AWERProcedia Information Technology & Computer Science

Vol 03 (2013) 889-894
3rd World Conference on Information Technology (WCIT-2012)

Thinking-Understanding approach in IT maintenance domain automation

Maxim Talanov *, Balkish 5, Kazan 420061, Russia.
Alexander Toshchev, Belomorskaya st 6-92, Kazan 420133, Russia.
Andrey Krehov, Balkish 5, Kazan 420061, Russia.
Dr. Airat Khasianovd, Kremlevskaya 18,Kazan 420008, Russia.

Suggested Citation:
Talanov, M., Toshchev, A., Krehov, A. & Khasianovd A. Thinking-Understanding approach in IT maintenance domain automation, *AWERProcedia Information Technology & Computer Science*. [Online]. 2013, 3, pp 889-894. Available from: http://www.world-education-center.org/index.php/P-ITCS. *Proceedings of 3rd World Conference on Information Technology* (WCIT-2012), 14-16 November 2012, University of Barcelon, Barcelona, Spain.

Received 1 April, 2013; revised 3 June, 2013; accepted 6 August, 2013.
Selection and peer review under responsibility of Prof. Dr. Hafize Keser.

Abstract
There is are a lot of applications in Software Engineering already created and most of them require maintenance of an IT personal, for example: bug fixing, change requests, etc (according to statistics). There are numbers of companies that work in IT outsourcing business model providing maintenance for software of third party companies. There are also a huge number of IT specialists that fix bugs every day or help user as IT support specialists (for example: remote software installation). In other words day-by-day they perform routine tasks. In this paper, we explore the idea of incident processing automation: starting with processing requests in natural language and ending with incident resolution. Our goal is to implement this automation approach using machine understanding mechanism. We like to reuse open-source solutions in our project, as much as we can. We use Minsky's thinking model, from his book The Emotion machine[1]. Natural language processor is the RelEx .

Keywords: Intelligence, Natural Language Processing, Automation, Software Development, Automation;

* ADDRESS FOR CORRESPONDENCE: **Maxim Talanov,** Balkish 5, Kazan 420061, Russia, *E-mail address:* max.talanov@gmail.com / Tel.: +79172731327

1. Introduction

In 2005 MIT has published prototype MIT Metaphor [2], that generates classes in Python Language based on description in shallow English language. In 2007 out first prototype was brought to light and demonstrated ability to modify existing web-application based on change request in shallow English.

Nomenclature

Company	outsource Company that provides IT support service
NLP	Natural Language Processing
WayToThink	Way to think or how we think (Minsky's terminology)
CryForHelp	A kind of WayToThink
TU	Thinking Understanding concept approach

1.1. The Problem

There are a lot of Software applications already created and being on market for decades and most of them require maintenance of an IT personal, for example: bug fixing, change requests, etc. Many and many companies around the world, that works under IT outsourcing business model providing maintenance for software of third party vendor's companies and its clients. Thousands of IT specialists that fix bugs every day or help users as IT support specialists (for example: remote software installation) do that on a daily basis for many years. In other words day-by-day they perform routine tasks.

The problem is that human's work is a high cost resource. So, automation of it can free up many resources in companies for a new work.

1.2. The Statistics

According to statistics approx. 60% incidents can be automated or partially automated. We use Fujitsu GDC Russia* statistics.

Example: Incident: "It seems to be user is missing Abode Flash Player on his computer". Operator should just install Adobe Flash Player. This incident processing could be easily automated..

Moreover, a part of incidents are automatically generated that can be parsed without Natural Language Processing

* Global Delivery Centre Russia (Fujitsu Services Ltd.), http://www.icl.ru/pages/968, A part of http://www.fujitsu.com/fts/

2. Materials and methods

System is based on Marvin Minsky's approach in Artificial Intelligence called "The Emotion Machine" [1]. Different approaches in Artificial Intelligence such as Artificial Intelligence "A Modern Approach" [3] has been studied. Minsky's thinking model for machine understanding is used, as suppose that human understanding is based on thinking as model.

2.1. Emotion Machine in action

According to Minsky's approach of thinking model there is a triple Critics->Selector->WayToThink base actors.

Critics are probabilistic predicates, they are reacting when specific rule is satisfied and activate the Selector via specific Selector request. Critics are used heavily in analysis of inbound information, tracing current state of the system, tracing intermediate results in context.

There are several types of Critic:

- Manager – simply triggers proper Way2Think via Selector request.
- Control – tracks the parameters of the system, ex.: time
- Analyser – Critic with complex predicate used over current context.

Selector is an object that retrieves resource required for processing and finally return Resource: Way2Think or Critic.

WayToThink is a how thinking process can be done. For example, one of a possible WayToThink is CryForHelp, which means "Ask somebody for help". On the Fig1 there is interaction process in triple: Critics->Selector->WayToThink.

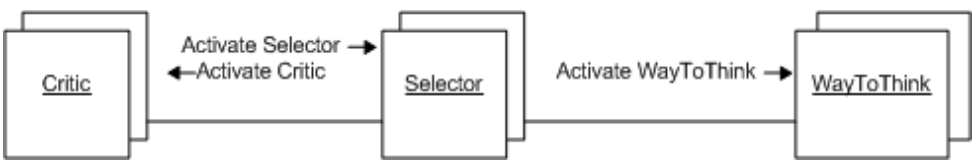

Figure. 1. Critics->Selector->WayToThink collaboration

In case of described system WayToThink modifies data or initiates End User Interaction

In IT Domain there are different classes of Incidents:

- Direct instructions (for example, "Please install Firefox")
- Problem with desired state (for example, "User has IE7 installed, but IE8 required")
- Generated incidents (for example, User %username% hasn't receive LOT 123)
- etc.

Please note that this is not final classification and can be extended dynamically. Critics are used to process those incidents.

In general, extended "Emotion machine" approach with Critics->Selector->Way2Think triple on It Support Domain is used.

However, base approach was applied to ITSM in real world with SLA and ITIL* foundation methodologies.

2.2. General overview

TU system is comprised of operational units such Training subsystem, Operating subsystem and Knowledge Base. TU Operating subsystem is integrated with target(customer) system via smart operating mechanism described below.

Training subsystem is capable of machine learning of a system of domain and common sense knowledge. Domain information is: problem symptoms, possible solutions, problem solution time limits (SLAs), main domain concepts like software, network, internet, browser, data base table etc.

Operating subsystem is capable of understanding problem descriptions and retrieving proper solution, clarifying problem description and processing such clarification, implementing found solution over target(customer) system via smart operating mechanism.

Knowledge base is the main storage that contains all data and is capable of reporting of every action performed/executed by the system.

Figure 2 shows system general overview diagram.

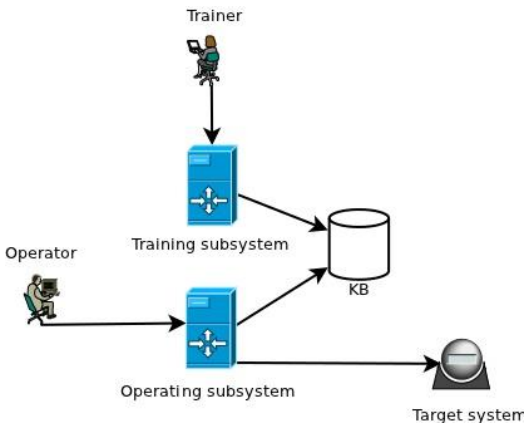

Figure 2. General system overview

* http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library Information Technology Infrastructure Library

2.3. K-Line

According to Minsky concept data in our Knowledge Base is stored as a K-Line. For example, there is a problem and it's context (I do not know how to eat soup with a spoon). When problem solved our mind creates connection between problem and context. (In our case we connect algorithm how – to eat soup with spoon).

2.4. Annotate Request

Initial annotating is performed using Knowledge Base, for example, sometimes NLP interprets "please" as an action, but actually this is form of politeness. This fact is known by us and stored in Knowledge Base, so after annotation number of errors during NLP processing is reduced.

Sometime requests contain errors semantic or linguistic. In this case preprocessing and correcting these errors using Google correction API *for spelling errors and After The Deadline †for semantic corrections are taken place.

2.5. Simulation

Simulation creates problem description model using domain model concepts.

Input is annotated narrative of problem description. Annotated narrative is actually a sequence of phrases with concepts annotating each of phrases. These input concepts are mapped to concepts of the domain model or trained domain data.

Each concept of problem description is searched in domain model, than each link of problem description is verified in domain model, most probable links are chosen. Default concepts are added to problem description simulated model.

Output of simulation is Semantic network of problem description model translated in domain model concepts.

2.6. Find and apply solution

Every conception of formalized request is mapped to special knowledge called how-to. How-to is the some kind of receipt, in other words: "how-to do something". For example, we have a "install software" conception. It is mapped on the special how-to "install software" that depend on several parameters like "user id", "workstation id", e.t.c. When request is formalized system tries to find suitable Solution for this request. Actually, solution is a set of How-to. In real word solution is a set of scripts that will be executed on the target client machines.

2.7. Smart solution applicator

Applying of the solution is not trivial task. So, Solution Applicator is a special component for this purpose. When system applies request and apply is failed, it tries another way and assign score for the success way. After some time system will be able to select most probable how-to for solution.

* See http://code.google.com/p/google-api-spelling-java/ for more information
† See http://afterthedeadline.com/ for more information

3. End user communication

When system can't find solution or encountered another problem it requires End User communication for problem solving. There is a special way2think for this action called "Cry4Help". This wayToThink interacts with human specialist whom will provide missing information for the system.

For example, when system can't formalize request selector activates "Cry4Help" WayToThink. System asks human specialists for assistance in problem. Human specialists should append data to Knowledge Base, so system can resolve problem.

4. Training component

System can perform action on 2 different ways: operational and training mode. In training mode trainer provides missing information for the system. All how-to can be exported to work instructions for the human specialist.

5. Results

At this point system can process 61% of Incidents, but it still evaluating. For future we are planning to cover at least 80% of incidents.

6. Conclusion and Recommendations

We suggest to continue develop the system to improve results. Also we are planning to contribute in NLP (RelEx) to improve processing of natural language and add new languages support.

However, application of Artificial Intelligence is our main goal and we are planning to extend approach on different domain not IT support only.

7. Acknowledgements

We are gratefully thank to Marvin Minsky for his book "Emotion Machine"[1] and MIT for inspiring "MIT Metafor"[2]

References

[1] Marvin Minsky. The art Emotion Machine. Simon & Schuster Paperbacks 2006.
[2] Hugo Liu, Henry Lieberman. Metafor: Visualizing Stories as Code, Cambridge, MIT Media Laboratory 2005
[3] Stuart Russell, Peter Norvig, Artificial Intelligence. A Modern approach, Pearson 2010