Natural language interfaces of next-generation intelligent computer systems

Artem Goylo
Minsk State Linguistic University
Minsk, Belarus
Email: artemgoylo@gmail.com

Sergei Nikiforov
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
Email: nikiforov.sergei.al@gmail.com

***Abstract*—The article describes an approach to the implementation of natural language interfaces of next-generation intelligent computer systems built using OSTIS technology, and also proposes a dialogue context model. In this approach, all stages of analysis, including lexical, syntactic and semantic analysis, can be performed directly in the knowledge base of such a system. This approach will effectively solve such problems as managing the global and local contexts of dialogue, as well as resolving linguistic phenomena such as anaphora, homonymy and tackling elliptical phrases.**

***Keywords*—Natural Language Processing, Natural Language Understanding, ontology, context, semantic network, Open Semantic Technology for Intelligent Systems (OSTIS), SC-code (Semantic Computer Code), constituency grammar**

I. Introduction

Currently, there is a large number of different interfaces of computer systems, which complicates interoperability between such systems and people as they need to familiarize themselves with the interface of each new system, which is rarely intuitive.

One of the main features of next-generation intelligent computer systems should be a user interface that can provide effective user interaction with the system, considering that users are often not professionally trained.

Speech is one of the most natural and convenient forms of information transfer between people, which leads to the increasing spread of natural language interfaces [1]. At the present time, no one doubts that this form of human-machine interaction plays and will continue to play a significant role in interaction with various computer systems.

However, it should be noted that a great diversity of languages (both natural and artificial) leads to the need to simplify the process of creating such interfaces for each individual language.

II. State of the art

Most approaches to natural language processing and understanding are based on machine learning [2], [3]. Undoubtedly, for most widely used languages, natural language processing models work very well and are improving every day, but despite the success in this area, this approach has several disadvantages:

- problems when working with different domains, for example, the meanings of words or sentences can be different depending on the subject domain. Thus, models for NLP may work well for a particular subject domain, but not be suitable for general application [4];
- creating a new model requires a large amount of data, and the quality of such data directly affects the quality of the resulting model, which leads to high costs for its training [5] [6];
- the model data is a "black box" because such models do not have the means to provide a description of its inference;
- every such model solves only a small amount of problems, there is no general approach to natural language processing. [4]

These shortcomings of the methods used cause some of the shortcomings of modern systems that implement a natural language interface. For example, despite the fact that now there is a large number of speech assistants created by different companies [7], [8], [9], [10], they have similar drawbacks, namely, an exclusively distributed implementation, due to end-user device performance being insufficient to run resource-intensive models. This in turn leads to privacy issues [11].

The speech understanding submodule of these systems generates a construction that reflects the meaning of the message using a frame model. A simplified example of such a construction is shown in figure 1.

```
{
  "text": "how many people between Tuesday and Friday",
  "intents": [
    { "name": "inquiry"}
  ],
  "entities": {
    "metric": [
      { "role": "metric", "value": "metric_visitor"}
    ],
    "datetime": [
      { "role": "datetime",  "type": "interval",
        "from": { "grain": "day", "value": "2020-05-05T00:00:00.000-07:00" },
        "to": { "grain": "day", "value": "2020-05-09T00:00:00.000-07:00" }
      }
    ]
  }
}
```

Figure 1. Message meaning formalization example

At the same time, other formats are used to present the results of intermediate stages of processing, the modules that implement them do not have any single foundation and interact through specialized software interfaces between them, which leads to incompatibility of the methods for presenting results at various stages of processing and the final result of text processing. This incompatibility, in turn, leads to significant overhead costs in the development of such a system and, in particular, in its modification.

As a solution to the compatibility problem, it is proposed to use an approach to natural language processing based on its formal model in the form of a set of ontologies formed using universal knowledge representation tools. This will contribute to interoperability of the component of natural language processing as a whole with other components of the system, and between parts of the component itself.

The aim of the article is to design an interface model based on an approach to natural language processing that uses ontologies containing a formal description of natural language.

III. Suggested approach

In the suggested approach to the implementation of natural language interfaces, it is proposed to carry out all stages of analysis, including lexical, syntactic and semantic analysis directly in the knowledge base of an intelligent system, presenting the results in a single unified form.

The description of the results of lexical, syntactic and semantic analysis is supposed to be carried out on the basis of the concepts introduced in the following subject domains:

- *Subject domain of the lexicon of natural languages*;
- *Subject domain of syntax of natural languages*;
- *Subject domain of denotational semantics of natural languages*.

However, the specification of these subject domains is not the aim of this article.

We also suggest to introduce a set of concepts to describe the context of the dialogue at various levels. The presence of such contexts will allow storing and using not only the history of the dialog (including the meaning of messages), but also other knowledge that can be used in the course of the dialog, including heterogeneous information about the interlocutor.

We propose to use the representation of knowledge about different languages (including knowledge about their syntax and semantics) in a unified form. This will significantly reduce overhead costs in the development of various systems that use the created ontologies.

In this arcticle it is proposed to base natural language interfaces on *OSTIS Technology* [12]. This technology allows to ensure the compatibility of heterogeneous problem solving models, and reduce the costs of development and modification (including adding a new problem solving model to the system).

Systems developed on the basis of the OSTIS Technology are called ostis-systems. The OSTIS Technology is based on a universal way of semantic representation of information in the memory of intelligent computer systems, called *SC-code*. SC-code texts are unified semantic networks with a basic set-theoretic interpretation. The elements of such semantic networks are called *sc-elements* (*sc-nodes* and *sc-connectors*, which, in turn, depending on their directivity, can be *sc-arcs* or textitsc-edges). *SC-code alphabet* consists of five main elements, on the basis of which SC-code constructions of any complexity are built, including the introduction of more specific types of sc-elements (for example, new concepts). The memory that stores SC-code constructions is called semantic memory or *sc-memory*.

Fragments (substructures) of the subject domains and ontologies under consideration, as well as structures related to the knowledge base and problem solver models, will be further shown in the form of SC-code texts (sc-texts).

A *problem solver* of any ostis system (more precisely, the sc-model of the problem solver of an ostis-system) is a hierarchical system of knowledge processing agents in semantic memory (*sc-agents*) that interact with each other exclusively by specifying their actions in the memory [13].

A system of sc-agents over a shared memory is a collective of sc-agents the initiation condition of which is the appearance of a certain construction in the system's memory. In this case, operations interact with each other through the system memory by generating constructions that serve as initiation conditions for another operation.

With this approach, it becomes possible to ensure the flexibility and extensibility of the system functionality by adding or removing a certain set of agents from its composition, without making changes that affect other agents.

A. Subject domain and ontology of natural language interfaces of ostis-systems

Natural language interface – SILK-interface (Speech, Image, Language, Knowledge) – is an interface where the exchange of information between the computer system and the user occurs through dialogue. The dialogue is conducted in one of the natural languages.

natural language interface
$\supset$ *speech interface*

Speech Interface is a SILK interface where information is exchanged through dialogue, during which the computer system and the user communicate using speech. This type of interface is closest to natural communication between people.

In the suggested approach, the following stages of natural language processing can be distinguished:

- lexical analysis;
- syntactic analysis;
- message comprehension.

In turn, lexical analysis includes decomposition of the text into tokens and their mapping to lexemes.

Understanding the message comes down to generating message meaning variants and choosing the correct one based on the context, as well as merging it with this context.

Provided below is the structure of the natural language interface problem solver.

Natural language interface problem solver
$\Rightarrow$ *abstract sc-agent decomposition**:

{• *Abstract sc-agent of lexical analysis*
⇒ *abstract sc-agent decomposition**:
{• *Abstract sc-agent for decomposing text into tokens*
• *Abstract sc-agent for mapping tokens to lexemes*
}
• *Abstract sc-agent of syntactic analysis*
• *Abstract sc-agent of message understanding*
}

In turn, *abstract sc-agent of message understanding* is decomposed into:

Message understanding agent
⇒ *abstract sc-agent decomposition**:
{• *Abstract sc-agent for generating message meaning variants*
• *Abstract sc-agent for context selection and update*
⇒ *abstract sc-agent decomposition**:
{• *Abstract sc-agent of context resolution*
• *Abstract sc-agent for choosing the meaning of a message based on the context*
• *Abstract sc-agent for embedding a message into a context*
}
}

The knowledge base must contain a specification of each agent, an example of a fragment of such a specification is shown in figure 2.

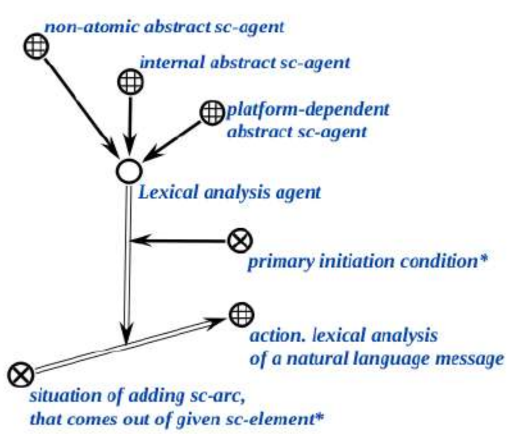

Figure 2. Agent specification example

B. Subject domain and ontology of lexical analysis of natural language messages included in an ostis-system

action. lexical analysis of a natural language message
⇒ *generalized decomposition**:
{• *action. decomposition of text into tokens*
• *action. mapping tokens to lexemes*
}

From the point of view of an ostis-system, any natural language text is a *file* (i.e. an SC-node with content).

The stage of lexical analysis is the decomposition of the text into a sequence of tokens and the mapping of lexemes to the tokens resulting from this decomposition. It should be noted that these tokens, if necessary, can be compared not with lexemes, but with their subsets included in its morphological paradigm that correspond to certain grammatical categories: case, number, gender, etc.

A result of lexical analysis is shown in figure 3.

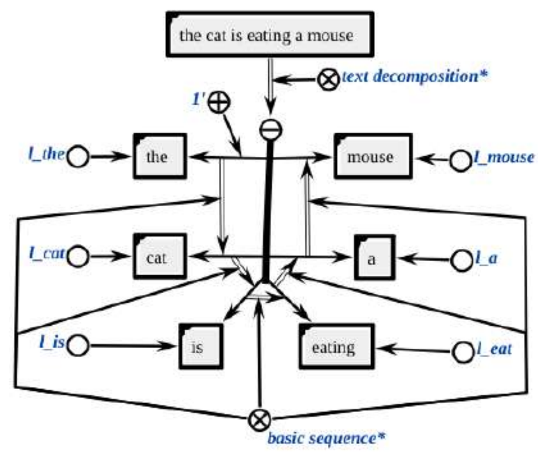

Figure 3. Lexical analysis result example.

To perform lexical analysis, the knowledge base of the system must also contain a lexicon with lexemes and their various word forms.

A lexeme is a unit of the lexicon of a language, a set of all forms of a certain word. An example of a lexeme specification in the knowledge base is shown in figure 4.

C. Subject domain and ontology of syntactic analysis of natural language messages included in an ostis-system

The agent of syntactic analysis performs the translation of the tokenized text into its syntactic structure. At the same time, due to the impossibility of resolving structural ambiguity at the stage of syntactic analysis, its result of syntactic analysis will generally be a set of potential syntactic structures.

An example of a syntactic structure is shown in figure 5.

D. Subject domain and ontology of understanding natural language messages included in an ostis-system

действие. понимание естественно-языкового сообщения

action. natural language message understanding

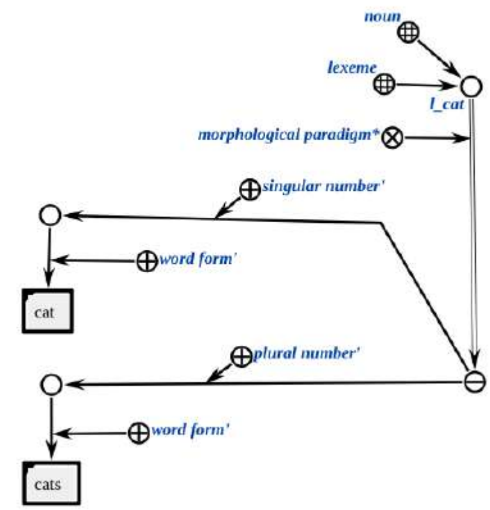

Figure 4. Example of a lexeme in the knowledge base.

⇒ *generalized decomposition**:
{• *action. generation of message meaning variants*
• *action. context selection and update*
  ⇒ *generalized decomposition**:
  {• *action. context resolution*
  • *action. selecting the meaning of the message based on the context*
  • *action. embedding the message in context*
  }
}

Action. message meaning variants deneration is an action during which the formation of a strict disjunction of potentially equivalent structures is carried out.

*Potentially equivalent structure** is a binary oriented relation that connects a structure and a set of structures that could potentially be equivalent to it, however, additional steps are required to reliably determine this.

At the same time, the transition from the result of syntactic analysis to structures potentially equivalent to the message is carried out according to the rules contained in the subject domain of denotational semantics. An example of one of the rules is shown in figure 6.

As a result of this action, a structure is formed in the knowledge base that describes possible variants of the meaning of the message, an example of such a structure based on constituency grammar [14] is given in figure 7. The presence of several such structures is explained by the fact that, in general, several variants of the syntactic structure are generated at the stage of syntactic analysis. The choice of the correct message meaning will be made in the course of the subsequent steps.

It should be noted that, if necessary, the meaning of the message can be generated not only on the basis of its syntactic structure based on constituency grammar, but also on other knowledge about this message, for example, subject-relation-object triples extracted from the text of this message, the result of message classification, etc.

Further steps in the message understanding process are performed based on the context.

Context is as sc-structure containing the knowledge used by the system during one or more dialogs. Generally, this knowledge includes both previously provided in the knowledge base and obtained in the course of operating sensors and / or communicating during a dialog.

dialog context
⊂ *context*
⇒ *subdividing**:
  Typology of dialog contexts by scope^
  = {• *thematic context*
  • *user context*
  • *global context*
  }

Thematic context is a dialog context containing topic-specific information (information obtained during the dialog on a certain topic, for example, when talking about a certain set of entities).

*A set of thematic contexts of a dialog** is a binary oriented relation, a dialog with the oriented set of its thematic contexts.

User context is a dialog context that contains user-specific information that can be used in a dialog with them on any topic. In general, user context intersects with *the approved part of the KB* (reliable information about the user previously provided in the KB that has passed the necessary moderation), but is not included in it entirely (the part received during the dialog that we are not sure about). An example of intersection of different types of contexts with an approved part of the knowledge base is shown in figure 8.

Global context is a dialog context that contains information that may be needed when conducting a dialog with any user. Global context is a subset of the approved part of the knowledge base that contains the information that may be used in the dialogue. For example, in a dialog with a specific user, it is unnecessary to use:

- proprietary information located in the knowledge base, which is necessary for the system to work but not intended for use in the dialog;
- parts of user contexts of other users.

dialog context
⇒ *subdividing**:
  Context typology by knowledge validity period^
  = {• *dialog context that does not change during system operation*
  • *dialog context that changes during system operation*
  }

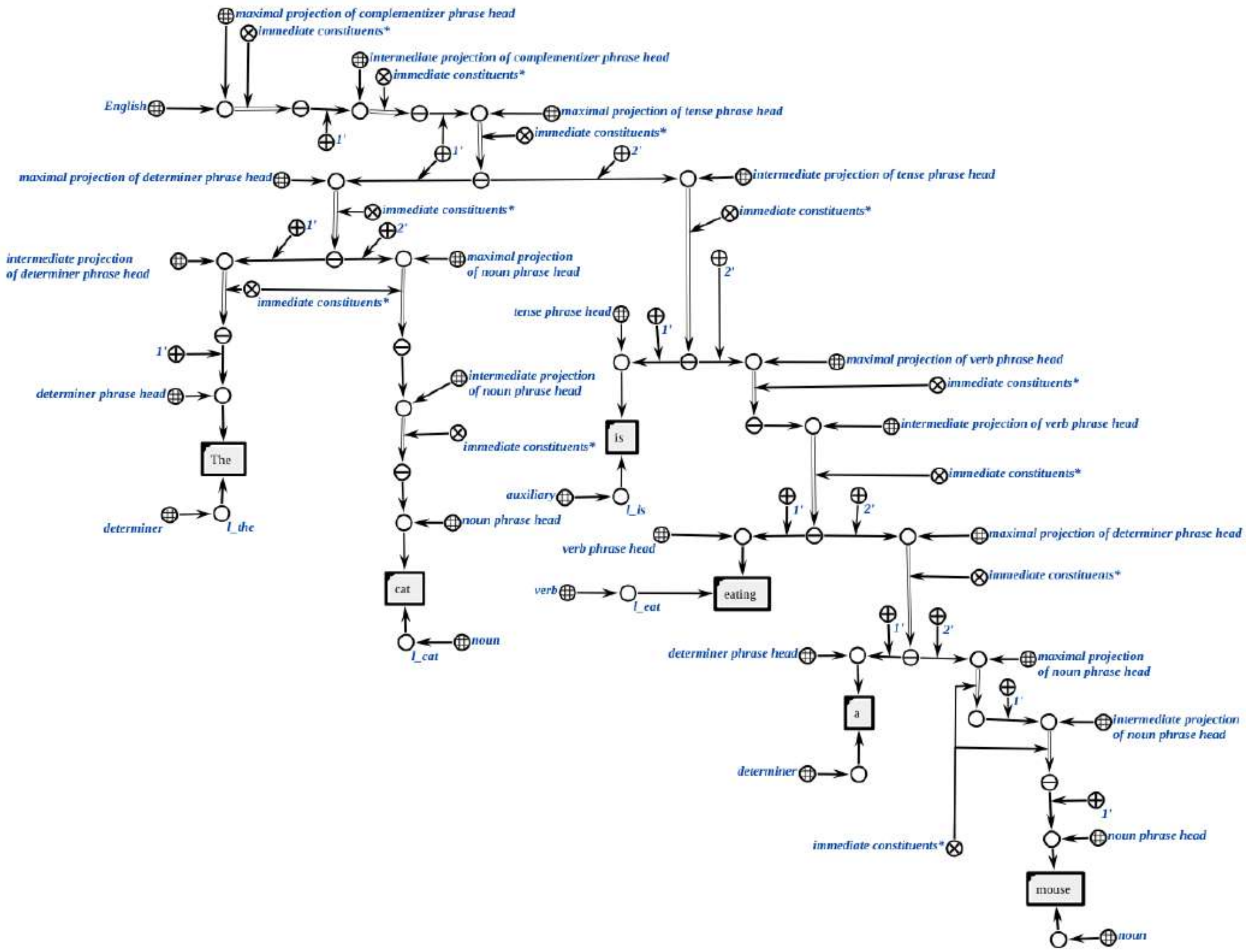

Figure 5. Syntactic structure example.

Dialogue context that does not change during system operation contains the knowledge necessary to ensure that the system performs its functions, which were put into it a priori by its developers and/or administrators and do not change during its operation on an ongoing basis.

Dialogue context that changes during system operation contains the knowledge necessary for the system to perform its functions, which were obtained during its operation and/or the validity of which is short-lived.

dialog context that changes during system operation
⇒ *subdividing**:
Typology of contexts that change during system operation knowledge source^
= {
- *dialog context containing knowledge from external sources*
- *dialog context containing knowledge gained during a dialog*

}

⇒ *subdividing**:
Typology of changing contexts according to their validity^
= {
- *valid dialog context*
- *invalid dialog context*

}

A subset of the context can be included in the approved part of the KB, for example, if we are talking about some biographical information previously entered in the KB - date of birth, etc.

At a given moment, one user dialog context is associated with a user (containing at minimum facts about them known in advance: name, age, etc.) and several thematic contexts. A context specification example is shown in figure 9.

Thus, **action. context resolution** is reduced to mapping each meaning variant to the corresponding context. The choice is made on the basis of the value of the function $F_{CTD}(T, C)$, where T is a translation variant, C is a thematic context. A suitable context for a translation variant is the one for which

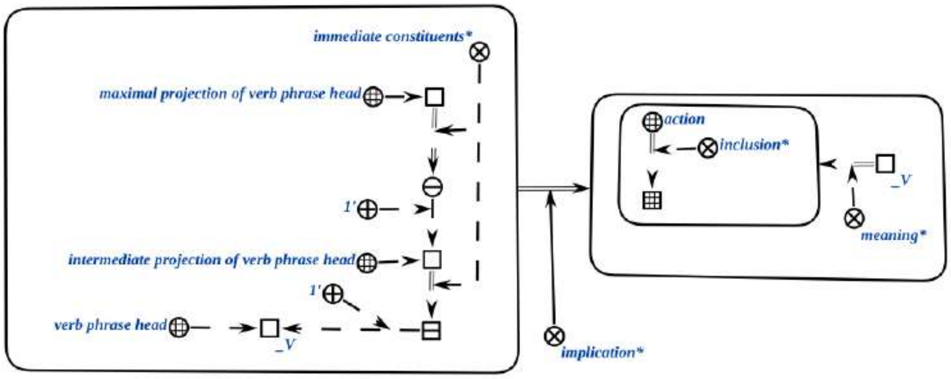

Figure 6. An example of transitioning from syntactic structure to its semantics.

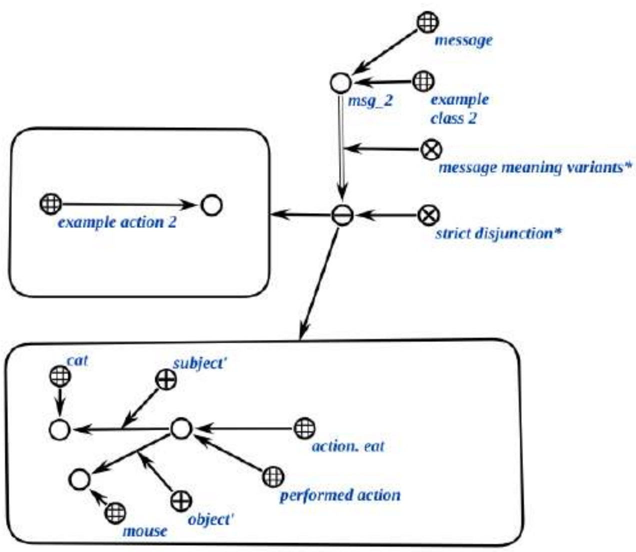

Figure 7. An example of a construction specifying potential meaning of a message.

the value of this function is the highest. If a suitable context is not found, a new one is generated. An example of the result of this action is shown in figure 10.

Action. choosing the meaning of the message is the choice from a set of translation options and their corresponding contexts of one pair and its designation as a construction equivalent to the message. In the simplest case, at this stage, it is permissible to make a choice in accordance with the values of the $F_{CTD}(T, C)$ function calculated at the previous stage for pairs of potentially equivalent structures and their corresponding contexts and choose the pair for which it has the highest value, but if necessary, it is also possible to introduce a separate function. An example of the result of this action is shown in figure 11.

Action. embedding of a message into a context is the embedding of the resulting meaning of a message into a

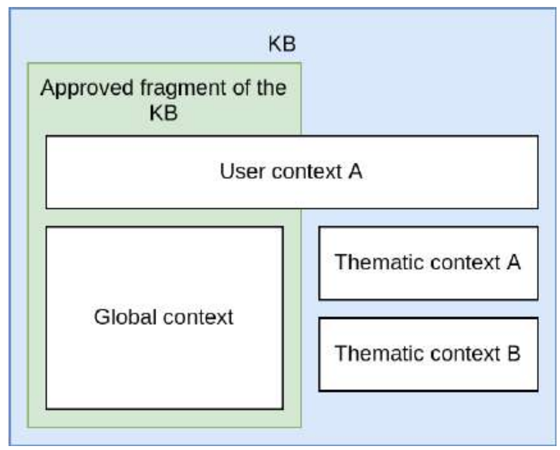

Figure 8. Relationship of contexts with the approved part of knowledge bases.

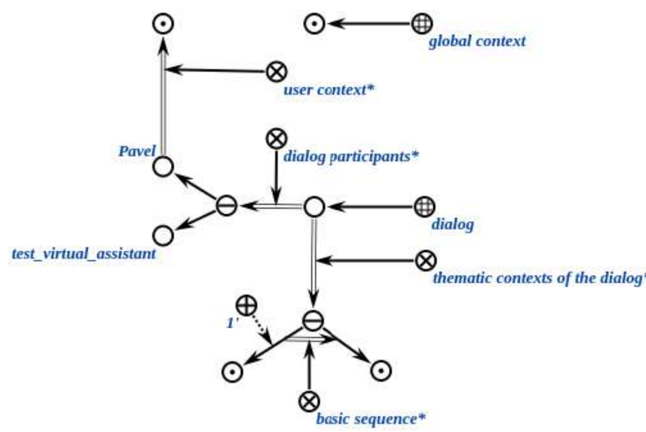

Figure 9. Context specification example.

context. In addition to the chosen meaning of the message, other information necessary for processing the message can be added to the context. Moreover, at this stage of the analysis, pronoun resolution should also be performed based on the

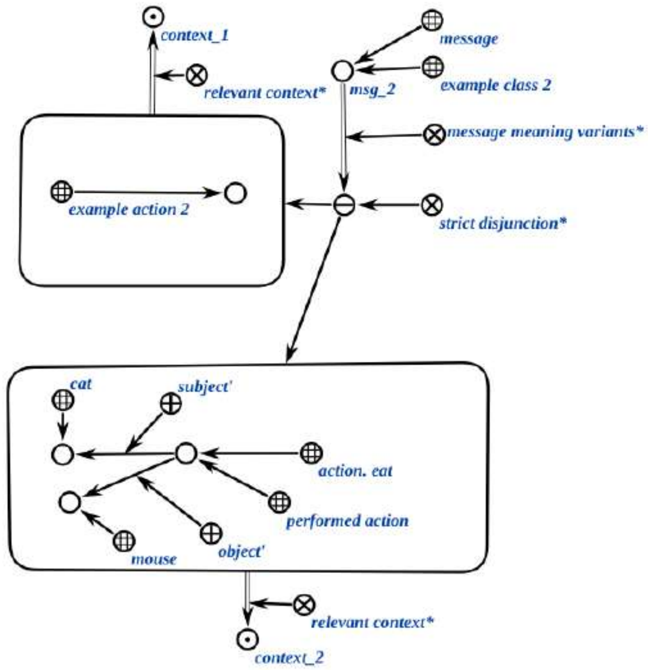

Figure 10. An example of a message with context associated with all meaning variants.

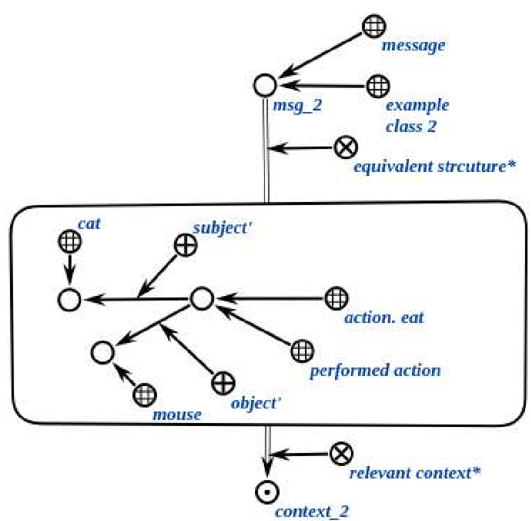

Figure 11. An example of a construction describing a structure equivalent to a message.

information stored in the context. Examples of contexts before and after the message is embedded in it are shown in figures 12 and 13.

Thus, relevant information is collected in a thematic context, and by combining it with the user context and the global context you can get a common context based on which the required system actions should be performed, including the generation

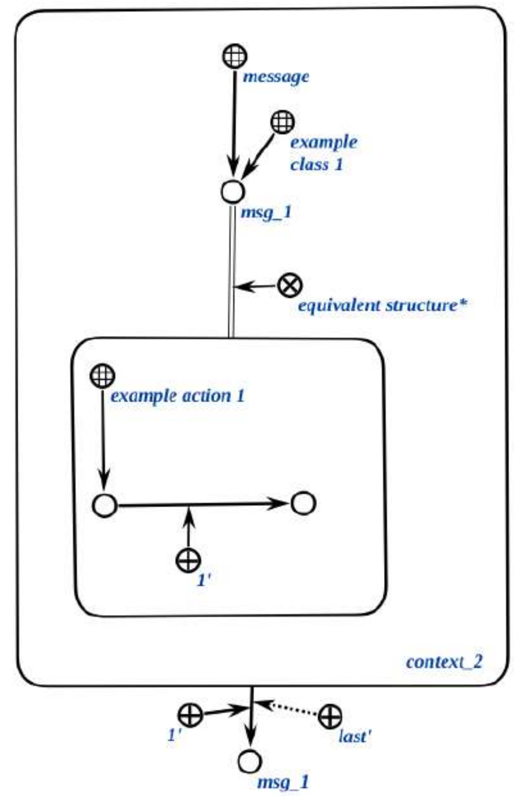

Figure 12. An example of a context before a message is embedded.

of a system response.

IV. Conclusion

An approach has been suggested for the implementation of natural language interfaces of next-generation intelligent computer systems built using OSTIS technology based on a formal model of a natural language, and a dialog context model was introduced.

In the suggested approach, all stages of analysis, including lexical, syntactic and semantic analysis, can be performed directly in the knowledge base of such a system. And all the results (both intermediate stages and the final one) are presented in a single unified form, which helps to ensure compatibility and reduce overhead costs for integrating a subsystem based on this approach.

The formalization of the language used in the basis of this approach is universal and extensible, which makes it possible to supplement it with a formalized specification of a given natural language to ensure the possibility of working with it.

Acknowledgment

The authors would like to thank the research teams of the Department of Intellectual Information Technologies of BSUIR and the Department of Theory and Practice of Translation #1 of MSLU for their help and valuable comments. This research was partially supported by the BRFFR (BRFFR-RFFR No. F21RM-139)..

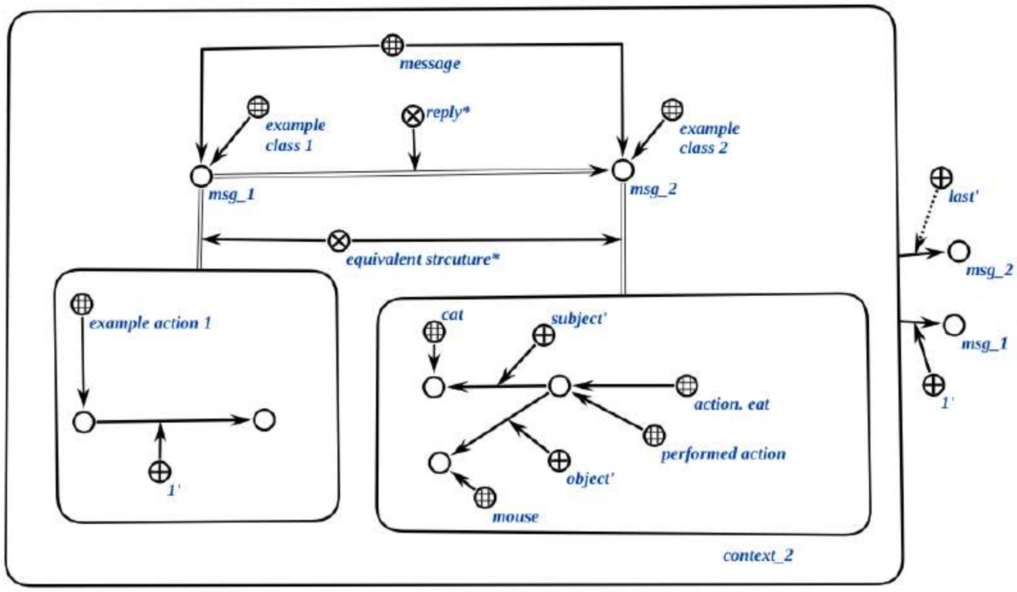

Figure 13. An exaple of a context after a message has been embedded.

References

[1] "Global Voice Assistant Market By Technology, By Application, By End User, By Region, Competition, Forecast & Opportunities, 2024," Available at: https://www.businesswire.com/news/home/20190916005535/en/Global-Voice-Assistant-Market-Projected-Grow-1.2, (accessed 2019, Dec).

[2] S. Pais, J. Cordeiro, and M. L. Jamil, "Nlp-based platform as a service: a brief review," *Journal of Big Data*, vol. 9, 04 2022.

[3] D. Trajanov, V. Trajkovski, M. Dimitrieva, J. Dobreva, M. Jovanovik, M. Klemen, A. Žagar, and M. Robnik-Sikonja, "Survey of nlp in pharmacology: Methodology, tasks, resources, knowledge, and tools," 08 2022.

[4] Khurana, D., Koli, A., Khatter, K. et al., "Natural language processing: state of the art, current trends and challenges," in *Multimed Tools Appl*, 2022.

[5] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[6] "Large language models: A new moore's law?" Available at: https://huggingface.co/blog/large-language-models, (accessed 2022, September).

[7] "Amazon Alexa Official Site: What Is Alexa?" Available at: https://developer.amazon.com/alexa, (accessed 2022, September).

[8] "Siri," Available at: https://www.apple.com/siri/, (accessed 2022, September).

[9] "Google Assistant, your own personal Google," Available at: https://assistant.google.com/, (accessed 2022, September).

[10] "Cortana helps you achieve more with less effort. Your personal productivity assistant helps you stay on top of what matters, follow through, and do your best work." Available at: https://www.microsoft.com/en-us/cortana, (accessed 2022, September).

[11] M. Hoy, "Alexa, siri, cortana, and more: An introduction to voice assistants," *Medical Reference Services Quarterly*, vol. 37, pp. 81–88, 01 2018.

[12] V. V. Golenkov, N. A. Gulyakina, D. V. Shunkevich, *Open technology for ontological design, production and operation of semantically compatible hybrid intelligent computer systems*, G. V.V., Ed. Minsk: Bestprint, 2021.

[13] V. V. Golenkov, "Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems," in *Software products and systems*. Moskow, 2020, pp. 404–412.

[14] R. Jackendoff, *X–bar Syntax: A Study of Phrase Structure*. Cambridge: MIT Press, 1977.

Естественно-языковые интерфейсы интеллектуальных компьютерных систем нового поколения

Гойло А. А. Никифоров С. А.

В данной работе рассматривается подход к реализации естественно-языковых интерфейсов интеллектуальных компьютерных систем нового поколения, построенных по технологии OSTIS, а также предлагается модель контекста диалога. В данном подходе все этапы анализа, включая лексический, синтаксический и семантический анализ могут производиться непосредственно в базе знаний такой системы. Такой подход позволит эффективно решать такие задачи как управление глобальным и локальным контекстами диалога, а также разрешение языковых явлений таких как анафоры, омонимия и эллиптические фразы.

Received 28.10.2022