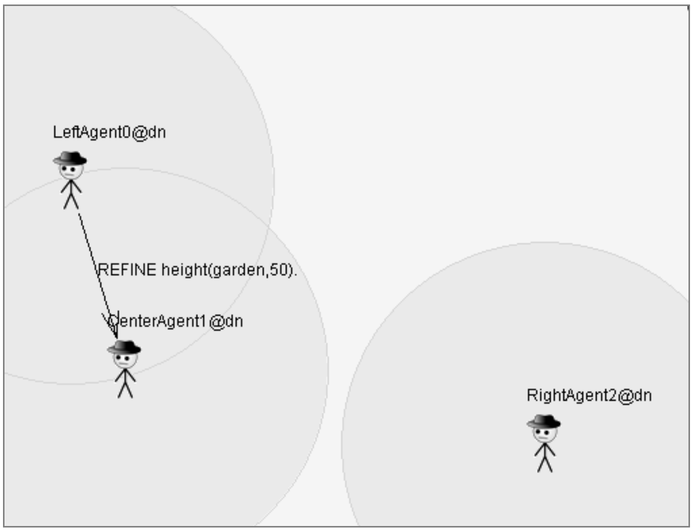

Fig. 2: Preview of simulation

9 Conclusion

Actual standards for communication in multiagent systems are based rather on syntax than on semantics. This can slow down the progress in future research. As an option to this we propose the language TIL-Script, which is based on a well elaborated Transparent Intensional Logic. This makes TIL-Script a semantically based language suitable for multiagent systems. High expressive power of TIL-Script makes it also a good tool for adopting other logics and languages into his semantic framework, so that TIL-Script can be used as a specification language. A big potential of TIL-Script can be also found in logical analysis of natural languages and so communication with human (non-expert) agents.
The TIL-Script language is being implemented and tested in multiagent systems using the Python language and Jade based simulation programs. Ontology support is made for frame-like ontologies supported by Jade, but as a next research adoption of OWL ontologies using Protége was developed and tested as a separate project.
We have also shown the "how to" let agents communicate with the respect to the GIS tier and proposed the way of the future implementation of the communication.

Acknowledgements. This research has been supported by the program "Information Society" of the Czech Academy of Sciences, project No. 1ET101940420 "Logic and Artificial Intelligence for multi-agent systems"

References

1. Marie Duží, Bjorn Jespersen, and Jaroslav Muller. Epistemic closure and inferable knowledge. In Libor Běhounek and Marta Bílková, editors, the Logica Yearbook 2004. Filosofia, 2005.
2. Marie Duží and Pavel Materna. Constructions. http://til.phil.muni.cz/text/constructions_duzi_materna.pdf, 2000.
3. Foundation for Intelligent Physical Agents. FIPA Abstract Architecture Specification. http://fipa.org/specs/fipa00001/, 2000.
4. Foundation for Intelligent Physical Agents. FIPA SL Content Language Specification. http://fipa.org/specs/fipa00008/, 2000.
5. Michael Luck, Peter McBurney, Onn Shehory, and Steve Willmott. Agent Technology: Computing as Interaction A Roadmap for Agent-Based Computing. AgentLink, 2005.
6. Richmont Thomason. Logic and artificial intelligence. In Edward N. Zalta, editor, The Stanford Encyclopedia of Philosofy. Summer 2005.
7. P. Tichy. The Foundations of Frege's Logic. Walter de Gruyter, Berlin-New York, 1988.
8. Ciprich, N., Duží, M., Košinár, M.: *The TIL-Script language*. To appear in the Proceedings of the 18th European Japanese Conference on Information Modelling and Knowledge Bases (EJC 2008), Tsukuba, Japan 2008.