TIL in Knowledge-Based Multi-Agent Systems

Tomáš Frydrych, Ondřej Kohut, Michal Košinár

VŠB–Technical University Ostrava
17. listopadu 15, 708 33 Ostrava, Czech Republic
tomas.frydrych.st@vsb.cz, ondrej.kohut.fei@vsb.cz, michal.kosinar@vsb.cz

Abstract. Transparent Intensional Logic (TIL) is a highly expressive logic system. Its potential applications in artificial intelligence and multi-agent systems are broad. We introduce the TIL-Script language, which is a computational variant of the language of TIL constructions. TIL-Script can be used as an alternative to FIPA SL language, whenever communication in a natural language is in need (e. g. human / computer interaction) and/or another systems are insufficient in their expressive power (for example in the area of knowledge representation for resource bounded knowledge based agents).

Key words: TIL, TIL-Script, Transparent Intensional Logic, Multiagent Systems, content language, FIPA

1 Introduction

Multi-agent systems are a relatively new technology which is still rapidly developing. One of the main problems a multi-agent system must deal with is communication and reasoning of agents. Current content languages are based on the first-order mathematical logic paradigm, extending the first-order framework whenever needed. However, these extensions are mostly specified syntactically, and their semantics is not very clear.

We propose the TIL-Script language based on Transparent Intensional Logic (TIL) which is a general (philosophical) logic system primarily designed for the logical analysis of natural languages. TIL-Script is well suited for the use as a content language of agent messages and it has a rigorously defined semantics. It can be also used as a general semantic framework for the specification and semantic analysis of other formal languages.

The paper is organized as follows. Section 2 describes the basic principles applied in agents' communication and interaction with environment. In Section 3 we discuss and critically examine FIPA SL content language. In Section 4 we briefly introduce the basic notions of TIL. In Section 5 ontologies and knowledge base model for TIL Script are briefly described. Finally, an example of agents' communication is presented in Section 6, and concluding remarks in Section 7.

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2008, pp. 31–40, 2008.

2 Multi-Agent Systems and Communication

Technologies based on agents are relatively new and very promising. Numerous applications of multi-agent technology can be found in artificial intelligence and large computer systems. A road-map of this approach is presented in [5]. In this paper we do not intend to deal with multi-agent systems (MAS) in general. Instead, 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 be of an arbitrary form, but it is supposed to have a structure containing several attributes. *Content* of a 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.

2.1 Agent and Environment Interaction

In order to introduce communication based on agents' knowledge, we are going to describe agents' reactions to the events in their environment, and interaction with the environment in general. Figure 1 illustrates agents' interaction with the environment.

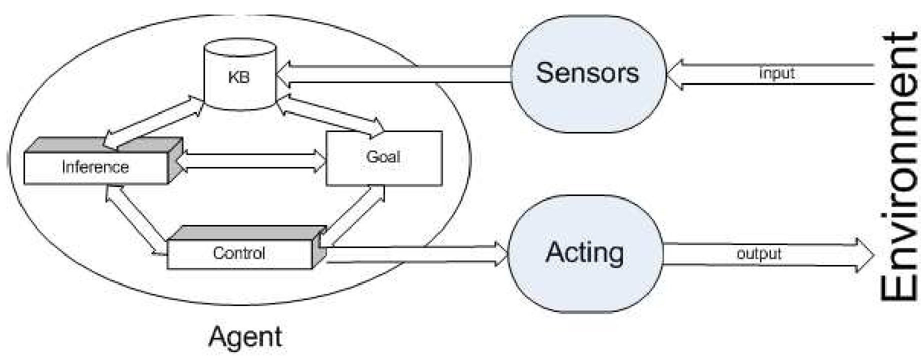

Fig. 1. Behaviour of agents in a real environment

Agents are autonomous, rational and goal-oriented. In order they can actively react on the events in their environment, they have to be equipped with:

- *Sensors* – “Ears”, “Eyes”
- *Acting parts* – “Mouth” for communication, “Limbs” for an active reaction (movement etc.)
- *Knowledge-Base* based on ontologies. This part serves as an agents’ memory that makes it possible to store perceived or learnt facts, entailed knowledge as well as general rules. (At least a minimal) ontology is needed to be shared with other agents, so that the agents’ understand each other.
- *Inference engine* that is based on the TIL-Script language (or Description Logic, FIPA SL, etc.)
- *Goals* are the purpose of agents’ life. An agent attempts to meet the goal assigned to them by applying their explicit knowledge stored in the knowledge base, and or inferred by the inference machine.
- *Control* part executes the actions to in accordance with a given agent’s goal. In this way the agent influence its environment.

3 FIPA SL

One of the objectives of this paper is to propose a new content language for agents’ communication in multi-agent systems. The languages like FIPA SL (Semantic Language) and KIF are mostly based on the First-Order Logic (FOL) paradigm, enriched with higher-order constructs wherever needed.[1] The enrichments extending FOL are well defined syntactically, while their semantics is often rather sketchy, which may lead to communication inconsistencies. Moreover, the bottom-up development from FOL to more complicated cases yields the versions that do not fully meet the needs of the MAS communication. In particular, agents’ attitudes and anaphora processing create a problem. In the paper we are going to demonstrate the need for an expressive logical tool of Transparent Intensional Logic (TIL) in order to encode the semantic content of messages in a near-to-natural language. Using TIL, the human-computer interface and communication is designed in a smooth way.

In this section we now briefly discuss the existing standard, FIPA SL, which is the only FIPA candidate content language for a ‘standard’.

One of advantages of the FOL approach is that FOL is a broadly used and well elaborated logic. But there are disadvantages as well. First, FOL is a mathematical logic. Its development was motivated by the program of logistic in mathematics. FOL is thus well suited for describing algebraic structures. But agents do not communicate in terms of algebraic structures. Moreover, Formulas of FOL express only assertions. However, queries and requests are also valid messages. Thus SL defines so-called identifying expressions. In SL we can also express propositional attitudes, i.e. assertions about other assertions like “John believes that it is raining.”. However, these attitudes are dealt with as relations of a believer to a piece of syntax.

---

[1] For details on FIPA SL, see `http://www.fipa.org/specs/fipa00008`; for KIF, Knowledge Interchange Format, see `http://www-ksl.stanford.edu/knowledge-sharing/kif/`.

SL is well defined syntactically, but problems appear when one wants to know its semantics. There is no proper specification of semantics in the standardization document; only the section "Notes on FIPA SL Semantics" can be found, which is (as it says) just notes. The standard counts upon well-known semantics of FOL, but due to numerous extensions it is not always applicable. The 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".

Another problem connected with the FIPA SL is that SL content is in general not interconnected with the semantic web. On the other hand, in our TIL-Script language we developed an interconnection between OWL concepts and TIL-Script.

For these reasons we propose the TIL-Script language to be used as a FIPA ACL content language.

4 Transparent Intensional Logic (TIL)

Transparent Intensional Logic (TIL) is the logical system founded by Pavel Tichý (see [7]). It was designed to provide a fine-grained semantics for natural language. Thus TIL is the system of a great expressive power applicable also and primarily to non-mathematical, i.e. empirical 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 has the capacity of capturing almost all the semantic features of natural language. It includes temporal and modal attitudes, epistemic logic, knowledge representation (even modeling knowledge of resource bounded agents) and dynamic aspects of these.

Here we are going to introduce TIL just briefly. For details see [1] and [2].

4.1 Basic Notions

The fundamental notion of TIL is the 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 of characters, and an interpretation is needed to reveal their meaning (semantics).

Constructions are not language expressions (strings of characters). They are abstract procedures, i.e., algorithmically structured objects. Since constructions are themselves semantic objects they do not need to be interpreted and they contain both semantic and syntactic components.

4.2 Types

Constructions are coded (and presented) in a language, which is formally derived from the language of typed lambda-calculus. However, terms of a lambda calculus are not constructions themselves; they are just forms of presentation of constructions. All the entities (including constructions) receive a type in TIL. Again, types are not strings of characters; rather they are objective collections of particular objects. For a type α we denote its elements 'α-objects'.

The infinite hierarchy of types in TIL arises from a *type base*. Type base is a (finite) set of basic *(atomic) types*. For the purpose of a natural language analysis, the standard type base of TIL is an epistemic base containing the following types:

o – *Truth values*. The type consisting of two elements: True (**T**) and False (**F**).
ι – *Individuals*. Simple, bare individuals: the 'lowest-level bearers of properties'.
τ – *Time points*. This type is just the set of real numbers.
ω – *Possible worlds*. The collection of all logically possible states of the world.

Over the basic atomic types *molecular types* are defined as the functional closure over the atomic types. The collection ($\alpha\, \beta_1 \ldots \beta_n$ of all (partial) functions mapping types $\beta_1, \ldots, \beta_n$ to a type α is a type. These types not involving constructions are called *types of order 1*. Since TIL constructions are objects *sui generis* and thus receive a type, we need to define higher-order types as well. However, first we define constructions.

4.3 Constructions

Constructions are the fundamental building blocks of TIL. Depending on valuation v, any construction $v-$constructs an object of some type, or it is v-improper (does not v-construct anything). TIL is a logic of partial functions. There are two kinds of constructions, *atomic* and *molecular*. Atomic constructions do not contain any other constituents but themselves. They supply objects on which compound constructions operate. There are two atomic constructions:

Trivialization ^0X is an elementary construction constructing the object Xa in the simplest way, without a mediation of any other construction.

Variable x is a construction ('x' is just a name). It constructs an object of a respective type dependently on a valuation v; it v-constructs.

Molecular (multiple-step) constructions are:[2]

Composition $[FC_1 \ldots C_n]$ is an instruction to apply a function to its arguments. If F v-constructs a function f of type $(\alpha\beta_1 \ldots \beta_n)$, and C_i v-construct objects c_i of type β_i, then the Composition v-constructs the value of f at the tuple-argument $\langle c_1, \ldots, c_n \rangle$. Otherwise the Composition is $v-$improper.

Closure $[\lambda x_1 \ldots x_n \quad C]$ is an instruction to construct a function in the manner of lambda calculi. If variables $x_1, \ldots, x_n$ range over $\beta_1, \ldots, \beta_n$, respectively, and C v-constructs an object of type α, Closure v-constructs the following function

[2] And two others, Execution and Double Execution, which we are not going to use here.

f: let v' be a valuation that associates x_i with b_i and is identical to 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.

4.4 Higher order types

Each construction is of some order. The order of a construction is the highest order of the type of objects constructed by sub-constructions of the given construction. Thus the basic type of order 2 is the type $*_1$ – the collection of all constructions of order 1 which v-construct non-constructional entities belonging to a type of order 1. The type $*_2$ is the type of constructions of order 2 $v-$constructing entities belonging to a type of order 2 or 1. And so on, *ad infinitum*. Any type of order n is also a type of order $n+1$ (type rising). Other types of order n are functional closures $(\alpha\beta_1 \ldots \beta_n)$ of defined types as specified in Section 4.2.

5 TIL-Script as a Content Language

Transparent Intensional Logic is a suitable logic system for utilization as a content language in multiagent systems. For this purpose its main advantages arise from the following TIL features:

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 almost any semantic feature of natural languages.

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

The TIL-Script language has been described in [8] and [9].

5.1 Ontologies for TIL-Script

OWL based Ontologies Any content language is tightly related to ontologies. All concepts used or mentioned by a content language must be defined in an ontology. And vice versa, the content language must be able to use any concept from the ontology. FIPA 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.

The recent trend is to use well-proven technologies of semantic web, in particular the OWL language, for defining ontologies. But the existing implementation tools for multi-agent systems do not support OWL very well. The way we have chosen for TIL-Script is to inter-connect the language with frame-like ontologies because of an implementation in Jade. Integration of OWL into TIL-Script is a subject of our recent research.

Concepts (or classes) of ontologies are sets of so-called individuals. We must not confuse these individuals with members of the TIL-Script type Indiv. Ontology individuals can be objects of any TIL-Script type. For TIL-Script this means that any ontology concept (class) which members are of type α is an object of type $(o\alpha)$, i.e. a set of α-objects. Ontology individuals (members of classes) are directly α-objects.

Inter-connection of TIL-Script with an ontology is mediated by the Trivialization construction. You may Trivialize any object or individual defined by the ontology. However, objects defined in the ontology cannot be Trivialised in TIL-Script. The only demand for an ontology to be used together with TIL-Script is that any class must have defined the TIL-Script type of its members.

GIS based Ontologies Geographic information systems (GIS) are generally used for gathering, analysis and visualization of information on the space aspects of real-world objects. The main advantage of GIS is the ability to relate different kind of information obtained from different sources of a spatial context. This feature enables agents to act in the real-world environment and make decisions based on its state. Agents usually dwell in the virtual world of a computer memory and they are not aware of their position, or of the objects and other agents around. GIS ontology enables agents to receive, manipulate and exchange spatial information.

Ontologies were developed to facilitate knowledge representation, sharing and exchanging. Spatial agents make use of geographic information as the input source for the decision-making mechanism. Situated agents are context-aware. They are aware of their position in space, actual speed, surrounding objects and relationships between them.

Agent actions are usually modelled by defining their behaviours. Each agent can perform the limited amount of predefined actions which can be combined into more complex behaviours. Creating ontology is then divided into two steps.

1. Identification and definition of agent behaviours.
2. Creating behaviour-specific ontology.

Agents can perceive their environment by sensors or geographic database and transform it into the symbolic representation of ontology.

Communication Reconstruction and Knowledge Base Summarizing the features introduced up to now, we can now present an agents' communication scheme based on FIPA standards using Knowledge base and TIL-Script as a content language. Figure 2 illustrates such a schema.

Example of a Knowledge Base:

Concept: "Nissan Skyline GTS-R"

Instance: *The* Nissan Skyline GTS-R, which I am parking outside.

In the TIL-Script language we have two separate ontologies. First, speech-act ontology is connected with the operative part of an agent. Agents are "born" with a minimal ontology in order to be able to communicate with its surroundings and learn by experience.

Second ontology that contains application domain is also replicated to an agent knowledge base. It instantiates its knowledge base and all concepts and roles are copied into it. It means that every agent has equal basic abilities at the time of their creation.

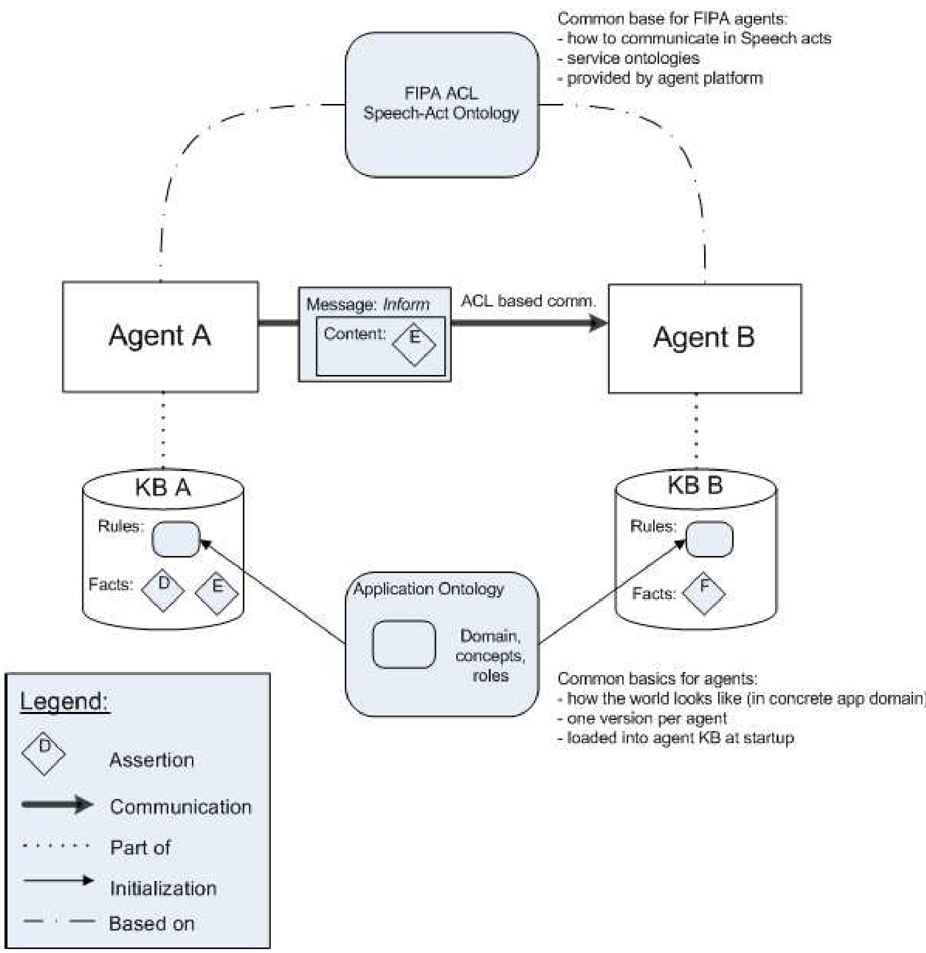

Fig. 2. FIPA Inform message with TIL-Script in the content field

6 Example

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

Scenario. The situation is simple. There is a large car parking-site, a railway station and two agents:

- Driver; an agent driving a car, who wants to park at the parking lot rather close to the railway station.
- Dis*patcher*; a dispatcher of the parking-site managing the pull-ins.

A sketch of their communication is as follows:

Driver: I want you to park me somewhere near to the railway station.

Dispatcher: OK, I can park you at this pull-in (concrete id).

Driver: I accept.

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

`TheDriver/Indiv` – The driver agent.

`TheDispatcher/Indiv` – The dispatcher agent.

`TheTrainStation/Indiv` – The train station the driver wants to park close to.

`Pull-in/((Bool Indiv)Time)World` – A property to be a pull-in at a car-park. The pull-in 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 a given time and state of the world).

`Arrange/(Bool (Indiv ((Bool)Time)World))` – This predicate means that an agent need to perform an action. In our conception, an agent (Indiv) has to arrange that a proposition ((Bool)Time)World)) become true.

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

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

Driver. Call for a proposal:

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

Dispatcher. Proposal:

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

Driver. I accept the proposal: (the content of the message is the same as in proposal, only the performative is “accept proposal”).

7 Conclusion

Actual standards for communication in multi-agent systems are based on syntax rather than semantics. This can slow down the progress in future research. As an option we propose the TIL-Script language, which is based on a well elaborated Transparent Intensional Logic. This makes TIL-Script a semantically based language suitable for communication in multi-agent systems.

High expressive power of TIL-Script makes it also a suitable tool for adopting other logics and languages into its 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 communication with human (non-expert) agents.

The TIL-Script language is being implemented and tested in multi-agent systems using the Python language and Jade based simulation programs. Ontology support is designed for frame-like ontologies supported by Jade. Using OWL ontologies supported by Protége has been developed and tested as a separate part of the research project. Storing ontologies into Knowledge Base is arranged with SQLite DataBase Management System (it behaves as a client application so that there is no centralized component in the system) with application layer implemented in Python (Jython).

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

References

1. Duží, M., Jespersen, B. and Muller, J. Epistemic closure and inferable knowledge. In Libor Běhounek and Marta Bílková, (Eds.), the *Logica Yearbook 2004*. Prague: Filosofia, 2005.
2. Duží, M. and Materna, P. *Constructions*. Retrievable at `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. Luck, M., McBurney, P., Shehory, O. and Willmott, S. Agent Technology: Computing as Interaction. A Roadmap for Agent-Based Computing. *AgentLink*, 2005.
6. Thomason, R. *Logic and artificial intelligence*. In: Edward N. Zalta, (Ed.): The Stanford Encyclopedia of Philosofy. Summer 2005.
7. Tichý, P. *The Foundations of Frege's Logic*. Walter de Gruyter, Berlin-New York, 1988.
8. Ciprich, N., Duží, M., Košinár, M.: *TIL-Script: Functional Programming Based on Transparent Intensional Logic*. In: *RASLAN 2007*, Sojka, P., Horák, A., (Eds.), Masaryk University Brno, 2007, pp. 37–42.
9. Ciprich, N., Duží, M. and 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.