# Ontology-based Project Solutions Instances Library Creation Method for the Reuse Concept in the Industry

Nikolay Voit
'Computer Engineering' department
Ulyanovsk State Technical University
Ulyanovsk, Russia
n.voit@ulstu.ru

Semen Bochkov
'Computer Engineering' department
Ulyanovsk State Technical University
Ulyanovsk, Russia
bochkovsi@ido.ulstu.ru

Sergey Kirillov
'Computer Engineering' department
Ulyanovsk State Technical University
Ulyanovsk, Russia
kirillovsyu@gmail.com

Maria Ukhanova
'Computer Engineering' department
Ulyanovsk State Technical University
Ulyanovsk, Russia
mari-u@inbox.ru

Sergey Brigadnov
'Computer Engineering' department
Ulyanovsk State Technical University
Ulyanovsk, Russia
sergbrig@yandex.ru

Dmitry Kanev
'Computer Engineering' department
Ulyanovsk State Technical University
Ulyanovsk, Russia
dima.kanev@gmail.com

***Abstract*— The paper considers and investigates main approaches, methods and means of systematization and accumulation of project solutions made with the CAD tools for reuse in industry. The authors propose an ontological model of the CAD domain for solving problems of accumulation and management of project knowledge. New method of forming libraries of project solutions instances has been developed, which allows to save and modify project solutions taking into account new project tasks. The authors developed an algorithm for the formation and filling of the library of copies of project solutions made in the CAD KOMPAS.**

***Keywords- CAD, Ontology, Project solutions, Instance library.***

## I. INTRODUCTION

Designing in the enterprise in modern conditions uses both CAD systems and automated knowledge management and storage systems. As a result of project activities, project solutions are created that need to be systematized and accumulated for reuse in production. The project solutions reuse effectiveness is purposed to reduce the time required for design and technological preparation of production, which is based on automatization of the technical documents development, storage and knowledge management performed in a unified information space. To solve the knowledge storage and management problem, ontological knowledge bases are used. The main goal of developing libraries is to provide systematic information on requirements and parameters for further use in the in-depth development of these products and projects based on them.

Thus, to improve the design activities on the enterprise, the project solutions database should contain a library of CAD-based project solutions instances, project information systematization methods and allow such information to be obtained with other programs presented in CAD series. If the project solutions database designer plans to upgrade it with new project solutions instances, then it is important to first systematize the contents of the project knowledge database. The first step will be the ontology-based semantic product model development [11]. The second will be the project solutions instances libraries method development. The next step is an intelligent search for project solutions on given parameters which will be considered in the future works.

## II. RELATED WORKS

Currently, several classes of knowledge models are applied, which illustrate the representation of knowledge as a formalization of information. Each class of the knowledge model formalizes information on the project solution in the form of knowledge, which is referred to or used in the design process, in accordance with certain structural rules. The main models of knowledge are: logical, production, frame, network, object-oriented, ontological.

In logical models, knowledge is represented as a set of given formulas for a formal description of the system. Each formula refers to a system of axioms and must fulfill the requirements of consistency, independence and completeness. Hilbert [1] pro-posed four axioms that must satisfy the specified requirements. However, the logic of statements as an independent model of knowledge is only

978-1-7281-9260-4/20/$31.00 ©2020 IEEE
DOI 10.1109/ICCSA50381.2020.00021

suitable for describing the relationships between statements that are considered as a whole.

The frame model concept was first introduced by Minsky [2] as a knowledge structure for solving problems with spatial scenes with pattern recognition and speech understanding. The advantages of the frame model are visibility, flexibility, uniformity, a high degree of structuring of knowledge, as well as the integration of declarative and procedural knowledge. At the same time, the frame model is characterized by the complexity of managing the output and the low efficiency of its procedures [3].

The network model of knowledge reflects a more general way of representing knowledge, which considers the set of aggregate objects and related relationships. In the practical application of semantic networks to represent knowledge, an important point is the unification of the object types and relations between them. The main advantage is that it is close to the ideas of the long-term human memory structure [4]. At the same time, they are characterized by the ambiguity of knowledge and relationships representations, as well as complex procedures for the semantic network output.

An object-oriented knowledge model is used in modern technologies for developing software tools and information systems. This model is able to display information about the complex relationships between objects. The disadvantages are high conceptual complexity, inconvenience of data processing and low query execution speed.

Ontologies allow the knowledge models organization by linking declarative descriptions and definitions. The following basic requirements of experts in applied fields to the means of building ontology are distinguished [5]: the proximity of the language to the professional language of the expert knowledge base; the ability to use the entered knowledge to solve most subject problems and the formation of new ones based on existing ones; the ability to add new language constructs that appear in a specific knowledge base. The ontology developer is forced to solve many problems [6], both typical ones that arise during design and have known solutions, as well as specific ones, the solution of which depends on a specific subject area.

In [7], the use of ontologies in software engineering is studied using a flexible approach. The authors implemented an OWL ontology that performs reasoning about the requirements for a software product. The requirements themselves are presented as user stories written in natural language. In turn, user stories include the role of the user, the goal achieved, the value to the user, the acceptance criterion.

Each ontological knowledge base is aimed at solving the problems of the selected subject area. In this paper, the formation of the product semantic model and its organizational and technical components based on the ontology is proposed. The formation of a semantic model occurs using the unstructured data, documents and PDM objects extraction method. These items are performed in a series of specialized CAD systems. As organizational and technical components in this work we assume the technical task, technical requirements, technical documentation and the project solution information.

## III. Ontology-based Subject Area Model Development

In the subject area, three levels of interaction are distinguished: the conceptual scheme, the project solutions view and the practice.

Conceptual scheme is a design engineering components set. The project solutions view is the accumulation and systematization of subject area knowledge in the ontology form. The practice level is the methods of using the accumulated knowledge of design engineering in the designer work. As an accumulated knowledge using method, structural and parametric CAD project solutions analysis is meant, which allows transforming a query into ontological project solutions base by search parameters.

The following classes are defined in the subject area: technical task for the complex technical product development; operating schedule; developers; technical requirements; design specifications; parts and assembly units; technical documentation and information.

Subject area ontology has the following form:

$$O\_PrO = (PrO, T, R, F), \qquad (1)$$

where PrO = $\{\text{pro}i \mid i = 1..\ x\}$ is a set of CAD project solutions, T is subject area terms inside the ontology.

Set of these terms is presented as the following model:

$$T = \{Class, Obj\}, \qquad (2)$$

where Class = {TZ, G, P, TT, PH, DSE, D} is the ontology classes set (TZ is technical task, G is work schedule, P are developers, TT is technical requirements, PH is CAD project solutions characteristics, DSE is parts and assembly units, D is documentation), the DSE class has a Boolean slot “is part of the assembly”, Obj is the subject area ontology classes objects set.

R is the set of relations between ontology classes objects:

$$R = \{Rhas, Rdefine, Rconsist, Rbased, Raffect, Rinst, Rdevel, Rtodev\}, \qquad (3)$$

where Rhas is a binary relation “in” having the semantics “available in” and associating the subject area ontology classes objects with the DSE class ones, Rdefine is a binary relation “defines” having semantics “define” and associating the TT class objects with the TZ ones, Rconsist is a binary relation "contained in", having the semantics of "consist_in" and linking the D and DSE ontology classes objects with the DSE ones, Rbased is a binary relation "based on", having the semantics of "based_on" and linking the TZ ontology class objects with the D ones, Raffect is binary relation "affects on”, having the semantics of “affect_on” and associating G and TT classes objects with the TZ and DSE ones, Rinst is a binary relation “established”, having the semantics "established" and associating an element of the "Matching signatures" subject field with objects of the D ontology class, Rdevel is a binary relation "developed", having the semantics "developed_by" and linking objects of the G, TZ, D ontology classes with objects of the P ontology class, Rtodev is a binary relation "on development", having the semantics "to_develop" and linking objects of the G ontology class with the DSE and D ones.

Set of interpretive functions of the ontology are presented in the form:

$$F = (Fdse\_op, Fdse\_ph, Fdse\_cl, Fdse\_tt, Fdse\_doc, Fg\_tz), \quad (4)$$

where Fdse_op: DSE → {Operation} is the DSE class object to the CAD design operations set mapping function, Fdse_ph: {DSE} → {PH} is the complex CAD project solution characteristics class mapping function, Fdse_cl: DSE → {Class} is the complex technical CAD project solution class determination function, Fdse_tt: {DSE} → {TT} is DSE class object data and numerical characteristics to TT class objects matching function, Fdse_doc: {DSE} → {D} is CAD project documentation displaying function, Fg_tz: {G} → {TZ} is the schedule work-based technical task formation function.

Example of the proposed subject area ontological model in the semantic network form is presented in Fig. 1. The main advantage of the proposed model is the systematized information about complex technical product designed by CAD tools [8]. This ontology-based project solution presentation in the form of a semantic model allows for structural and parametric analysis of CAD products [9, 10] according to the requirements and design characteristics stated in the terms of reference.

## IV. Project Solutions Instances Library Method Development

Issue of the method is to create a CAD project solutions instances library, modify them to reflect new design tasks, using the reuse concept [12]. To create a CAD project solutions instances library, project solutions construction templates are used. They present as set of options for constructing a three-dimensional solid-state model of an engineering product using project operations performed by the designer in CAD. At the same time, the construction template also includes parameters of sequential project operations and their interconnectedness.

The model library of instances of project solutions has the following form:

$$Library = \{id, proi, template\}, \quad (5)$$

where id is the CAD project solution identifier, proi ϵ subject area is the CAD project solution, template ϵ Templates is the proi project solution construction template.

Set of these project solution construction templates implemented in CAD are presented in the form of the following model:

$$Templates = \{O\_PrOi, Operations, OperationType, Params, F\_historyDSE\}, \quad (6)$$

where O_PrOi is the ontological representation of the i-th project solution implemented in CAD, which includes DSE as a set of parts and assembly units included in the assembly of the final product, Class as a set of project solutions classes (for example, sleeve, screw, washer, etc. ), Operations is the set of project operations for solid-state modeling of DCE in CAD, OperationType is a set of project operations types (for example, extrusion, rounding, tilt, rotation, etc.), Params is a set of parameters and numerical characteristics of project operations, F_historyDSE = Operations × OperationType × Params → l_buildHistory is the function forming the project solution construction history in the form of a project operations sequence performed by the designer during construction in CAD.

Such model representation makes possible to form a history of building a three-dimensional model [13], highlight key parameters and characteristics of a project solution, save the generated template to a library of instances of project solutions for reuse by designers in the process of designing industrial products using CAD tools. CAD KOMPAS, which is one of the most popular packages for designing industrial products at domestic enterprises, especially at the enterprises of the military-industrial complex, was chosen as the main design software.

The authors have developed an algorithm [15] for the formation and filling of a project solutions instances library made in CAD and based on the developed ontological model of the subject area, which consists of the following steps.

1) ***Step 1.*** Getting started, opening a project solution in CAD.

2) ***Step 2.*** Formation of a list of active DSEs included in the assembly of the final product. DSE is associated with the docs project solution opened in CAD KOMPAS.

```
KompasAPI7.Documents docs =
(Manager.kompas.ksGetApplication7() as
KompasAPI7.IApplication).Documents; // assembly
parts list
int max = docs.Count; // parts count
```

3) ***Step 3.*** The model tree and DSE parameters elements set formation. The model tree element is associated with the

topPart variable, and the DSE parameters with the s_param array.

```
ksPart topPart =
doc.GetPart((short)Part_Type.pTop_Part) as ksPart;
// DSE
name_file = Path.GetFileName(topPart.fileName); //
part name
ksVariableCollection vr =
topPart.VariableCollection() as
ksVariableCollection; // parameters collection
int numpart = vr.GetCount(); // parameters count
string[] s_param = new string[numpart]; //
parameters array
```

*4) **Step 4.*** Formation of the project solution construction history, the establishment of relationships between the model tree elements.

```
Part p = new Part(topPart, Manager.treeManager);
// active DSE
Node node = p.GetNode(); // get DSE item
Manager.lastAsctiveNode = node; // last active DSE
item
TreeNode tn =
tvNode.Nodes.Add(Manager.lastAsctiveNode.ToString(
)); // add it to the building history
tn.Tag = Manager.lastAsctiveNode;
Manager.BuildNodeTree(tn.Nodes,
Manager.lastAsctiveNode.ListNode); // relation set
Model.Instance.LoadNode(Manager.lastAsctiveNode);
// history forming
```

*5) **Step 5.*** Retrieving parameters and characteristics for each element of the DSE construction history. The DSE parameters and characteristics are extracted by index into the univar variable and entered into the s_param [i] array.

```
private void params_to_file(int numpart, string
path, ksVariableCollection vr, string[] s_param,
string name_file, string result_file, string
material) {
WriteTextAsync("\r\n", result_file, path);
WriteTextAsync("For detail --- " + name_file + "--
- selected parameters: \r\n", result_file, path);
WriteTextAsync("Materials - " + material + "
\r\n", result_file, path);
for (int i = 0; i < numpart; i++) {
ksVariable univar = vr.GetByIndex(i) as
ksVariable;
s_param[i] = univar.name + " = " + univar.value +
" //" + univar.note + "\r\n";
WriteTextAsync(s_param[i], result_file, path);}}
```

*6) **Step 6.*** Saving the DSE in the file storage of the project solutions instances library. After filling in the project solution template and ontological representation (the generated construction history, parameters, characteristics), new instance of the project solution is saved in the library (updated, if one already exists).

```
void copydoc(string s, string path){
var str = s.Split(new[] { '\\' }).Last();
if (!Directory.Exists(mydocpath + path))
Directory.CreateDirectory(mydocpath + path);
File.Copy(s, mydocpath + path + str, true);}
```

*7) **Step 7.*** Filling in the parameters and characteristics of the DSE in the project solutions instances library. Already completed ontological representation of the project solution detail with the generated fields is considered for further recording of the project solution in the corresponding table of the CAD project solutions instances library.

```
private void insertInDB(int numpart,
ksVariableCollection vr, string dtype, string
material) {
var detail = new DBObject{
Type = dtype,
FileName = name_file,
Material = material,
Parameters = new string[numpart][]
};
for (int i = 0; i < numpart; i++){
ksVariable univar = vr.GetByIndex(i) as
ksVariable;
detail.Parameters[i] = new string[3] { "" +
univar.name, "" + univar.value, "" + univar.note
};
}
using (var db = new
LiteDatabase(@"SolutionStore.db")){
var details = db.GetCollection<DBObject>("Rings");
var parameters =
db.GetCollection<DetailsParameter>("RingsParameter
");
switch(dtype){
case "Ring": details =
db.GetCollection<DBObject>("Rings");
parameters =
db.GetCollection<DetailsParameter>("RingsParameter
");
break;
case "Nut": details =
db.GetCollection<DBObject>("Nuts");
parameters =
db.GetCollection<DetailsParameter>("NutsParameter"
);
break;
………}
```

```
if (details.Exists(Query.EQ("FileName",
detail.FileName))) { details.Update(detail);
}else{ details.Insert(detail);}}
```

*8)* ***Step 8.*** Closing the project solution in CAD, completion of work.

Thus, using the proposed algorithm, the project solutions instances library is formed and populated. Project solutions instances can be reused in the design activities processes with CAD tools.

For the ontology-based project solutions instances library functioning, the following rules must be observed:

1. Upon receipt of a new copy of the project solution in the ontological database, it must be verified [14] in order to coordinate new information with the projects already contained in the library of copies.
2. The verification result may be a modification of new instances of project solutions or the missing information clarification request generation.
3. When searching for a similar project solution with an external request, if it is not in the project solutions instances library, the mechanisms for projects output that are as close as possible in terms of parameters can be used.

The analyzed and completed ontology-based project solution example for CAD KOMPAS in the form of a semantic network is presented below in Fig. 2.

## V. Conclusion

The authors researched and developed the software for the ontology-based CAD subject area project solutions instances library.

An ontological model of the subject area is developed, the main advantage of which is the systematization of information about a complex technical product designed by CAD tools. Its main levels of interaction and key terms are described.

New method allows to create a three-dimensional model building history, highlight key project solution parameters and characteristics, save generated template to the library for reuse by designers in the design process of industrial products using CAD tools. The authors have developed an algorithm for the formation and filling of the library.

At the next stage of the work, the authors will consider the intelligent search system development for project solutions with given parameters.


## Acknowledgment

The reported research was funded by Russian Foundation for Basic Research and the government of the region of the Russian Federation, grant № 18-47-730032.


## References


[1] Hilbert D., Bernays P.: Grundlagen der Mathematik. I. Zweite Auflage. Springer-Verlag, Berlin (1968).

[2] Minsky, M.: A Framework for Representing Knowledge, http://hdl.handle.net/1721.1/6089 last accessed 2020/02/29.

[3] Bashmakov, A., Bashmakov, I.: Intelligent Information Technologies. Publishing House of Bauman MSTU, Moscow (2005).

[4] Karas, S.: Knowledge representation models and cognitive psychology: two sides of the same coin. Tomsk State University Bulletin. Philosophy. Sociology. Political science 2(10), 23-26 (2010).

[5] Veniaminov, E., Mashunina, M..: The principles of constructing an open language of template expressions in the system of knowledge representation. Scientific and technical information. Vol. 2. Information processes and systems 7, 10–17 (2000).

[6] Zagorulko, G., Borovikova, O.: Problems of developing an ontology for an information-analytical Internet resource for decision-making support. In: Information technology and systems: proceedings of the 6th International Scientific Conference, pp. 108-112 (2017).

[7] Murtazina, M.: The Decision Support System in Agile Requirements Engineering Based on OWL Ontology. Vestnik of the Astrakhan State Technical University. Series: Management, Computer Science and Informatics 4, 43-55 (2018).

[8] Voit N., Kirillov S., Ukhanova M., Bochkov S., Ionova I., Brigadnov S.: The method of extracting the design characteristics of the product from the life cycle management systems of complex technical objects. Radioengineering 9 (14), 100-107 (2019).

[9] Voit N., Ukhanova M., Brigadnov S., Kanev D.: Method to Get Assembly Design Parameters. In: Proceedings of the 14th International Conference on Interactive Systems: Problems of Human-Computer Interaction, pp. 82-96. UlSTU, Ulyanovsk (2019).

[10] Voit N., Ukhanova M., Kirillov S., Bochkov S.: Method to Create the Library of Workflows In: Proceedings of the 14th International Conference on Interactive Systems: Problems of Human-Computer Interaction, pp. 97-107. UlSTU, Ulyanovsk (2019).

[11] Ukhanova M.: Ontology-Based Organizational and Technical Components Semantic Model Development. Information and measuring and control systems 6 (11), 98-108 (2018).

[12] Larsson J.: The importance of reuse of project solutions. CAD and graphics 2 (208), 70-73 (2014).

[13] Afanasyev A., Brigadnov S., Kanev D.: Development of an automated system for the analysis of project solutions in CAD KOMPAS-3D. Automation control systems 1 (51), 108–117 (2018).

[14] Galkina E. Analysis of verification tools for project. Scientific and Technical Volga region Bulletin 6, 95–97 (2018).

[15] Voit, N., Ukhanova, M., Brigadnov, S., Kirillov, S., Bochkov, S.: Method for Extracting Product Design Characteristics from Life Cycle Management Systems of Complex Technical Objects. In: Proceedings of The IEEE 13th International Conference Application of Information and Communication Technologies (AICT2019), pp. 1-6. IEEE, Baku, Azerbaijan (2019).

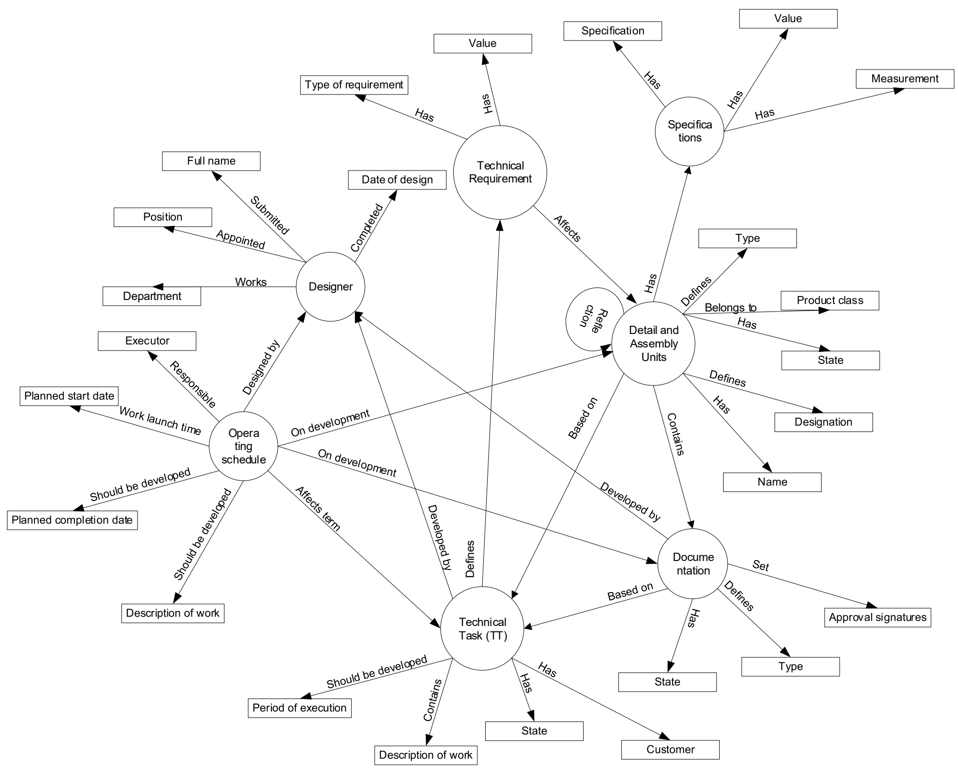


Figure 1. The ontology-based project solution presentation in the form of a semantic network.

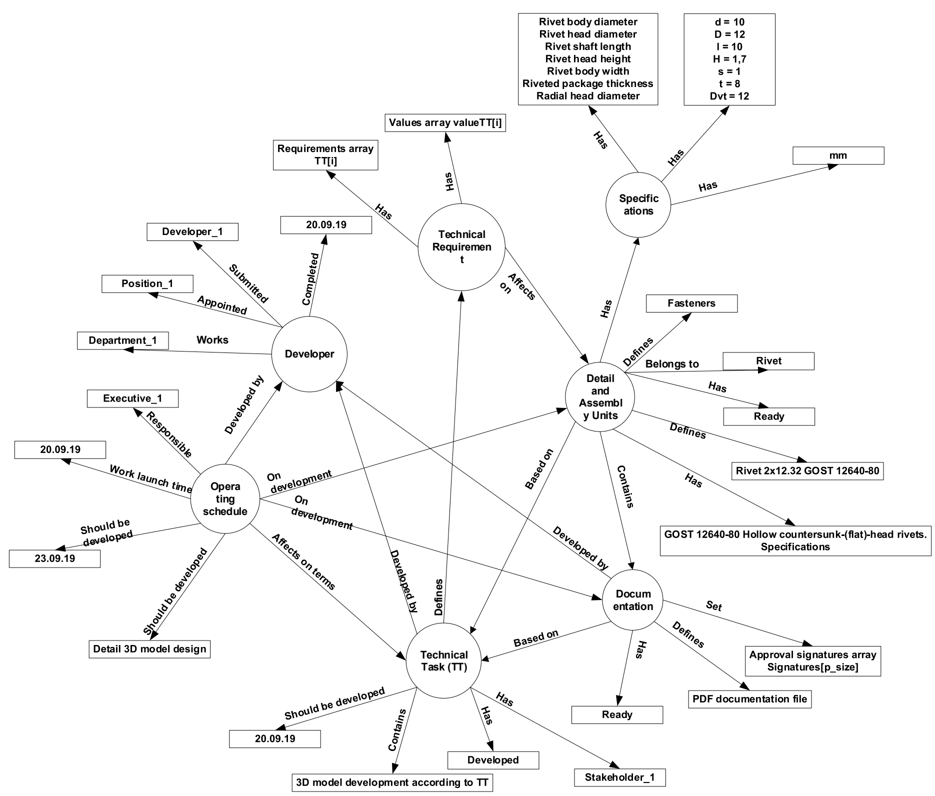


Figure 2. Ontology-based project solution instance example.