Chapter 11
Natural Language Processing Applications

Abstract The principles of applying the theory of K-representations to the design of two semantics-oriented natural language processing systems are set forth. The first one is the computer system Mailagent1; the task to be solved by this system is semantic classification of e-mail messages stored in the user's mailbox for enabling the user to more quickly react to the more important and/or urgent messages. The second system is the linguistic processor NL-OWL1 transforming the descriptions in restricted Russian language of situations (in particular, events) and the definitions of notions first into the K-representations and then into the OWL-expressions.

11.1 The Structure of a Computer Intelligent Agent for Semantic Classification of E-mail Messages

11.1.1 The Problem of Semantic Classification of E-mail Messages

One of the peculiarities of the new information society is a huge number of e-mail messages received every day by intensively working specialists. For instance, it is noted in [169] that a specialist may receive every day approximately 100 e-mail messages. That is why in case of a 4-day business trip he/she will face the necessity of analyzing 500 messages (400 received during his/her trip and 100 received just after the return from the trip). The realization of such an analysis demands to spend a lot of time. Most often, a specialist has no such time. Hence it is very likely that he/she will be unable to answer some important messages in due time. That is why the task was posed of designing an intelligent computer agent (an electronic secretary) being able to classify the e-mail messages in the English language. As a result, the computer intelligent agent Mailagent1 was elaborated [99]. Its functions are as follows:

The system Mailagent1 is intended for automatic classification of the e-mail messages stored in the mailbox of a user. The work of the user with the preliminary

V.A. Fomichov, *Semantics-Oriented Natural Language Processing*, IFSR International Series on Systems Science and Engineering 27, DOI 10.1007/978-0-387-72926-8_11,

sorted e-mail messages saves time and enables the user to more quickly react to the more important and/or urgent messages.

The elaborated system has two main components: the adaptation subsystem and the subsystem of linguistic analysis. It is assumed that the receipt of the e-mail messages is the function of a usual e-mail program. That is why the described system destined for the classification of e-mail messages has the possibility of adaptation to the format used for saving the messages on a hard disk of the user's computer.

The subsystem of linguistic analysis proceeds from the following parameters in order to classify the e-mail messages: (1) whether the receiver is waiting for the considered message (from a particular specialist, from a particular address, or from an address belonging to a particular group of addresses); (2) what is the indicated deadline for sending a reply; (3) is it a message sent personally to the receiver (an individual letter) or a message sent at all addresses of a mailing list (e.g., DBWORLD). If the deadline for sending a reply isn't indicated in the text of the message but a hyperlink (an URL) to a Web page is given, the program finds the corresponding Web page and analyzes the content of this Web page. If this Web document indicates the deadline for sending a reply (e.g., for submitting a paper to the Program Committee of an international conference), then the program adds this information to the considered e-mail message.

The messages addressed personally to a receiver form the most important part of the correspondence. That is why a semantic analysis of such messages is carried out in order to understand (in general) their meaning. The basis for fulfilling the semantic analysis of e-mail messages is a linguistic database. Its central components are the system of semantic-syntactic patterns and the lexical – semantic dictionary. The result of semantic analysis is the distribution of the messages into the conceptual categories. The program Mailagent1 possesses the means for visually representing the categories of the messages obtained in the course of its work. Besides, this program provides the possibility of viewing the contents of the e-mail messages belonging to each conceptual category. This makes the viewing of the results a quick and convenient process.

All methods of linguistic analysis employed in the system of automatic classification of the e-mail messages have shown in practice their working properties and effectiveness. A background for elaborating these methods was provided by the theory of K-representations. If necessary, the methods of linguistic analysis can be expanded for the work with new situations by means of a modification of the linguistic database of the program. The computer system Mailagent1 is implemented with the help of the programming language Java.

11.1.2 An Outline of the Computer Intelligent Agent Mailagent1

The constructed computer system Mailagent1 functions in the following way. After the user of this system inputs the date of viewing the received e-mail messages, Mailagent1 forms two systems of folders: (1) EXPECTED MESSAGES,

(2) OTHER MESSAGES. The folder 1 contains the e-mail messages received from (a) the persons stored in a special list in the computer's memory. A part of these people is simply important for the end user, and he/she expects to receive a message from other people in this list.

For the formation of the folder 1, the following information is used: (a) the first and last names of people being particularly important for the end user; (b) the list of fixed e-mail addresses; (c) the list of fixed Internet sites, if the end user is waiting for a message from such an e-mail address that an ending of this address shows its association with an Internet site from this list.

For instance, a researcher is expecting a message with the decision of the Organizing Committee of an International Conference to be held at the University of Bergen, Norway, about a grant for attending this conference. This decision can be sent by any of the members of the Organizing Committee from his/her personal address, and the only common feature of these addresses is an ending. So in the considered case the list of stored information about the Internet sites is to include the ending "uib.no."

The folder 2, called OTHER MESSAGES, is destined for storing all other received e-mail messages, i.e., the e-mail messages not included in the folder EXPECTED MESSAGES. Both the folders have subfolders called (1) UNDEFINED, (2) OVERDUE, (3) 1 WEEK, (4) 1 MONTH, (5) OTHER MESSAGES. The subfolder UNDEFINED contains all messages with indefinite last date of a reply. All such received messages that the deadline for returning a reply has been over are stored in the subfolder OVERDUE. The subfolders 1 WEEK and 1 MONTH are destined for the messages that are to be answered in one week and one month, respectively.

Each of the folders 1–5 contains the subfolders (1) PERSONAL MESSAGES and (2) COLLECTIVE MESSAGES. If an e-mail message is addressed just to a particular person, then this message is included in the subfolder (1), otherwise (e.g., in case of a message from a mailing list, such as DBWORLD) a message is included in the folder (2).

Personal messages may be most interesting to the end user. Such messages may contain the proposals to participate in an international scientific project, to prepare an article for a special issue of an international journal or to write a chapter for a book, to become a member of the Program Committee of an international scientific conference or a member of the Editorial Board of an international scientific journal, etc. Let's say that this part of personal messages encourages the end user to carry out some action. The other part of personal messages may express gratitude for some action fulfilled before by the end user; for sending a hard copy of a paper, for an invitation to take part in an international workshop, etc. Obviously, there are also some other categories of the received e-mail messages.

That is why the elaborated computer system tries to "extract" from an individual message its generalized meaning and, proceeding from this extracted meaning, to associate this message with some conceptual category. The examples of such conceptual categories (or the goals of sending a message) are ACTION ENCOURAGEMENT and THANKS. The messages may express, in particular, the generalized meanings SEND PAPER and COLLOBORATE IN THE PROJECT.

In a message, the generalized meaning "TO SEND PAPER" can be formulated in two possible contexts: (C1) "I would be glad if you send me your recent paper ⟨...⟩" and (C2) "I would be glad to send you the paper ⟨...⟩." So it is very important to reflect in the representation of the goal of sending a message whether it reflects an expected action of the recipient of this message or it reflects an itention of its sender. With this aim, the meaning of the message is completed with the information "Action" for the context C1 or "Sender's intention" for the context C2. So the meaning of a message for the contexts C1 and C2 is represented with the help of the string "Action: SEND PAPER" and the string "Sender's Intention: SEND PAPER."

11.1.3 General Structure of Computer Intelligent Agent Mailagent1

The elaborated system Mailagent1 consists of the following four modules interacting with its main modules:

1. the module of looking for individually oriented messages;
2. the module of finding the deadline for a reply to a message;
3. the module of the analysis whether a message has been expected; item the module of semantic analysis of individual letters.

The joint work of these modules enables the system to write every message in a folder corresponding to this message. In this chapter, the principal attention is paid to describing the work of the last module.

The structure of the folders of the computer system Mailagent1 is reflected in Fig. 11.1. The subfolders of the lower levels of the folder PERSONAL MESSAGES are formed automatically in the process of semantically classifying the stored e-mail messages. For the messages from such subfolders, their generalized meaning is indicated, and a connection of the described action with the sender of a message or its receiver is explained by means of the heading "Action" or "Sender's Intention."

11.1.4 The Structure of Semantic-Syntactic Patterns and Lexico-Semantic Dictionary

In the elaborated computer intelligent agent Mailagent1, the linguistic analysis of the contents of the received e-mail messages is based on the use of special semantic-syntactic patterns (SSPs). The idea is that the employment of rather simple means is able to give a considerable effect as concerns semantic classification of the received messages. A collection of SSPs (stored as textual files) is a part of the Linguistic Database (LDB) of Mailagent1.

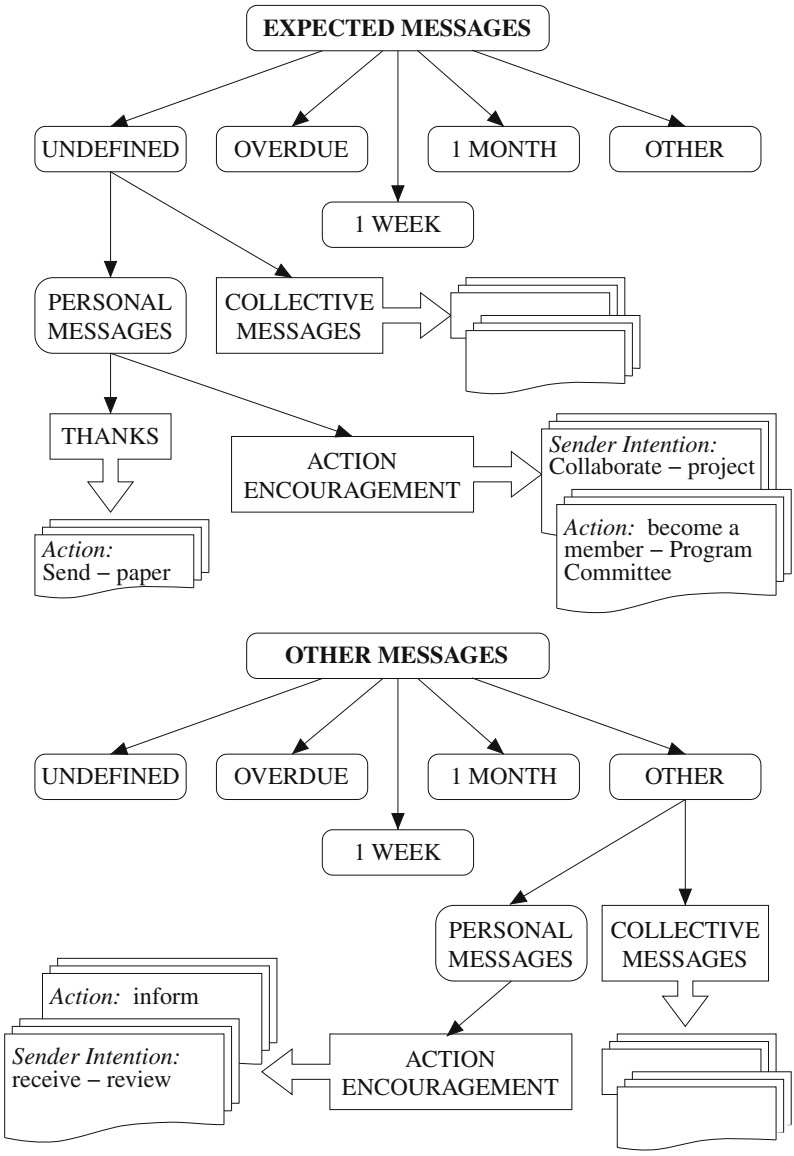

Fig. 11.1 General structure of the folders of the computer system Mailagent1

From the formal point of view, SSPs are strings. Basic components of these strings are substrings called positive indicators of units(PIUs) and negative indicators of units (NIUs). Suppose that arbitrary PIU A1 and NIU B1 are the components of an arbitrary SSP Pt1. Then A1 shows that every NL-text T1 being "compatible"

with Pt1 is to include a component of some first definite kind. To the contrary, the meaning of B1 is that such text T1 is not to include the components of some second definite kind.

Let's consider the distinguished classes of PIU (the number of such classes is four). The PIUs of the first class have the beginning $'1\#'$ and show that some English words or short word combinations (for instance, "would," 'grateful," etc.) are to occur in the analyzed text. If for expressing some meaning, any of the synonimical words $d_1, \ldots, d_n$, where $n > 1$, can be used, then a corresponding PIU is to include the fragment $d_1/, d_2/ \ldots, /d_n$..

For instance, the expression $1\#grateful/thankful/appreciate$ is an example of a PIU.

The PIUs of the *second class* have the beginning $'2\#'$ and say that every text being compatible with the considered semantic-syntactic pattern is to include a word belonging to one of the speech parts indicated after the beginning $'2\#'$. The expressions $'2\#subs't, '2\#subst/attr'$ are the examples of the PIUs from the second class, where subst , attr designate the speech parts "substantive" and "attribute."

The *third class* of PIUs is formed by the elements demanding the occurrence in the considered texts of the words and expressions being the designations of some semantic items: "information transmission", "telephone," "a transport means," etc. The PIUs from this class have the beginning $'3\#'$. The examples of such elements are as follows: $'3\#inform.transmis', '3\#telephone'$.

The PIUs of the last, *fourth class* designate such sort or a combination of sorts that this sort s_1 or a combination of sorts

$$s_1 * s_2 * \ldots * s_n,$$

where $n > 1$, is to be associated with some word being necessary for expressing a given meaning.

For instance, a semantic-syntactic pattern may include the element

$$'4\#dynam\_phys\_ob * intel\_syst',$$

where the expressions *dynam_phys_ob* and *intel_syst* are to be interpreted as the sorts "dynamic physical object" and "intelligent system," respectively. This component of an SSP can be used in order to show that any NL-text being compatible with the used SSP is to include a designation of a person (being simultaneously a dynamic physical object and an intelligent system).

The idea of using the concatenations of the sorts stems from the theory of K-representations. Since very many words are associated with several sorts, i.e., general semantic items (speaking metaphorically, the entities denoted by such words have different "coordinates" on different "semantic axes"), different "semantic coordinates" of a word are taken into account in order to find conceptual connections of the words in NL-texts.

Each negative indicator of a unit is a string of the form *%%Expr*, where *Expr* is any PIU. Such components of semantic-syntactic patterns demand the lack in the analyzed NL-texts of the fragments of the four kinds discussed above.

For example, the NIU %%1# would of an arbitrary SSP Pt1 require that any NL-text being compatible with $Pt1$ doesn't include the word "would." Such an element can be used as follows. If a message includes the fragment

"I (or we) would be grateful (or thankful) to you for $\langle Designation\,of\,an\,Action\rangle$",

we understand that the indicated action is a desired action, but it wasn't fulfilled by the moment of sending the message. To the contrary, a similar fragment without the word "would" usually shows that the mentioned action was fulfilled by the moment of sending the message, and one of the purposes of this message is to thank the recipient of the message for some carried out action.

Some special components of the SSPs establish the direction and borders in order to look for the needed fragments of the analyzed texts. Such components are the expressions of the form "M-1," "M-2," "M-3," "M+1," "M+2," etc. If such expression includes the sign "minus," then the needed fragments are to be searched in the sentences preceding the considered sentence. The number k after the sign "minus" ("plus") indicates how many sentences before (after) the considered phrase are to be analyzed.

For instance, if $k = 1$, then only the preceding sentences are to be processed; if $k = +2$, then the considered phrase and next two sentences are to be analyzed. The ending of the zone for the search is the symbol "/".

Every semantic-syntactic pattern (SSP) Pt can be represented in the form

$$A***B***C,$$

where the fragments A, B, C are as follows: The fragment A is a sequence of positive and (only in some patterns) negative indicators of units (PIUs and NIUs), these indicators are separated by commas. Such a sequence of indicators expresses a system of requirements to be satisfied by each NL-text being compatible with the considered SSP Pt.

The fragment B is a designation of the meaning of every NL-text being compatible with this SSP Pt. The fragment C is a sequence of positive indicators of units enabling to concretize the meaning of the analyzed NL-text and to submit this meaning to the recipient of the e-mail messages. If a component of the fragment C of an SSP isn't destined for including in the meaning submitted to the recipient of the messages, one poses the sign "minus" ("–") before the corresponding component of C.

Example 1. Let $Pt1$ be the expression

$$1\#I/we,\ \%\%1\#would/in\,advance,$$

$$1\#thankful/grateful/appreciate$$

$$***THANKS***4\#phys\_action,$$

$$4\#inf\_object*phys\_object/phys\_object.$$

This expression is a semantic-syntactic pattern (SSP) including both positive and negative indicators of units. If any NL-text T1 is compatible with the SSP $Pt1$, the

computer intelligent agent Mailagent1 draws the conclusion that the generalized meaning of T1 is THANKS, i.e., the purpose of sending an e-mail message with T1 was to express gratitude for carrying out some physical action. An important argument in favor of this conclusion is the lack in T1 both of the word "would" and the expression "in advance."

The SSP $Pt1$ helps also to find a fragment of T1 describing such physical action. It is done in the following way. After finding one of the words "thankful," "grateful," or "appreciate" (let's denote it by $d1$), the computer agent analyzes the words to the right from the word $d1$. Suppose that it first discovers the word $d2$ that can be associated with the sort "physical action," and then finds (to the right from $d2$) the word $d3$ such that $d3$ can be associated either both with the sort "informational object" and "physical object" or with the only sort "physical object."

Then the computer agent submits to the mail box user the string $d2\,h\,d3$, where h is either the null string or the substring of T1 separating $d2$ and $d3$. In this case, the submitted string $d2\,h\,d3$ is interpreted as a description of the carried out action (this was the cause of sending an e-mail message with an expression of gratitude).

Example 2. Let's consider the expression $Pt2$ of the form

$$1\#thank,\ 1\#in\,advance,\ M-2/3\#possibility***$$

$$ACTION\ ENCOURAGEMENT***M-3/4\#phys\_action,$$

$$4\#inf\_object*phys*object/phys\_object.$$

The ideas underlying this SSP are as follows. An e-mail letter can implicitly encourage its receiver to carry out some physical action. Let's imagine, for instance, that a letter contains the following fragment T2:

"I would be happy to receive from you a hard copy of your paper published last year in Australia. Would it be possible? If yes, thank you in advance for your time and efforts."

The SSP $Pt2$ will be matched against T2 as follows. The third sentence of T2 contains the word "thank" and the expression "in advance." The element $'M-2/'$ stimulates the computer agent to look for the semantic unit "possibility" as one of the semantic units associated with the words in two sentences before the considered third sentence. Since the second sentence contains the word "possible," Mailagent1 draws the conclusion that the generalized meaning of T2 is ACTION ENCOURAGEMENT.

The next question is what action is to be carried out. For answering this question, the SSP $Pt2$ recommends to analyze three sentences to the left from the found word expressing the meaning "possibility" (due to the element "M-3/") and to look in these sentences for (a) a word or word combination that can be associated with the sort "physical action" and (b) for the word or word combination that can be associated with each of two sorts "informational object," "physical object" or with the only sort "physical object". The case (a) takes place, for instance, for the verbs and verbal substantives "to send," "the sending," "to sign," and the case (b) – for the expressions "a hard copy of your article," "CD-ROM," "this contract."

The system of semantic-syntactic patterns is associated with the other component of a linguistic database (LDB); this component is called Lexico-Semantic Dictionary. It is one of the relations of the LDB; its attributes are as follows: (1) word, (2) speech part, (3) semantic unit (SU), (4) sort 1 associated with the semantic unit, (4) sort 2 associated with the SU or the empty sort NIL, (5) sort 3 associated with the SU or the empty sort NIL. This dictionary is stored in the computer's memory (for instance, in the form of a table) and can be modified. For the elaborated computer program, the lexico-semantic dictionary is a textual file.

11.1.5 Implementation Data

The computer intelligent agent Mailagent1 has been implemented in the programming language Java (version JDK 1.1.5). This computer system is intended for automatic semantic classification of the e-mail messages in English stored on the hard disk of a computer. The program Mailagent1 was tested in the environment Windows 95/98/2000/NT. The file with the messages for the classification was represented as a file of the e-mail system that received these messages from the Internet.

11.2 A Transformer of Natural Language Knowledge Descriptions into OWL-Expressions

In the context of transforming step by step the existing Web into Semantic Web (see Sect. 6.10), the need for large Web-based and interrelated collections of formally represented pieces of knowledge covering many fields of professional activity is a weighty ground for increasing the interest of the researchers to the problem of automated formation of ontologies.

It seems that the most obvious and broadly applicable way is to construct a family of NLPSs being able to transform the descriptions of knowledge pieces in NL (in English, Russian, German, Chinese, Japanese, etc.) into the OWL-expressions and later, possibly, into the expressions of an advanced formalism for developing ontologies.

This idea underlay the design of the computer system NL-OWL1, it is a Russian-language interface implementing a modification of the algorithm of semantic-syntactic analysis *SemSyn* stated in [85]. The main directions of expanding the input language of the algorithm *SemSyn* are as follows:

- the definitions of notions in restricted Russian language can be the input texts of the system;
- a mechanism of processing the homogeneous members of the sentence is added to the algorithm of semantic-syntactic analysis;

- a part of input sentences (the descriptions of the events and the definitions of notions) is transformed not only into the K-representations (i.e., into the expressions of a certain SK-language) but also, at the second stage, into the OWL-expressions.

Figure 11.2 illustrates the structure of the computer system NL-OWL1.

Let's consider the examples illustrating the principles of processing NL-texts by the experimental Russian-language interface NL-OWL1, implemented in the Web programming system PHP.

Example 1. Definition: "Carburettor is a device for preparing a gas mixture of petrol and air."

K-representation:

$$ModuleOfKnowledge\,(definition;\, carburetor;\, x1;$$

$$(Definition1\,(certn\ carburetor\ :\ x1,\, certn\ device\ :\ x2 \wedge$$

$$Purpose\,(x2,\, certn\ preparation1\ :\ x\_e1) \wedge$$

$$Description\,(preparation1,\, Object1\,(certn\ mixture *$$

$$(Type,\, gas)\ :\ x3)) \wedge Product1\,(x3,\, certn\ petrol\ :\ x4)$$

$$\wedge Product1\,(x3,\, air))))$$

OWL-expression:
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}ModuleOfKnowledge\text{”}/\rangle$
$\langle hasFormModule\ rdf\ :\ resource = \text{“}\#definition\text{”}/\rangle$
$\langle hasConcept\ rdf\ :\ resource = \text{“}\#carburetor\_x1\text{”}/\rangle$
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}Action\text{”}/\rangle$
$\langle Action\ rdf\ :\ ID = \text{“}Concept\text{”}/\rangle$
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}Situation\text{”}/\rangle$
$\langle Situation\ rdf\ :\ ID = \text{“}x1\text{”}\rangle$
$\langle hasAction\ rdf\ :\ resource = \text{“}\#Concept\text{”}/\rangle$
$\langle hasDetermination\ rdf\ :\ resource = \text{“}\#Device\_x2\text{”}/\rangle$
$\langle /Situation\rangle$
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}Device\text{”}/\rangle$
$\langle Device\ rdf\ :\ ID = \text{“}Device\_x2\text{”}/\rangle$
$\langle Destination\ rdf\ :\ resource = \text{“}\#preparation\_x\_e1\text{”}/\rangle$
$\langle /Device\rangle$
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}Destination\text{”}/\rangle$
$\langle Destination\ rdf\ :\ ID = \text{“}Destination\_x\_e1\text{”}/\rangle$
$\langle Object1\ rdf\ :\ resource = \text{“}\#Mixture\_x3\text{”}/\rangle$
$\langle /Destination\rangle$
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}Mixture\text{”}/\rangle$
$\langle Mixture\ rdf\ :\ ID = \text{“}Mixture\_x3\text{”}/\rangle$
$\langle Form\ rdf\ :\ resource = \text{“}\#gas\text{”}/\rangle$
$\langle Product1\ rdf\ :\ resource = \text{“}\#Petrol\_x4\text{”}/\rangle$

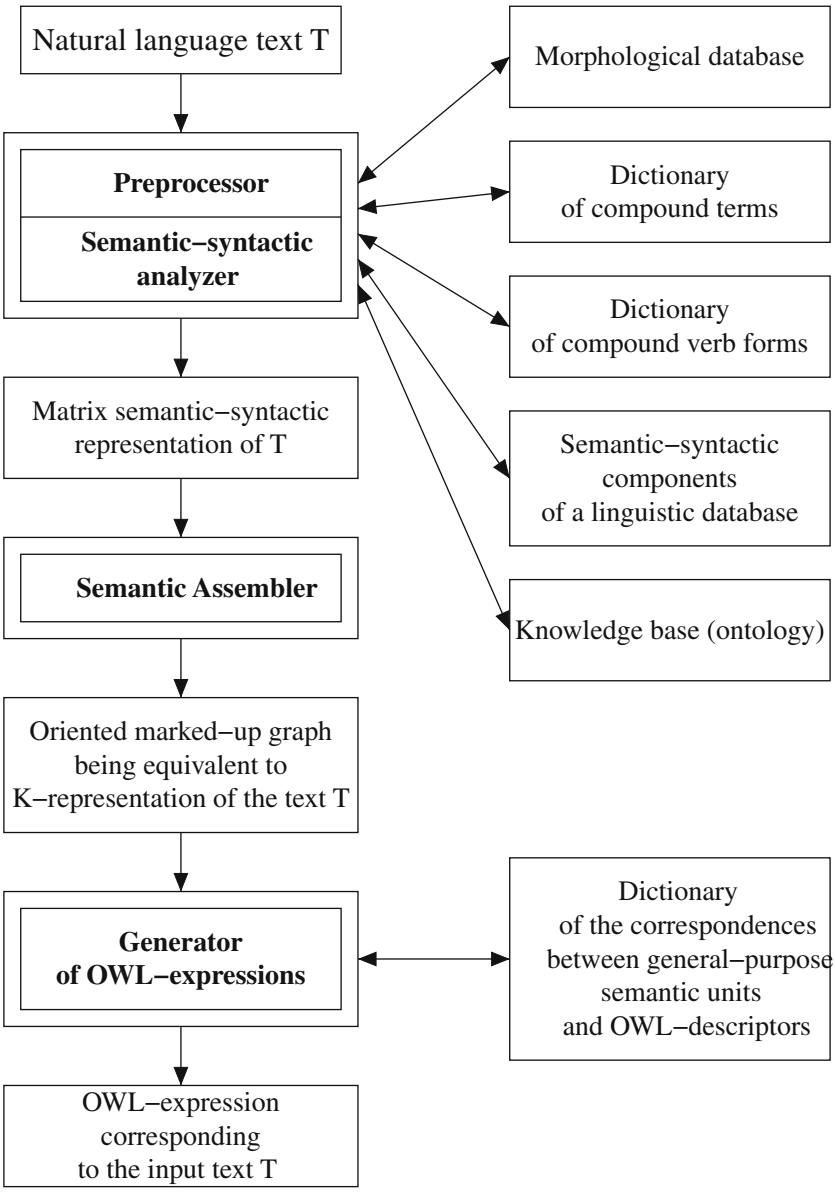

Fig. 11.2 The structure of the computer system NL-OWL1

$\langle Product1\ rdf\ :\ resource\ =\ \text{“\#}Air\_x5\text{”}/\rangle$
$\langle/Mixture\rangle$

Example 2. Ramp is an inclined surface for the entrance of transport means.
K-representation:

$$ModuleOfKnowledge\,(definition;\, ramp;\, x1;\, (Definition1$$

$$(certn\ ramp\ :\ x1,\, certn\ surface * (Position,\, inclined)\ :\ x2 \wedge$$

$$Purpose\,(x2,\, certn\ entrance1\ :\ x\_e1) \wedge Description\,(entrance1,$$

$$Object1\,(certn\ set * (Compos,\, mean * (Type,\, transport))\ :\ S1)))))$$

OWL-expression:
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}ModuleOfKnowledge\text{”}/\rangle$
$\langle hasFormModule\ rdf\ :\ resource = \text{“}\#definition\text{”}/\rangle$
$\langle hasConcept\ rdf\ :\ resource = \text{“}\#ramp\_x1\text{”}/\rangle$
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}Action\text{”}/\rangle$
$\langle Action\ rdf\ :\ ID = \text{“}Concept\text{”}/\rangle$
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}Situation\text{”}/\rangle$
$\langle Situation\ rdf\ :\ ID = \text{“}x1''\rangle$
$\langle hasAction\ rdf\ :\ resource = \text{“}\#Concept\text{”}/\rangle$
$\langle hasDetermination\ rdf\ :\ resource = \text{“}\#Surface\_x2\text{”}/\rangle$
$\langle /Situation\rangle$
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}Surface\text{”}/\rangle$
$\langle Surface\ rdf\ :\ ID = \text{“}Surface\_x2\text{”}/\rangle$
$\langle Disposition\ rdf\ :\ resource = \text{“}\#inclined\text{”}/\rangle$
$\langle Destination\ rdf\ :\ resource = \text{“}\#entrance\_x\_e1\text{”}/\rangle$
$\langle /Surface\rangle$
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}Destination\text{”}/\rangle$
$\langle Destination\ rdf\ :\ ID = \text{“}Destination\_x\_e1\text{”}/\rangle$
$\langle Object1\ rdf\ :\ resource = \text{“}\#Resource\_S1\text{”}/\rangle$
$\langle /Destination\rangle$
$\langle owl\ :\ Class\ rdf\ :\ ID = \text{“}Resource\text{”}/\rangle$
$\langle Resource\ rdf\ :\ ID = \text{“}Resource\_S1\text{”}/\rangle$
$\langle Form\ rdf\ :\ resource = \text{“}\#transport\text{”}/\rangle$
$\langle /Resource\rangle$

Example 3. A hand screw is a transportable mechanism for lifting and holding an object at a small height.

K-representation:

$$ModuleOfKnowledge\,(definition;\, with\ a\ hand\ screw;\, x1;$$

$$(Definition1\,(certn\ hand\ screw\ :\ x1,$$

$$certn\ mechanism * (Feature,\, transportability)\ :\ x2 \wedge$$

$$Purpose\,(x2,\, certn\ lifting1\ :\ x\_e1) \wedge$$

$$Purpose\,(x2,\, certn\ holding1\ :\ x\_e2) \wedge$$

$$Description\,(holding1,\, Object1\,(certn\ object\ :\ x3)) \wedge$$

$$Place1\,(x3,\, certn\ height * (Degree,\, small)\ :\ x4))))$$

OWL-expression:
$\langle owl : Class\ rdf : ID = \text{“}ModuleOfKnowledge\text{”}/\rangle$
$\langle hasFormModule\ rdf : resource = \text{“}\#definition\text{”}/\rangle$
$\langle hasConcept\ rdf : resource = \text{“}\#with\ a\ hand\ screw\_x1\text{”}/\rangle$
$\langle owl : Class\ rdf : ID = \text{“}Action\text{”}/\rangle$
$\langle Action\ rdf : ID = \text{“}Concept\text{”}/\rangle$
$\langle owl : Class\ rdf : ID = \text{“}Situation\text{”}/\rangle$
$\langle Situation\ rdf : ID = \text{“}x1''\rangle$
$\langle hasAction\ rdf : resource = \text{“}\#Concept\text{”}/\rangle$
$\langle hasTime\ rdf : resource = \text{“}\#Now\text{”}/\rangle$
$\langle hasDetermination\ rdf : resource = \text{“}\#Mechanism\_x2\text{”}/\rangle$
$\langle /Situation\rangle$
$\langle owl : Class\ rdf : ID = \text{“}Mechanism\text{”}/\rangle$
$\langle Mechanism\ rdf : ID = \text{“}Mechanism\_x2\text{”}/\rangle$
$\langle Property\ rdf : resource = \text{“}\#transportable\text{”}/\rangle$
$\langle Destination\ rdf : resource = \text{“}\#lifting\_x\_e1\text{”}/\rangle$
$\langle Destination\ rdf : resource = \text{“}\#holding\_x\_e2\text{”}/\rangle$
$\langle /Mechanism\rangle$
$\langle owl : Class\ rdf : ID = \text{“}Destination\text{”}/\rangle$
$\langle Destination\ rdf : ID = \text{“}Destination\_x\_e2\text{”}/\rangle$
$\langle Object1\ rdf : resource = \text{“}\#Object\_x3\text{”}/\rangle$
$\langle /Destination\rangle$
$\langle owl : Class\ rdf : ID = \text{“}Object\text{”}/\rangle$
$\langle Object\ rdf : ID = \text{“}Object\_x3\text{”}/\rangle$
$\langle Place1\ rdf : resource = \text{“}\#Height\_x4\text{”}/\rangle$
$\langle /Object\rangle$
$\langle owl : Class\ rdf : ID = \text{“}Height\text{”}/\rangle$
$\langle Height\ rdf : ID = \text{“}Height\_x4\text{”}/\rangle$
$\langle Extent\ rdf : resource = \text{“}\#small\text{”}/\rangle$
$\langle /Height\rangle$.

Due to the broad expressive possibilities of SK-languages, the intelligent power of the transformer NL-OWL1 can be considerably enhanced. That is why the formal methods underlying the design of the system NL-OWL1 enrich the theoretical foundations of the Semantic Web project.