American Journal of Applied Sciences 10 (10): 1298-1306, 2013
ISSN: 1546-9239

doi:10.3844/ajassp.2013.1298.1306 Published Online 10 (10) 2013 (http://www.thescipub.com/ajas.toc)

Understanding a Simple Arabic Stories Using Event Calculus

[1]Mohamed Abd El-Salam, [2]Ahmed Kamal Hussein and [2]Aly Aly Fahmy

[1]Department of Basic Science, Cairo University, Giza, Egypt
[2]Department of Computer Science, Cairo University, Giza, Egypt

Received 2013-07-07, Revised 2013-08-01; Accepted 2013-09-05

ABSTRACT

This study investigates the use of event calculus for understanding stories written in Arabic natural language. System makes a representation for commonsense knowledge and story events and then performs a commonsense reasoning to infer the details events which are not explicitly described in the story; furthermore it manages changes in objects properties over time and handles the actions synchronizations problem.

Keywords: Story Understanding, Animating Story, Event Calculus, Commonsense Reasoning, Reasoning about Action

1. INTRODUCTION

The field of story understanding is concerned with building computer programs that can understand stories. To date the field has dealt with short stories a few paragraphs long, rather than full-length novels, which is a much harder problem (Mueller, 2003a). The ability to understand stories has applications within many sorts of intelligent systems, such as advisory, dialogue, filtering, information retrieval, question answering and summarization systems (Mueller, 2004a).

The aim of the system is to produce a graphical model for describing the animation of the story.

There are five basic problems that must be addressed by any story understanding program:

- Representing the commonsense knowledge and the story events
- Inferring the details events, which are not explicitly described in the story, by performing a commonsense reasoning
- Managing the changes of properties of each object of the story over time
- Managing synchronizations between story events
- Determining the spatial positions of each object over time

Story understanding researchers have built computer programs for understanding stories. They have proposed a number of knowledge structures and processing mechanisms for this task. for example, story grammar (Rumelhart, 1975), demons as in story comprehension model program (Charniak, 1972), scripts (Schank and Abelson, 1977) as in SAM program (Cullingford, 1978), plans and goals knowledge structures (Wilensky, 1980) as in PAM program (Wilensky, 1981), frames structure as in comprehension model (Rosenberg, 1977), plot units as in BORIS (Kowalski and Sergot, 1986), subsymbolic representation as in DISCERN program (Miller and Shanahan, 2002), event calculus (Lehnert, 1982) as in Model-based story (Mueller, 2003b; 2004c; 2007).

The proposed system uses event calculus in the understanding process. Event calculus is formalism, based on first-order logic, for representing and reasoning about actions over time and their effects (Mueller, 2010).

It first presented by (Kowalski and Sergot, 1986). It was extended by (Miller and Shanahan, 2002). The event calculus is well-suited to the representation of the meaning of natural language and making inferences in

Corresponding Author: Mohamed Abd El-Salam, Department of Basic Science, Cairo University, Giza, Egypt Tel: +201017303108

natural language understanding systems. It is being used to represent tense and aspect and for inference in narrative comprehension systems, given a commonsense knowledge base about a domain and a narrative consisting of known properties and events in that domain, the event calculus can be used to fill in missing properties and events (Mueller, 2009). The system uses a classical logic axiomatization of the event calculus called the discrete event calculus DEC, which is equivalent to a standard axiomatization provided in a paper by (Miller and Shanahan, 2002). DEC restricts the time-point sort to the integers.

1.1. Action Module

Action module (**Fig. 1**) is the first module in the system. It performs a morphological analysis for each word in the story and then it makes a syntax analyzer to each sentence. After analyzing the sentence, it extracts information about subjects, verb and objects of the sentence to construct the story actions. The system deals with children stories for age 4-6 years old. The structure of stories for this stage should be very simple to be suitable for children to understand. Therefore, the syntax of the sentences usually is not complex. We chose fifteen of these stories, as a sample, to represent the corpus of the system. We made an analysis for these stories to extract verbs, nouns, propositions and adjectives. In addition, we analyze the content to specify all syntax used in it. All these data is entered to the system as startup knowledge. In addition, system is allowing users to extend verbs and nouns. Subsequently, the action module consists of three internal modules: morphological analyzer, syntax analyzer and semantic analyzer.

1.2. Morphological Analyzer

There are mainly two strategies for developing Arabic morphologies depending on the level of analysis: stem-based morphologies and root-based morphologies. Stem-based morphologies analysis is chosen for analyzing the stories due to its simplicity and straight forward implementation

1.3. Syntax Analyzer

In this study, we concentrate on studying the complexity of sentences structure of the selected stories to build a suitable parser; the following figure represents the context free grammar which is used:

<table>
<tr><td>< sentence ></td><td>::=</td><td>< sentence><term> | <term></td></tr>
<tr><td><term></td><td>::=</td><td>< name term> | <verb term> | <adverb term> |<prep term >|<conj term></td></tr>
<tr><td><noun term></td><td>::=</td><td><noun> <description></td></tr>
<tr><td><verb term></td><td>::=</td><td><verb> | < verb> + subject pronoun</td></tr>
<tr><td><adverb term></td><td>::=</td><td><adverb> definite بال | < adverb > name</td></tr>
<tr><td><prep term></td><td>::=</td><td><preposition> < noun term></td></tr>
<tr><td><conj term></td><td>::=</td><td><conjunction> <verb term> | <conjunction> <noun term></td></tr>
<tr><td><noun></td><td>::=</td><td>definite بال |name | <genitive> | indefinite | <pronoun></td></tr>
<tr><td><genitive></td><td>::=</td><td>Indefinite+pronoun | Indefinite Indefiniteبال | Indefinite name</td></tr>
<tr><td>< pronoun></td><td>::=</td><td>أنا | أنت | هو | هي | هما | هم | هن</td></tr>
<tr><td><description></td><td>::=</td><td>property <description> | name <description> | λ</td></tr>
</table>

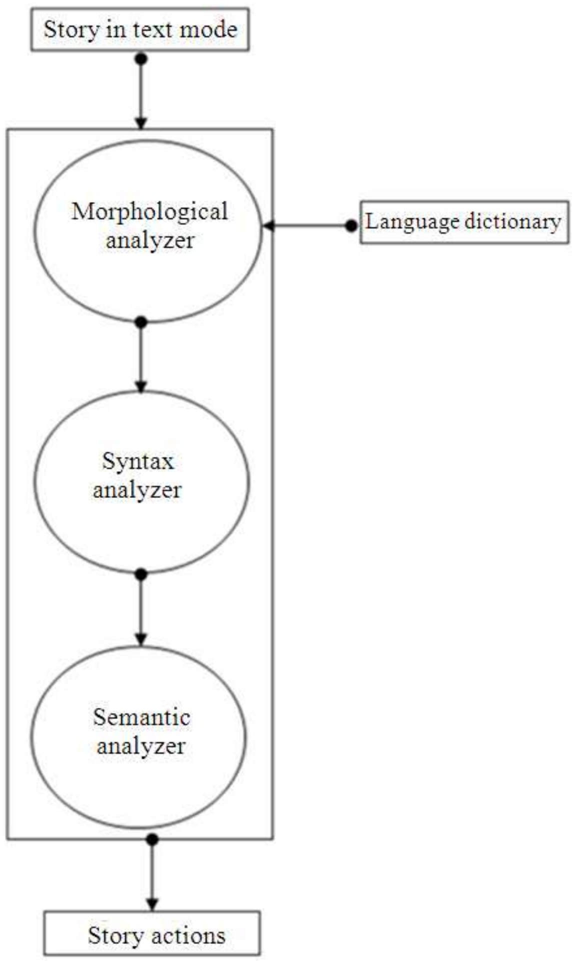

Fig. 1. Action module

1.4. Semantic Analyzer

Many problems arise during the process of semantic analysis, such as the hidden of subject or object, this section shows how system solves these problems.

1.5. Hidden Subject

ex: He played at the garden لعب في الحديقة

In this case the system considers that the subject of this sentence is the subject of the previous sentence.

1.6. Hidden Object

ex: The man played at the garden لعب الرجل في الحديقة

In this case the object term is empty. In order to describe this action the system considers the default description of the verb (play, لعب) which is the verb (run, جري).

1.7. Hidden Verb (Nominal Sentence)

ex: The man is strong الرجل قوي.

Note

This problem does not exist in English sentences. In this case the sentence describes the subject (الرجل), It have no action. So the system only add (strong, قوي) to the properties of the subject (the man, الرجل).

1.8. Subject Object Confusion

ex: He played the ball لعب الكرة
The man played لعب الرجل

Note

This problem does not exist in English sentences. In above example each sentence has the syntax terms (verb-noun). However, the noun is the subject in the first sentence and it is the object in the other one. The system solves this problem by determining the category (human, animal, mineral, adjective …) for each noun in the knowledge base.

IF noun is living being and no subject detected yet then it is subject:

ex: noun (the man, الرجل) in the following sentence:

The man played لعب الرجل

IF noun is living being and subject detected

Then it is object:

ex: noun (the man, الرجل) in the following sentence:

The dog bit the man عض كلب الرجل

IF noun is NOT living being and subject detected.

Then it is object:

ex: noun (the ball ,الكرة) in the following sentence:
The man played the ball لعب الرجل الكرة

IF noun is NOT living being and no subject detected yet AND verb can be done by noun
Then it is subject:

ex: noun (the fire, النار) in the following sentence

The fire burn the home حرقت النار البيت

5- Adjective confusion

Consider the following two sentences:

The man played the ball لعب الرجل الكرة
The big man played لعب الرجل الكبير.

In the previous sentences, each one has the syntax (verb-noun1-noun2). However, the third term, noun2, is the object of the first sentence and it is an adjective in the other one. The system solves this problem by allowing users to add more information about each noun in the knowledge base.

If noun is إسم فاعل-إسم مفعول OR description of noun is صفة

Then it is adjective else it is subject or object.

1.9. Understanding Module

Understanding module receives the actions of the story and commonsense knowledge and infers the details events, which are not explicitly described in the story. It consists of three internal modules, event calculus module, reasoning and module graphical module (**Fig. 2**).

1.10. Event Calculus Module

It converts natural language into event calculus form. Therefore, it converts both commonsense knowledge base and story into event calculus form.

1.11. Reasoning Module

It receives both commonsense knowledge base and story, in form of event calculus and produces a reasoning model, which describes the motion details of the story.

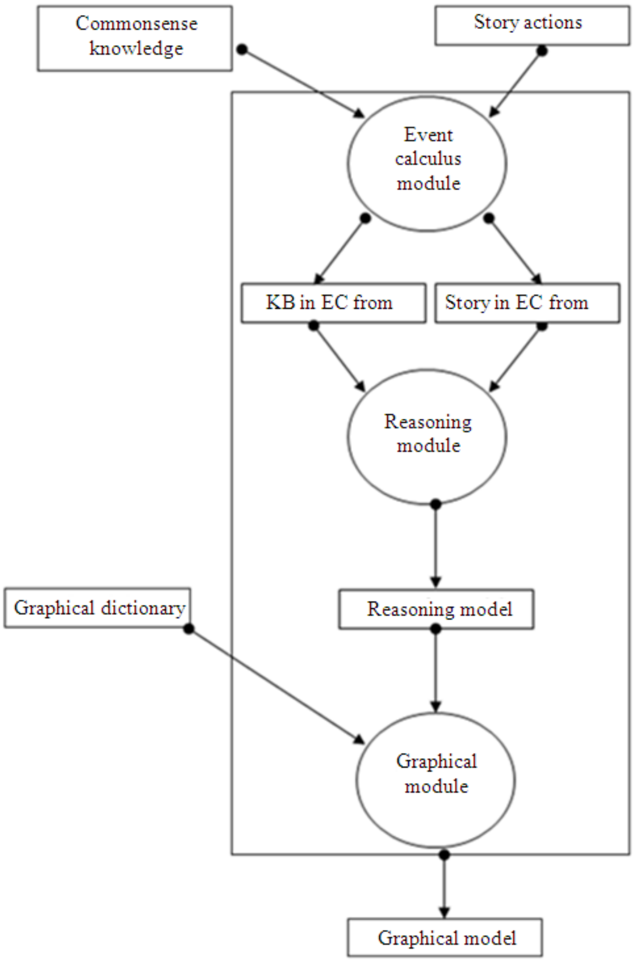

Fig. 2. Understanding module

1.12. Graphical Module

It translates the reasoning model into set of instructions to animate the story. This conversion is done according to the rules of the graphical dictionary.

The system needs a commonsense knowledge base which gives the necessary information facilitate the understanding processes. Commonsense knowledge base is written in a text file as sets of sentences. These sentences have specified structures according to the type of information it gives. After writing it, the system will convert it into event calculus form. Each set of information must falls into one of the flowing categories.

1.13. Event Description

It gives information about the details of each event in the story world. Therefore, a complex motion can be written as a sequence of simple events. For example, the following set of information describes the motion details of the event (person told someone statement):

(person) says to → (other) (words)	قال (الشخص) ل(الرجل) (قول)←
(person) look at (other) +	نظر (الشخص) إلي (الرجل) +
(person) say (words) +	قال (الشخص) (قول) +
(person) silent	سكت (الشخص)

1.14. Event Effects

It gives information about the side effect after execution of the event (not during the execution of the event). These effects usually change the values of system fluents. For example, the following set of information describes the effects of the event (person go to place):

(person) walk to (place)→	←(الشخص) إلي (مكان) مشي
(person) look at (other) +	+ (الشخص) ماشي
(person) is not stand +	+ (الشخص) ليس واقف
(person) is not sit down	(الشخص) ليس جالس

1.15. Startup Information

It gives information about the initial state of the system, i.e., the state of each object before executing the first event of the story:

(Ahmed) is sitting	(أحمد) جالس
(teacher) is standing	(المعلم) واقف

Commonsense Knowledge rule statement has some restrictions:

- Syntax is true and doesn't contain any pronouns or hidden subjects.
- Event description rule sentence takes form
- <verbal sentence> —> <verbal or nominal sentence> +
 <verbal or nominal sentence> + ...
- Event effects rule sentence takes form
o <verbal sentence> —> <subject> [<negation>] <properties>
o <properties> => <verb> | <adjective>
o <negation> => "ليس" | "ليست" | ""غير'' | "لا" (means not | no | never)
- Startup rule sentence takes form <subject> <properties>

Event calculus module converts commonsense knowledge base from text mode to event calculus mode. It depends on converting each term (sentence) to corresponding event calculus predicate. For example:

Happens ((سكت (الشخص)	(الشخص) سكت, time)
Happens ((مشي (الشخص, مكان)	(مشي (الشخص) إلي (مكان), time)
HoldsAt ((مشي (الشخص)	(الشخص) ماشي
ReleasedAt ((وقف (الشخص)	(الشخص) ليس واقف
Happens (Silent (person)	Silent (person), time)
Happens (Walk (person, place)	Walk (person, place), time)
HoldsAt (Walk (person))	Walk (person)
ReleasedAt (Stand (person))	Not Stand (person)

1.16. Choosing the Predicate Type Rules

- Verbal sentence is *Happens* predicate
- Nominal sentence is *HoldAt* predicate
- Negated nominal sentence is *ReleasedAt* predicate
  Writing events and fluents rules:

- Name of event, or fluent, is the verb source.
- In case of nominal sentence ex: ("(الشخص) ماشي ","the man is walking")
- The name of fluent is the source of the verb from the adjective, in this example ("مشي", walk)
- Parameters are subject and objects respectively

1.17. Graphical Dictionary

After converting the story into event calculus model, system uses this dictionary to translate this model into readable model, (graphical model). The idea of the dictionary that each verb in a story event may change some objects properties or make object do some actions, according to the graphical language which used (like Alice Language). So the dictionary explains the effects of each verb in the system. This dictionary can be changed by the user.

Example

Consider that, according to the graphical language of user, verb "وقف" property stand of the actor will be true and the actor will do action stand(). Also verb "أمسك" property hold of the actor will be true and the actor will do action hold(object) and so on. The following **Table 1** is an example of a graphical dictionary for these examples.

1.18. Graphical Output Model

It is the output of the system. It is an intermediate representation that is independent on any animation technologies. This model describes each scene of the story, the properties of each object at each scene and the action which happen at each scene. The system considers that the first scene is defined to the user, as a default state.

1.19. Story Scenes

The model is divided into senses. Each one consists of actors at certain situation and action (motion) which done at this scene.

1.20. Scene Action

It describes the action which happens in this scene. for example:

(Ahmed) opened the door   فتح (أحمد) الباب

1.21. Actors and objects properties

It describes the properties of actors and objects, consider the scene contains actor (Ahmed,أحمد) and object (door,باب). If (Ahmed,أحمد) open and (door,باب), then scene becomes:

Door : open	باب : فتح
Ahmed : stand	أحمد : وقف

Table 1. Graphical dictionary

	-------------Word---------------		-------------Property -----------		------------Function---------	
1	وقف	stand	وقف	stand	وقف	stand
2	جلس	Sit	جلس	Sit	جلس	Sit
3	أمسك	Hold	أمسك =	Hold =	أمسك	Hold
4	فتح	Open	فتح	Open	فتح	Open
5	قال	Say	تكلم	Speak	تكلم	Say
6	نظر	look	نظر	Look	نظر	Look
7	مشي	Walk	مشي	Walk	مشي	Walk
8	غلق	close	غلق	close	غلق	close
1	تكلم	Speak	تكلم	Speak	تكلم	Speak
2	جلس	Sit	جلس	Sit	جلس	Sit
3	سكت	Silent	----	----	سكت	Silent
4	فتح	Open	فتح	Open	فتح	Open
5	قال	Say	تكلم	Speak	تكلم	Say
6	نظر	look	نظر	Look	نظر	Look

Previous lines says that in scene, properties of (فتح) for (باب) becomes on and properties of (وقوف) for (أحمد) becomes on and other properties for actors are off.

Graphical output model contains two types of statements. Action description statement and actor properties statement.

Action description statement takes the form:

VERB = <verb> SUBJECT = <subject> OBJECT = <object>

Example

الفعل = فتح	الفاعل = أحمد	المفعول = باب
object = door	subject = Ahmed	verb = open

Actor properties statement takes the form:

<subject> : <property>

Example

Door: open	باب: فتح
Ahmed: stand	أحمد: وقف

Example

Consider the knowledge base contains following information:

(person) opened the door →	الباب مفتوح + الباب ليس مغلق
+ door opened +	فتح (الشخص) الباب ←
door not closed + (person)	(الشخص) ليس جالس +
standing+	(الشخص) واقف +
(person) not sitting+ (person) not walking	(الشخص) ليس ماشي

This information means that when any actor open object (باب), then:

- Property of (فتح, open) for (باب, door) become on
- Property of (غلق, close) for (باب, door) become off
- Property (وقف, stand) for the (actor) become on
- Property (جلس, sit) for the (actor) become on
- Property (مشي, walk) for the (actor) become on

Following example describes two of senses. They describe the world before and after actor, (أحمد, Ahmed), do action (فتح, open) for object (باب, door):

------------1------------	------------1------------
Door: close	باب: أغلق
Ahmed: stand	أحمد: وقف
-------------2------------	-------------2------------
Door: open	باب: فتح
Ahmed: stand	أحمد: وقف
Verb = open Subject = Ahmed	الفاعل = أحمد الفعل = فتح
Object = door	المفعول = باب

1.22. Characteristics of the Proposed System

Story understanding systems are characterized according to four dimensions (Mueller, 2004b); breadth of coverage, depth of understanding, ability to handle new stories, ability to handle real-world input. According to the these dimensions, the proposed system has following characteristics.

1.23. Breadth of Coverage

The system is a narrow coverage system that handles stories involving the commonsense knowledge base.

1.24. Depth of Understanding

The aim of understanding is to produce an animated story; therefore, we limited the scope of the depth understanding in this direction. System will study the details of objects actions and the changes of objects characteristics through the time.

1.25. Ability to Handle New Stories

This system is designed to handle unlimited number of new stories, in the range of the commonsense knowledge base.

1.26. Ability to Handle Real-World Input

The input is a free text story. In addition, user can easily update the commonsense knowledge base, but writing the commonsense knowledge base is submitted to some rules.

2. CONCLUSION

Event calculus is a good approach for representation commonsense knowledge and story events, also for performing commonsense reasoning. The proposed system uses that to infer the details of the events of the story and the changes of objects characteristics during each scene of the story. Also the system manages events synchronizations inside the sentence. It describes the actions of each object over the time and detects changes in objects properties and displays that in a readable model, graphical model. Also it can understand an unlimited number of stories, in range of commonsense knowledge base, also it allows users to, easily, write and change the knowledge base in text mode.

2.1. Limitations and Future Work

The problem of story understanding is a very hard. So we restrict the system with some limitations which can be treated in future work.

2.2. Actions Synchronizations

The System manages events synchronizations inside the sentence, when more than subjects do the same action at the same time. In other cases more than objects do different actions at the same time, to manage this situation we must keep in mind time of execution of each action and the speed of each object.

2.3. Traffics Problems

Since we talk about animation, we expect traffics problems during the motion of objects. To solve this problems the system must make interactions between the spatial positions of objects of the stories and objects of the world to determine spatial position of each object over time and to handle any problem during motions.

2.4. Working with Uncertainty

If information about uncertainty were added to input templates, the system could be extended to build and several alternative models for a story and then it will be able to determine when information cannot reliably be inferred from the story and when information from the story is speculative.

2.5. Complete Example

2.5.1. System Knowledge Base

1-	(Ahmed) sit	(أحمد) جالس
2-	(teacher) stand	(المعلم) واقف
3-	(person) say to (other) (words) (person) say + (person) silent	قال (الشخص) ل(الرجل) (قول) = (الشخص) قال (الشخص) (قول) + سكت
4-	(person) say person) speak	قال (الشخص) (قول) ←
5-	(person) silent →	(الشخص) يتكلم سكت (الشخص)←(الشخص) لا يتكلم

Corresponding event calculus form:

1-	(Ahmed) sit (stand (Ahmed), 1)	HoldsAt (1, (أحمد) وقف)
2-	HoldsAt (sit (teacher), 1)	HoldsAt (1, (لمعلم) جلس)
3-	Happens (say(person, words, other), time) = Happens (say(person, other), time) and	Happens (قال (الشخص,قول الرجل), time) = Happens (قال (الشخص , قول) time) and Happens (سكت (الشخص) time)
4-	Happens (silent(person), time)	
5-	Happens (say(person, words), time) → HoldsAt (speak (person), time) Happens (silent(person), time) → ReleasedAt (speak (person), time)	Happens (قال (الشخص,قول) time) → HoldsAt ( ) تكلم الشخص) time) Happens (سكت (الشخص), time) → ReleasedAt ( ) تكلم الشخص), time)

where, rules (1) and (2) are startup rules and (3) is action description rule and (4) and (5) are action affects rules.

Consider story script is:

قال (المعلم) ل (أحمد) "قف من فضلك"
(teacher) said to(Ahmed) "stand up please"

Corresponding event calculus, form rule (3), is:

Happens (قال (المعلم), "قف من فضلك", أحمد, time)
Happens (say(teacher,"stand up please",Ahmed), time)

Details of this event, from rule (3), are:

Happens (say(teacher, ,"stand up please"),1)	Happens (1) (قال (المعلم , " قف من فضلك")
Happens (silent(teacher),2)	Happens (2 , (سكت (المعلم)

Commonsense reasoning gives following model:

-------------1------------	-------------1------------
sit(Ahmed)	جلس(أحمد)
stand(teacher)	وقف(المعلم)
speak(teacher)	تكلم(المعلم)
Happens (say("stand up please"),1)	Happens(قال (المعلم " قف من فضلك") ,1)
------------2-------------	------------2-------------
(Ahmed) sit	جلس(أحمد)
(teacher) stand	وقف(المعلم)
Happens (silent(teacher,1)	Happens (سكت (المعلم) ,2)
------------3-------------	------------3-------------
(Ahmed) sit	جلس(أحمد)
(teacher) stand	وقف(المعلم)

2.6. Consider Dictionary is

2.6.1. Result Graphical Model is

-------------1------------	-------------1------------
Ahmed: sit	أحمد: جلس
teacher: stand	المعلم : وقف
teacher: speak	المعلم : تكلم
Verb=say	الفعل = تكلم الفاعل = المعلم
Subject=Ahmed	المفعول = "قف من فضلك"
Object="stand up please"	
------------2-------------	-------------2------------
Ahmed: sit	أحمد: جلس
teacher: stand	المعلم : وقف
Verb= silent	الفعل = سكت
Subject=Ahmed	الفاعل = المعلم
------------3-------------	-------------3------------

Ahmed: sit	أحمد: جلس
teacher: stand	المعلم : وقف

Notes

2.7. The Script

قال (المعلم) ل (أحمد) "قف من فضلك"
(teacher) said to(Ahmed) "stand up please"

Is converting according to rule (3) to the following actions:

Happens (قال (المعلم), "قف من فضلك", أحمد, time)
Happens (say(teacher,"stand up please", Ahmed), time)

So the result is three senses. At startup according to rule (1) and (2) and according to the dictionary the result is:

Ahmed: sit	أحمد: جلس
teacher: stand	المعلم: وقف

At first scene the action:

(teacher) said "stand up please"	قال المعلم "قف من فضلك"

Is done and according to rule (5) the one property is added:

teacher: speak	المعلم : تكلم

3. REFERENCES

Charniak, E., 1972. Toward a model of children's story comprehension. Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Cullingford, R.E., 1978. Script Application: Computer Understanding of Newspaper Stories. 1st Edn., NTIS, Springfield.

Kowalski, R. and M. Sergot, 1986. A logic-based calculus of events. New Generat. Comput., 4: 67-95. DOI: 10.1007/BF03037383

Lehnert, W.G., 1982. Plot Units: A Narrative Summarization Strategy. In: Strategies for Natural Language Processing, Lehnert, W.G. and M.H. Ringle (Eds.). Lawrence Erlbaum Associates, Hillsdale, ISBN-10: 0898591910, pp: 375-412.

Miller, R. and M. Shanahan, 2002. Some Alternative Formulations of the Event Calculus. In: Computational Logic: Logic programming and Beyond: Essays in Honour of Robert A. Kowalski, Kakas, A.C., F. Sadri and R.A. Kowalski (Eds.), Springer, Berlin, ISBN-10: 3540439595, pp: 452-490.

Mueller, E.T., 2003a. Story Understanding. In: Encyclopedia of Cognitive Science, Nadel, L. (Ed.), Nature Publishing Group, London, ISBN-10: 0333792610, pp: 38-46.

Mueller, E.T., 2003b. Story understanding through multi-representation model construction. Proceedings of the HLT-NAACL Workshop on Text Meaning, (WTM' 03), Association for Computational Linguistics Stroudsburg, PA, USA., pp: 46-53. DOI: 10.3115/1119239.1119246

Mueller, E.T., 2004a. Event calculus reasoning through satisfiability. J. Logic Comput., 14: 703-30. DOI: 10.1093/logcom/14.5.703

Mueller, E.T., 2004b. Understanding script-based stories using commonsense reasoning. Cognitive Syst. Res., 5: 307-340. DOI: 10.1016/j.cogsys.2004.06.001

Mueller, E.T., 2004c. A tool for satisfiability-based commonsense reasoning in the event calculus. Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference, (RSC' 04), Miami Beach, Florida, USA., pp: 147-52.

Mueller, E.T., 2007. Modelling space and time in narratives about restaurants. Literary Linguistic Comput., 22: 67-84. DOI: 10.1093/llc/fql014

Mueller, E.T., 2009. Automating commonsense reasoning using the event calculus. Commun. ACM, 52: 113-117. DOI: 10.1145/1435417.1435443

Mueller, E.T., 2010. Commonsense Reasoning. 1st Edn., Morgan Kaufmann, Amsterdam, ISBN-10: 0080476619, pp: 432.

Rosenberg, S.T., 1977. Frame-based text processing. Artificial Intelligence Laboratory.

Rumelhart, D.E., 1975. Notes on a Schema for Stories. In: Representation and Understanding: Studies in Cognitive Science, Bobrow, D.G. and A.M. Collins (Eds.), Academic Press, New York, ISBN-10: 0121085503, pp: 211-236.

Schank, R.C. and R.P. Abelson, 1977. Scripts, Plans, Goals and Understanding: An Inquiry Into Human Knowledge Structures. 1st Edn., Lawrence Erlbaum, Hillsdale, ISBN-10: 0470990333, pp: 248.

Wilensky, R., 1980. Understanding Goal-Based Stories. Ph.D. Thesis, Yale University.

Wilensky, R., 1981. PAM and Micro PAM. In: Inside Computer Understanding, Schank, R.C. and C.K. Riesbeck (Eds.), Erlbaum, Hillsdale, NJ., ISBN-10: 089859071X.