question generation to the automatic scoring of test papers and the effective evaluation of subsystems, the basic principles of answer verification to subjective questions are thus briefly presented. For a detailed understanding of the process of constructing semantic models for subjective questions and user answer verification to subjective questions please refer to the literature [3].

Once the similarity between the answers to the subjective questions is obtained, the correctness and completeness of the user answers can be verified combined with the evaluation strategy for the subjective questions. The evaluation strategy for subjective questions includes:

- if the similarity between the answers is equal to 1, the user answer is completely correct;
- if the similarity between the answers is less than 1 and the precision is equal to 1, the user answer is correct but incomplete and the user score is $R_{sc} * Max_{score}$;
- if the similarity between the answers is greater than 0 and less than 1, and the precision is less than 1, then the user answer is partially correct and the user score is $Fsc * Max_{score}$;
- if the similarity between the answers is equal to 0, the user answer is wrong.

The proposed approach to automatic verification of user answers has the following advantages:

- verifying the correctness and completeness of user answers based on semantics;
- the logical equivalence between answers can be determined;
- the similarity between any two semantic graphs in the knowledge base can be calculated;
- the developed component using the proposed approach can be easily transplanted to other ostis-systems.

IV. KNOWLEDGE BASE OF THE SUBSYSTEM

The knowledge base of subsystem is used to store automatically generated test questions, and it also allows to automatically extract a series of test questions and form test papers according to user requirements. Therefore, in order to improve the efficiency of accessing the knowledge base of the subsystem and the efficiency of extracting the test questions, an approach to construct the knowledge base of the subsystem according to the type of test questions and the generation strategy of the test questions is proposed in this article.

The basis of the knowledge base of any ostis-system (more precisely, the sc-model of the knowledge base) is a hierarchical system of subject domains and their corresponding ontologies [1], [2], [5]. Let's consider the hierarchy of the knowledge base of subsystem in SCn-code:

Section. Subject domain of test questions
$\Leftarrow$ *section decomposition*:*
{ • *Section. Subject domain of subjective questions*
$\Leftarrow$ *section decomposition*:*
{ • *Section. Subject domain of definition explanation question*
• *Section. Subject domain of proof question*
• *Section. Subject domain of problem-solving task*
}
• *Section. Subject domain of objective questions*
$\Leftarrow$ *section decomposition*:*
{ • *Section. Subject domain of multiple-choice question*
• *Section. Subject domain of fill in the blank question*
• *Section. Subject domain of judgment question*
}
}

Objective types of test questions are decomposed into more specific types according to their characteristics and corresponding test question generation strategies. Next, taking the judgment question as an example let us consider its semantic specification in SCn-code:

judgment question
$\in$ *maximum class of explored objects':*
Subject domain of judgment question
$\Leftarrow$ *subdividing*:*
{ • *judgment question based on relation attributes*
• *judgment question based on axioms*
• *judgment question based on image examples*
• *judgment question based on identifiers*
• *judgment question based on elements*
$\Leftarrow$ *subdividing*:*
{ • *judgment question based on role relation*
• *judgment question based on binary relation*
}
• *multiple-choice question based on classes*
$\Leftarrow$ *subdividing*:*
{ • *judgment question based on subdividing relation*
• *judgment question based on inclusion relation*
• *judgment question based on strict inclusion relation*
}