Structural-dynamic approach to the formalization of information exchange objects under Integrated Information Environment

IOP Conf. Series: Materials Science and Engineering **537** (2019) 032077 doi:10.1088/1757-899X/537/3/032077

Structural-dynamic approach to the formalization of information exchange objects under Integrated Information Environment

O V Drozd, P A Russkikh, S V Chentsov and D V Kapulin

Siberian Federal University, 79, Svobodny Ave., Krasnoyarsk, 660041, Russia

E-mail: odrozd@sfu-kras.ru

Abstract. The structural-dynamic approach to the formalization of the information exchange objects structure and behavior under Integrated Information Environment is considered: electronic design document and a typical design object. For the objects formalization and modeling the basic of subject-oriented ontologies, automata and set theory were used. The subject area (scope) and the electronic design document structure is formalized in ontology. Graphic representation of the information exchange objects life cycle performed as colored oriented nets. The ontological approach and colored networks allows revealing the relationship between structure and behavior of information objects, avoiding information inconsistency of complex multiply connected structures. The proposed approach provides an opportunity to take into attention the entirety of electronic design documents submission and design objects, which is difficult to ensure using alternative modeling methods.

1. Introduction

The implementation of the microelectronic systems design process involves the active use of information support tools for the mutual integration of software and hardware with the deployment of an Integrated Information Environment (IIE) for enterprise activity organization [1–3]. Modeling of information exchange objects is widely used for formalizing the structure and researching the processes of their life cycle when designing IIE [4, 5]. The information exchange objects under IIE includes electronic design document (EDD) and typical design objects (DO): project, product, electronic product structure (EPS), product information model (PIM) and a set of design documents (DD) for characterizing the product in the project framework.

The initial data for the simulation are survey results of the valid engineering design and lifecycle processes organization control under information structure enterprise-developer of microelectronic systems. The result of the survey is a holistic view of the form and content for used design documents, design objects, as well as their mutual transformations during product design.

For mathematical modeling, simulation of information systems processes and its structure are widely used matrix, finite-automatic and ontological approaches [6–9]. However, these methods does not provide the full completeness of EDD and DO in a heterogeneous information environment [10]. Thereby, this paper proposes a structural-dynamic approach to solving the following objectives related to the formalization of information exchange objects:

- formalization of the structural organization for electronic design document;
- formalization of the EDD lifecycle processes dynamic (behavior);

- formalization of the DO lifecycle processes dynamic inside the microelectronic design activity.

Next, it will be considered specific methods for solving these objectives.

2. Model of the electronic design document structure

The formal description of the electronic design document structural organization is made by using methods of domain-specific ontologies theory. This modeling approach was considered in detail in the paper [10]. In this paper are presented the key details of the EDD structural organization description.

The Electronic Design Document can be represented as the following domain-specific ontology:

$$O = \langle C, I, E, L, F_{ext}, F_{md}, F_s, F_{exc} \rangle, \quad (1)$$

where C – is the set of document categories; I – is the set of information resources that belongs to different categories; E – is the unified EDD exchange format; L – is the set of external design data formats; $F_{ext}(L_i \rightarrow E)$ – is the function of converting data from the external format (i) to the unified; $F_{md}(E \rightarrow L)$ – is the function of EDD converting from unified format to the set of data in external format; $F_s(I_t \leftrightarrow I)$ – is the searching function for information resources by search criteria (t); $F_{exc}(I_j \leftrightarrow I_k)$ – is the function of information exchange between the participants of design process for product (i and j) under IIE.

The Electronic Design Document as an information resource can be represented by:

$$I = \langle p, c, V, D, S, F \rangle, \quad (2)$$

where p – is the information resource name; c – is the information resource category; V – is the set of information resource attribute values; D – is the set of access rights descriptions to information resource; S – is the set of information exchange parameters sufficient to perform the information exchange function $F_{exc}(I_j \rightarrow I_k)$; F – is the set of attached files of the EDD informative part.

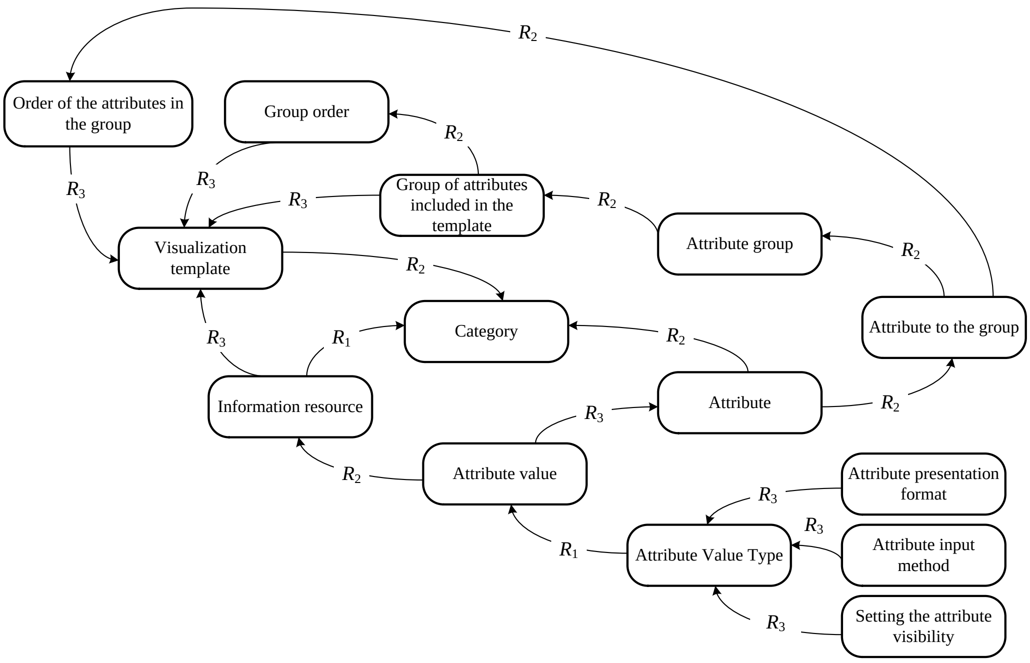


Figure 1. Model of relationships between the elements of the ontology dictionary.

Relationships between dictionary elements that describes the structural organization of the EDD as an information resource can be represented graphically (figure 1), with the following types:

- R_1 – element (A) is an instance of element (B);

- R_2 – element (A) contains a set of values from element (B);
- R_3 – element (A) includes part of element (B).

3. Behavior model of information exchange objects lifecycle

Formalizing the dynamics of design data lifecycle processes, which includes (in addition EDD) design objects, such as PMI, EPS, DD and the product of design, it is necessary to develop the algorithmic basis of software for microelectronic systems design information assistance. The components of EPS are congruence with different composition layers of the microelectronic system structure at appropriate stages of design flow. The set of EPS components constructs a product (the one component of the product set corresponds to several elements of the EPS set).

A typical methodology for modelling the lifecycle processes of design data includes the following stages:

- compiling a list of objects;
- definition of transition operations between the objects lifecycle stages;
- representation the set of transitions as a finite automaton; compiling the matrices of transitions and outputs of the automaton;
- selection of the initial and final vertices of the automaton graph;
- color differentiation of the graph vertices depending on the type of object;
- connecting the vertices of the graph in accordance with the sequence of changes in the stages of the object lifecycle.

Many typical transition operations between the stages of design data lifecycle can be divided into five categories (elementary operations): object creation operation (o_a), object read operation (o_r), object editing operation (o_w), object save operation (o_a), object delete operation (o_d). Execution of all operations presupposes the user has the rights to access the object and the operation itself. As an example, it is considered the operation of creating an object – add new object u_j into the set of specified type objects U by new user p_q from the set of users P in accordance with specified set of object attributes D^* extracted from the set of A^* (object template attributes):

$$o_a\left(D^*, p_q\right): \otimes \mapsto u_i, \otimes \mapsto c_{i1}, \left(a_{it}, d_{i1t}\right) \mapsto \left(a_{it}, d_{it}\right), \quad (3)$$

where $a_{it} \in A$ – is the set of created object attributes; c_{i1} – is the current stage of the object lifecycle in which the new object or its version is created (the point of creation); (d_{i1t}, d_{it}) – is the attribute value sets of an u_i object template and an u_i object instance created.

Typical operations of transitions between the stages of the design data lifecycle and related elementary operations are presented in Table 1. In Table 2 is listed the transition functions between the life cycle stages of an electronic design document. Formalization of the sequence changes for lifecycle stages of typical project objects are presented in table 3 by descriptions of the transition dependences in the lifecycle stages.

Table 1. Typical operations of transitions between the stages of the design data lifecycle.

Lifecycle stage	Elementary operation	Transition to the stage (input action/signal)	
		Name	Designation
1. New	o_a	To Create	x_1
2. Published	o_s	To Publish	x_2
3. On editing	o_w	To Edit	x_3
4. New version	o_a	To Create new version	x_4
5. Approved version	o_s	To Approve	x_5
6. Agreed version	o_r	To Agree	x_6
7. Released version, original	o_s	To Release	x_7
8. In developing, Ready for production	o_r	To Transfer to the next state	x_8
9. Not active	o_w	To Enter, To Disable	x_9, x_{10}

Lifecycle stage	Elementary operation	Transition to the stage (input action/signal)	
		Name	Designation
10. Canceled	o_d	To Cancel	x_{11}

Table 2. The transition functions between the lifecycle stages of an electronic design document.

Lifecycle stage	Transition function to the stage
1. New (v_1)	–
2. In developing (v_2)	$f(v_1, x_1)$
3. Published (v_3)	$f(v_2, x_2) = f(v_9, x_{10}) = f(v_5, x_8)$
4. On editing (v_4)	$f(v_3, x_8) = f(v_7, x_8)$
5. Approved version (v_5)	$f(v_4, x_5)$
6. Agreed version (v_6)	$f(v_5, x_6)$
7. New version (v_7)	$f(v_6, x_4)$
8. Released version (v_8)	$f(v_6, x_7)$
9. Document is not active (v_9)	$f(v_2, x_9)$
10. Document is canceled (v_{10})	$f(v_2, x_{11})$

Table 3. Dependencies of transitions of design objects lifecycle stages.

Type of project object	Lifecycle stage	Transition function to the stage
Product	1. New (q_1)	–
	2. In developing (q_2)	$f(q_4, x_8)$
	3. Ready for production (q_3)	$f(q_{18}, x_8)$
PIM	1. New (q_4)	$f(q_1, x_1)$
	2. Published (q_5)	$f(q_4, x_2)$
	3. On editing (q_6)	$f(q_5, x_3) = f(q_{16}, x_3) = f(q_{14}, x_3) = f(q_8, x_3) = f(q_7, x_3)$
	4. New version (q_7)	$f(q_5, x_4) = f(q_6, x_4)$
	5. Released (q_8)	$f(q_5, x_5) = f(q_7, x_5)$
	6. Agreed (q_9)	$f(q_8, x_6)$
	7. Original (q_{10})	$f(q_{17}, x_2)$
EPS	1. New version (q_{11})	$f(q_9, x_1)$
	2. Released (q_{12})	$f(q_{16}, x_5)$
DD	1. New (q_{13})	$f(q_{11}, x_1)$
	2. Published (q_{14})	$f(q_{13}, x_2)$
	3. New version (q_{15})	$f(q_{14}, x_4)$
	4. Released (q_{16})	$f(q_{14}, x_5) = f(q_{15}, x_5)$
	5. Agreed (q_{17})	$f(q_{16}, x_6)$
	6. Original (q_{18})	$f(q_{10}, x_7)$

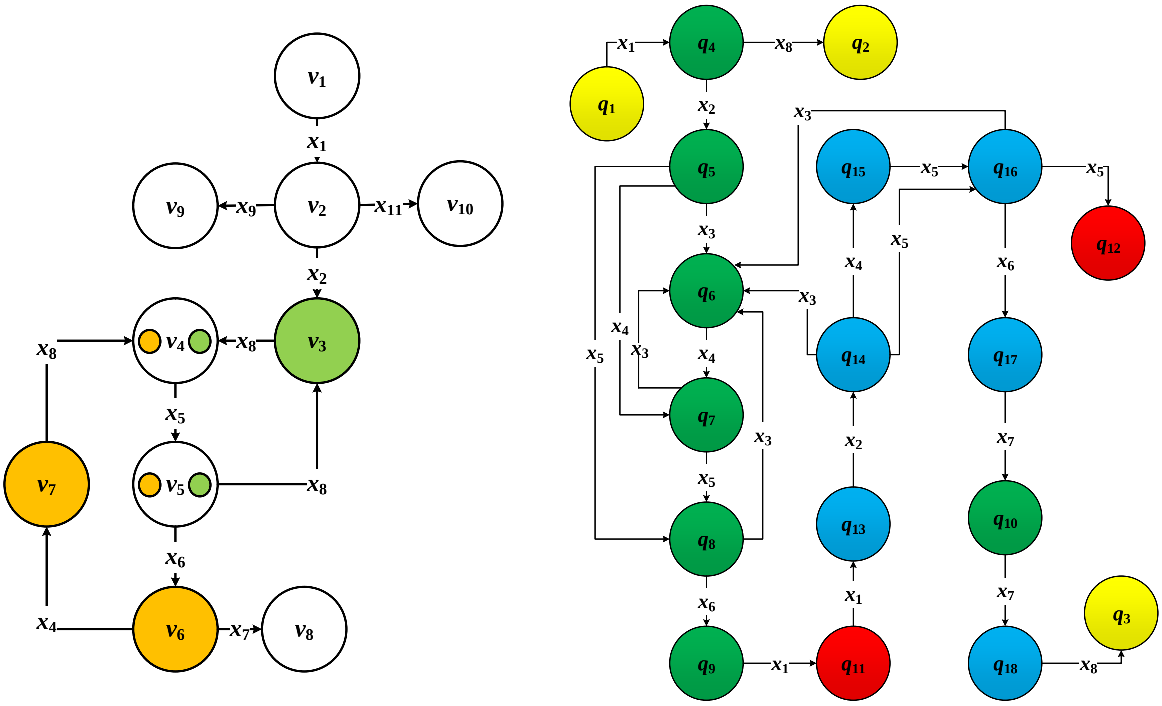
The initial vertex of EDD lifecycle graph is «EDD – New», the final vertex is «EDD – Released Version» and «EDD – Document is canceled». By connecting the vertices of the graph in the direction of design flow from the initial vertex to the final vertex, we obtain a model of the EDD lifecycle stage changes (figure 2, a).

The resulting graph contains the following closed contours, the movement of which means changes in EDD with creating new version:

- «Published» → «On editing» → «New version» (marked by light green) – is active when changes in EDD are rejected;
- «On editing» → «Approved version» → «Agreed version» → «New version» (marked by orange) – is active when creating a new version of the approved document.

In order to increase the visibility stages changes of design flow, the vertices of the graph should be colored in accordance with the type of design object: product – yellow, PMI – green, EPS – red, DD –

blue. The vertex «Product → New» is taken as the initial point of the graph, and «Product → Ready for production» as the final point. The resulting graph is shown in the figure 2, b.



(a) (b)

Figure 2. Dynamics of lifecycle stage changes for EDD (a) and typical design objects (b).

The proposed EDD and design object models can be implemented programmatically using the following typical behavioral design patterns:

- «State» – controlled object must change its behavior depending on its state.
- «Observer» – defines a one-to-many relationship between controlled objects, and when a state of any object changes, all depended objects are notified of this event.

4. Conclusion

The structural-dynamic approach to the formalization of the information exchange objects structure and behavior under Integrated Information Environment for enterprise-developer of microelectronic systems is proposed. The formalization of such objects as the electronic design document, the electronic product structure, and the product information model are performed by the basic of subject-oriented ontologies, automata and sets theory.

The domain-specific ontologies for development structure model of an electronic design document made it possible to avoid information inconsistency of complex multiply connected structures. Relationships between different types of design objects were adequately showed in the colored oriented graph, which can be considered as a graphical representation of the design flow as a whole. The proposed approach provides an opportunity to take into attention the entirety of electronic design documents submission and design objects, which is difficult to ensure using alternative modeling methods.

Acknowledgments

This work was supported by the Ministry of Education and Science of the Russian Federation in the framework of the Federal target program «Research and development of priority directions of development of the scientific-technological complex of Russia for 2014-2020» (agreement № 14.575.21.0142, unique ID project RFMEFI57517X0142).

IOP Conf. Series: Materials Science and Engineering **537** (2019) 032077 doi:10.1088/1757-899X/537/3/032077

References

[1] Bruun H P L, Mortensen N H, Harlou U, Wörösch M and Proschowsky M 2015 PLM system support for modular product development *Comput. Ind.* **67** 97–111
[2] Messaadia M, Belkadi F, Eynard B and Sahraoui A-E-K 2012 System Engineering and PLM as an integrated approach for industry collaboration management *IFAC Proc. Vol.* **45** 1135–40
[3] Chang K-H 2014 *Product Design Modeling Using CAD/CAE* (Elsevier) pp 15–9
[4] Makarova I, Shubenkova K and Pashkevich A 2018 Improving Reliability Through the Product's Life Cycle Management *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)* (IEEE) pp 154–9
[5] Glock T, Sillman B, Kobold M, Rebmann S and Sax E 2018 Model-based validation and testing of industry 4.0 plants *2018 Annual IEEE International Systems Conference (SysCon)* (IEEE) pp 1–8
[6] Lennartson B, Bengtsson K, Wigstrom O and Riazi S 2016 Modeling and Optimization of Hybrid Systems for the Tweeting Factory *IEEE Trans. Autom. Sci. Eng.* **13** 191–205
[7] Khodayi-mehr R, Aquino W and Zavlanos M M 2019 Model-Based Active Source Identification in Complex Environments *IEEE Trans. Robot.* 1–20
[8] Lavrishcheva E 2015 Ontological approach to the formal specification of the standard life cycle *2015 Science and Information Conference (SAI)* (IEEE) pp 965–72
[9] Mao X, Li Q, Zhang Z and Zhu Q 2009 Application of Spatial Information Search Engine Based on Ontology in Public Health Emergence *2009 3rd International Conference on Bioinformatics and Biomedical Engineering* (IEEE) pp 1–4
[10] Drozd O V and Kapulin D V 2017 The model of electronic design document as a part of integrated information environment of radioelectronic enterprise *2017 Dynamics of Systems, Mechanisms and Machines (Dynamics)* (IEEE) pp 1–5