

Data Conversion Model Using the Principles of Geometric Model Structure Proximity Comparison

Alexey Boytyakov¹ and Alexandr Filinskikh¹

¹Nizhny Novgorod State Technical University n.a. R.E. Alekseev, 24 Minin Str., Nizhny Novgorod, 603950, Russia

Abstract

Research has been carried out into possible data conversion losses and the integration of heterogeneous automated systems in enterprises. A model of data conversion in the framework of heterogeneous automated systems interaction on the example of geometrical model structures comparison of heterogeneous automated systems is proposed. The model can be used for loss estimation using the representation of geometric models as data structures and conversion metrics. The article deals with the problem at the current stage of information support for lifecycles processes is the lack of integration of multi-vendor automation systems in enterprises. Losses in one stage of the lifecycle can lead to technical and economic difficulties in other stages of the lifecycle and problems can also be encountered when integrating automation systems and data conversion between enterprises. There is a need to develop an advanced parameter conversion model and compare the proximity of GM structures between automation systems. It is required to evaluate the efficiency of data conversion between environments using different formats using metrics.

Keywords

Geometric model, data format, graph, graph structure, parameter classification, CAD system, PDM system

1. Introduction

Today, the principle of a unified information space (UIS) [1] is one of the priority processes in the development of Russian industrial enterprises. The situation in which there is no full automation of the life cycle stages is common. In this case, software products of different foreign and domestic vendors, the formats of which may be incompatible, are used. At different stages of product lifecycle (LCL), including design and development processes, the use of different formats can lead to additional time and financial expenses for data conversion. The current trend at Russian enterprises is towards import substitution of foreign vendors with domestic ones.

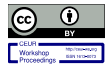
The main task of product lifecycle information support technologies (CALs or IPI) is to create an UIS for all participants of the product lifecycle (LCL), which ensures information interaction between CALs components. A distinctive feature is the extensive use of digital information model (DIM) of the product and its components at most stages of the lifecycle. The base of DIM is a combination of geometric model (GM) and attributive information. The components of DIM include computer-aided design system (CAD), as well as product data management system (PDM) [2]. International standards have been developed for universal interaction of design and manufacturing automation systems (STEP, IGES).

The main problem at the current stage of information support for lifecycles processes (Figure 1) is the lack of integration of multi-vendor automation systems in enterprises [3]. For example, losses in one stage of the lifecycle can lead to technical and economic difficulties in other stages of the lifecycle.

GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27-30, 2021, Nizhny Novgorod, Russia

EMAIL: alexey.boytyakov@gmail.com (Alexey Boytyakov); alexfil@yandex.ru (Alexandr Filinskikh)

ORCID: 0000-0002-6477-9303 (Alexey Boytyakov); 0000-0003-3826-6771 (Alexandr Filinskikh)



© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

Problems can also be encountered when integrating between enterprises: further design of products by other enterprises in other formats can lead to significant time and resource costs, or even lead to re-development of the product [4]. To date, neutral data conversion formats have been developed for data formats from different vendors: e.g. STEP, IGES, etc. [5]. But with these formats it is not possible to transfer all geometric model (GM) parameters without losses [6]. So there is a need to develop an advanced parameter conversion model and compare the proximity of GM structures between automation systems. It is required to evaluate the efficiency of data conversion between environments using the above mentioned formats using metrics. These topical issues have become the subject of our research.

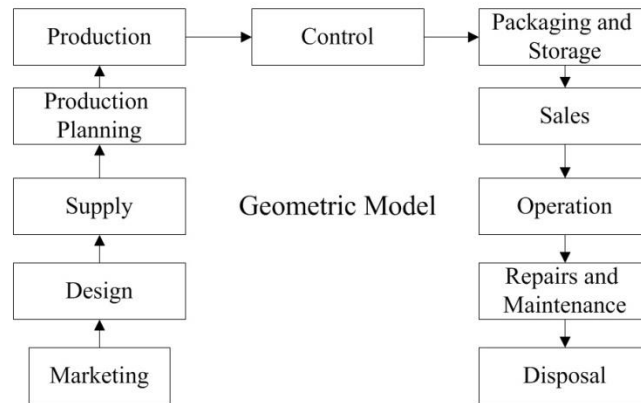


Figure 1: Information support for product lifecycle stages

2. Problems of the data conversion process between automated systems

An important role in the design is played by automated systems, which include computer-aided design systems, product data management, and others. An example of geometric model in automated system is shown in Figure 2. These systems carry out the calculations necessary for the engineer during the development of the product model in CAD through the data that is located in the PDM. If there is a need to calculate the behavior of products, such a system can be connected to PDM [7], using specialized engineering analysis systems. When interacting with PDM, CAD will have access to the results of the operation of these systems.



Figure 2: An example of geometric model in automated system

There are several levels of interaction of heterogeneous automated systems [8]. The highest level is when a single data model is used in the whole enterprise. All computer systems (CAD, PDM, automated

enterprise management system (ASUP), etc.) work with a single database. But to implement this level of the systems' interaction is very difficult in practice.

Another level of interaction uses direct access to the database. All systems have their own databases, each can send and receive data from other systems (the method is found in practice: for example, the Tflex Docs PDM system has a mechanism for its implementation).

The main problem of this level of interaction is that the manufacturers usually offer specific solutions. There are no universal solutions [9], the integration of systems is hidden, so there is no way to define a more universal system. The interaction of heterogeneous automated systems can be carried out via the application programming interfaces (APIs), as represented in Figure 3 [10]. When implementing a large number of systems in an enterprise, a large number of converters are required to ensure data conversion. It obviously leads to considerable raise of implementation costs. The disadvantages also include the need for a complete reworking of the software, in case it is necessary to replace one of the systems with a system from another manufacturer, or when changing the API of any of the systems.

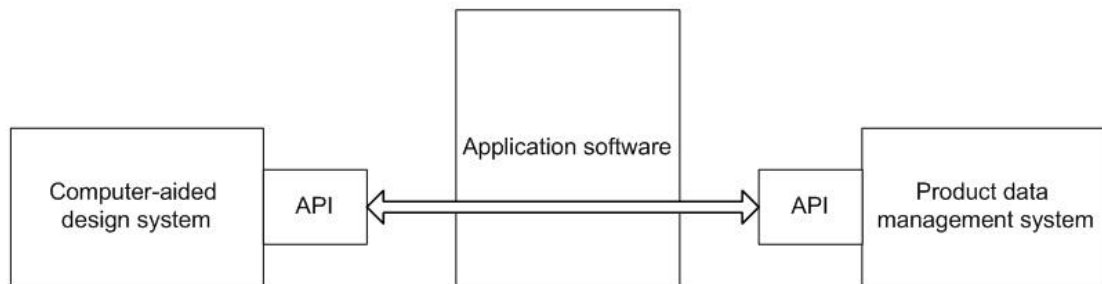


Figure 3: Interaction of the automated systems via the APIs

There is also a concept of a Unified Information Space (UIP), which includes the concept of PLM technologies [11] and the concept of IPI technologies [12]. This concept involves the use of files for data exchange between systems. When converting, the first system generates a file that contains the transmitted data, and the second system reads this file after receiving the data. To create a file, special converters are used that convert the data from the application system format to the exchange file format and vice versa. When choosing formats, it is possible to use a neutral format, the ISO 10303 STEP standard [13].

The concept of PLM is to perform tasks using a set of software products from a single developer. However, there may be a situation in which an engineer cannot replace the program with another vendor, but only the entire complex. On the other hand, the use of systems of independent vendors can lead to the problems with the data integration and transfer, i.e. the possibility of data conversion without significant losses.

The concept of IPI technologies is to free the user from a single developer, using a neutral data conversion format (Figure 4). This approach is based on a unified information space UIP (an international term is shared data environment, SDE), which is implemented using international data presentation standards. The IPI strategy includes information support for the product lifecycle based on the use of an integrated information environment, paperless presentation of information, the use of electronic digital signatures, standardization of information descriptions of management objects, improvement of business processes, parallel engineering, parallelization of a number of design works and stages of the product lifecycle, and others [14].

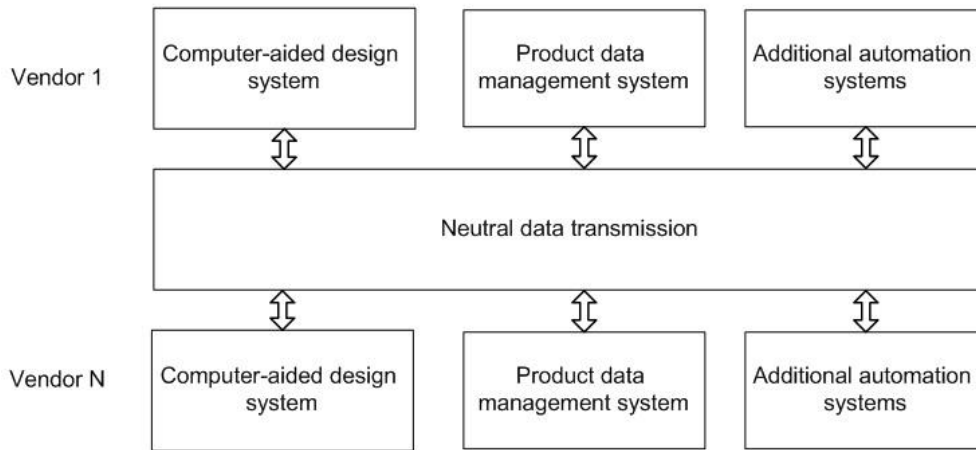


Figure 4: The concept of IPI technologies in the UIP (SDE) organization

One of the main stages in the implementation of the IPI strategy is the creation of the unified information space of the enterprise, which is based on interacting CAD and PDM [15].

In world practice, there are many examples of successful application of the IPI concept at enterprises of various industries [16]: aircraft construction, automotive industry, mechanical engineering, medicine. In Russia, for example, JSC "Tupolev", Voronezh Mechanical Plant, AVKP "Sukhoi" and others have successfully implemented the IPI concept in their production cycles.

Open distributed automated systems for design and management at industrial enterprises are the basis of modern IPI technologies. The main problem is the transition to a uniform description and interpretation of data, as well as regardless of the location and time characteristics of their receipt in the system, which may have global scales.

3. Methods for Comparing the Proximity of Data Structures

Each product can have a tree structure, which is a graphical representation of the hierarchical structure. The principles of use in the lifecycle stages and operation of products involve checking at each stage how the structure has been changed. So to compare structures, it is proposed to apply graph theory to describe the methods for comparing the proximity of data structures [17].

In the following, examples of the product model that have been applied to analyse the conversion process are discussed. Initially the model is a GM which is some product, and the product needs to be converted to another vendor's automation system. The final model represents some outcome of the conversion to another automation system developed by another vendor, i.e. a set of operations associated with the conversion process and with the GM data is identified at the output.

It is required to identify probable difficulties in converting a GM within a data operation in heterogeneous automation systems using graphs and mathematically propose a description in the form of a "tree". We have created the structure of GM using graphs, presented as a set of elements for the product model. A graph is known to be a mathematical object, a complex of two sets which are a set of elements including a variety of edges and vertices. This set of elements of the product model includes integration parameters, geometry parameters and such data as attributive information [18] and proposed structure containing frames [19] and product tree. The integration parameters include a number of information such as: information about the open and vendor-supported CAD or PDM API [20]; presence of CAD API functions for creating, converting and synchronizing properties and attribute information of CAD files; presence of PDM functions for structured loading/unloading, tracking and managing CAD data, etc. The GM is represented by a structure comprising a product tree and frames containing data about GM parameters. The structure of the product model is denoted as graph $G = (X, A)$. The graph can include versions of the above-mentioned elements as well as their characteristics.

There are several stages of comparison. The first is a proximity comparison of GM trees only, based on a mathematical representation of trees in the form of adjacency matrices. The adjacency matrix is a square matrix with logical values (0 or 1). The graph consists of vertices and edges, which are links

between the vertices. So the data on GM parameters is reflected in the presence of the graph edges and also in the vertices where the information is contained first in the case of tree graphs of a product model structure. If a vertex of the tree graph is lost, an edge is also lost. The following is a description of a part of the the assembly of the original GM shown in Figure 5.

The adjacency matrix is a binary square matrix, with rows and columns having values of 1 or 0, the number of rows being matched to the number of columns. The matrix has dimension $n \times n$, (where n is the vertices of the structure as a graph), uniquely representing its structure. This is one of the variations of graph structure as a matrix. The first row and the first column, which do not consist in a matrix but are written down for ease of perception, contain the numbers at the intersection of which each of the elements is located and determine the index value of the latter [21].

$A = \{a_{ij}\}$, $i, j = 1, 2, \dots, n$, so each element of the matrix is defined as follows:

$a_{ij} = 1$, if there is an arc $\exists (x_i, x_j)$; $a_{ij} = 0$, if there is no arc (x_i, x_j) .

Such binary matrices are used to parse the conversion process and to identify unobservable differences in graph structure. In the context of the conversion assessment task, this is to identify the difference in structure of the product model after the data conversion process within a multivendor framework. A matrix representation of the graphs is used to compare them. Algebraic operations are performed with the matrices to reveal the result of how similar or different the graphs are. The adjacency matrix of the original GM as well as the binary values of the product model are shown below.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} & a_{49} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} & a_{59} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} & a_{69} \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} & a_{79} \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\ a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & a_{99} \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 && 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (1)$$

Next, the GM was converted and then transferred to another vendor's automation system, resulting in some collisions. For clarity, a part of the converted GM assembly is shown in Figure 5.

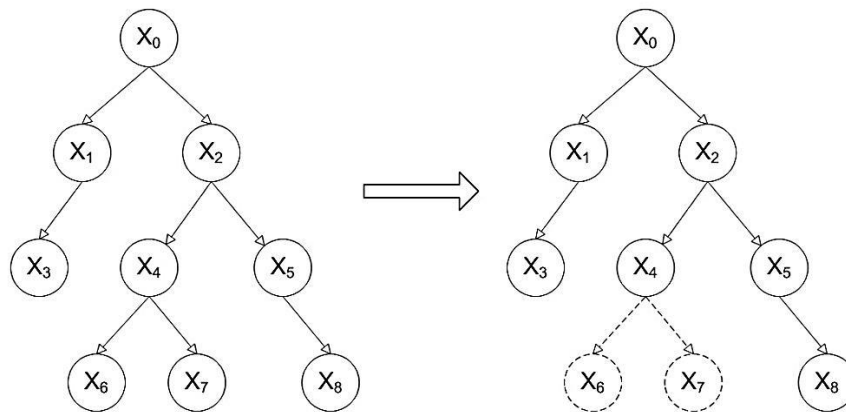


Figure 5: An example of the original GM and a possible result of the GM conversion

As depicted in Figure 5 and Figure 6, the conversion process reveals some losses as part of the data transfer to another vendor's automation system. The binary square matrix of the transferred GM has the same size as the original GM, as the transferred GM is compared to the original GM. The size of the binary square adjacency matrix is determined by the number of vertices in the graph, so a procedure is required to add zero rows and columns to the so-called "right places" (lost data), which must first be determined. The following describes the part of the GM assembly after data conversion.

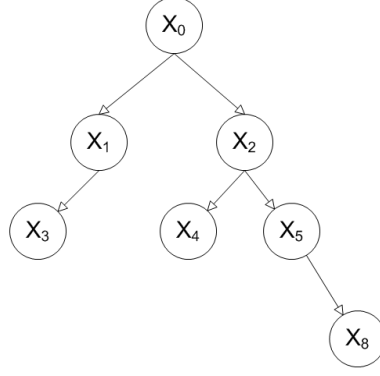


Figure 6: Final result of the GM conversion

The adjacency matrix and binary matrix values of this GM are as follows:

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} & b_{17} & b_{18} & b_{19} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} & b_{27} & b_{28} & b_{29} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} & b_{37} & b_{38} & b_{39} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} & b_{46} & b_{47} & b_{48} & b_{49} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} & b_{56} & b_{57} & b_{58} & b_{59} \\ b_{61} & b_{62} & b_{63} & b_{64} & b_{65} & b_{66} & b_{67} & b_{68} & b_{69} \\ b_{71} & b_{72} & b_{73} & b_{74} & b_{75} & b_{76} & b_{77} & b_{78} & b_{79} \\ b_{81} & b_{82} & b_{83} & b_{84} & b_{85} & b_{86} & b_{87} & b_{88} & b_{89} \\ b_{91} & b_{92} & b_{93} & b_{94} & b_{95} & b_{96} & b_{97} & b_{98} & b_{99} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (2)$$

4. Application of metrics in determining the proximity of data structures

We carried out a study of the proximity of GMs using graph theory. The structure of GM products and transfer results are mathematically represented in the form of a graph and contain groups of parameters previously described in more detail, including integration data. This structure is the source of data for determining the structural weights of GM elements [22]. There is another option to improve the accuracy of data conversion estimation, which requires additional conversion data for each node of the GM tree. Here a graph structure [23] of GM transfer parameters is applied. The graph structure includes a tree-like graph and a data set for each GM node, within a frame data representation, containing a list of GM parameters. The initial layer contains the GM parameter data for the whole product, represented as a tree view. In addition, each node of the next level contains an additional list of parameters, represented as frames. Post-conversion comparisons were considered within the assembly, within each individual structure level and at the node level. The data structure of the original GM is shown in Figure 7.

The proximity of the graphs is calculated by applying a metric based on the Hamming distance expression if nominal conversion data is required:

$$d_{ij} = \sum_{k=1}^p |x_{ik} - x_{jk}|, \quad (3)$$

We get the following expression:

$$d_{ij} = \sum_{i=1}^n \sum_{j=1}^n |a_{ij} - b_{ij}|, \quad (4)$$

where a – parameters of the 1st GM of i -th row and j -th column; b – parameters of the 2nd GM of i -th row and j -th column obtained after conversion; n – number of elements.

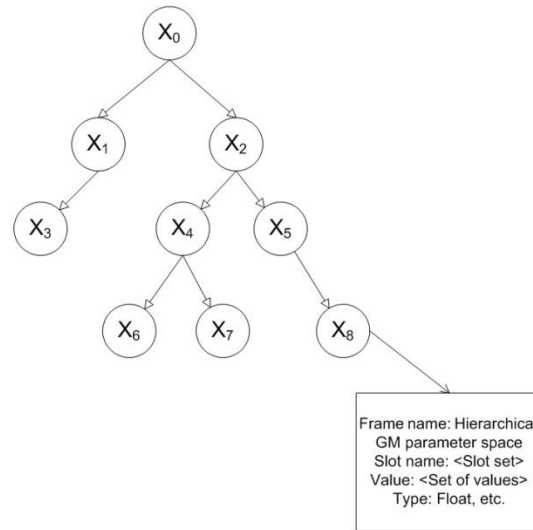


Figure 7: An example of graphical structure of a GM

Another calculation of the proximity of graphs using the metric is based on the Sorensen measure if quantitative conversion data is required:

$$K_S = \frac{2c}{a + b}, \quad (5)$$

where a – number of parameters of the 1st GM, $a = \{X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9\}$, b – number of parameters on the 2nd GM as a result of the conversion, $b = \{X_1, X_2, X_3, X_4, X_5, X_8, X_9\}$, c – number of parameters common to the 1st and 2nd GM, $c = \{X_1, X_2, X_3, X_4, X_5, X_8, X_9\}$.

The problems when converting product models very often do not depend linearly on the number of elements in the GM, but on the formats and vendors of the design automation systems. Therefore it was necessary to find out possible data loss during conversion under conditions of different software vendors and to what extent it is possible to apply neutral formats for data conversion for different software vendors. The conversion experiments with neutral formats yielded metric values based on a comparison of the proximity of the GM graphs from 0 to 0.5. The value for each vendor will be different, so each case should be considered in detail: it is necessary to assess how satisfied the obtained result is, what were the conversion losses, what additional recovery costs will be required. It was found that when using engineering automation systems of a single vendor the conversion problems are not format dependent but rather random. If production plants use software from different vendors, the dependence on vendor formats was found. Often the different formats are incompatible, resulting in more data loss and higher recovery costs. It was found that it is possible to use neutral formats, under certain conditions: for example, when the losses are not great and will not affect the further development of the product and work with the product model. Proper evaluation of data conversion losses should have a positive impact on the further support of the product life cycle stages.

5. Conclusion

The lack of a model to support data conversion between automation systems in the form of a generalized machine-independent model based on a comparison of the proximity of graphs and graph structures is detected. Our proposed model makes it possible to estimate the labor intensity of data recovery if there have been losses during data conversion using graphs and graph structures. We propose a methodology for obtaining the values of the metric for estimating data conversion losses. The principles and problems of integration of automation systems and product data management systems

are revealed. The estimation of data conversion in the interaction of heterogeneous automated systems within the framework of UIP based on metric estimates is proposed.

6. References

- [1] Norenkov I.P., Kuzmin P.K., Information support for knowledge-intensive products CALS technology, Bauman Moscow State Technical University Publishers, Moscow, 2002, 320 p. (in Russian).
- [2] Norenkov I.P., Fundamentals of computer-aided design, Bauman Moscow State Technical University Publishers, Moscow, 2002, 336 p. (in Russian).
- [3] Kutin, A., Dolgov, V., Sedykh, M., Ivashin, S., Integration of Different Computer-aided Systems in Product Designing and Process Planning on Digital Manufacturing. *Procedia CIRP* 67, 2018, pp. 476-481. doi:10.1016/j.procir.2017.12.247.
- [4] Filinskikh, A.D., Component approach to the translation of geometric models. 8th International Scientific Conference on Computing in Physics and Technology, 2020, pp. 220-225. doi:10.30987/conferencearticle_5fce27710c7721.55039399.
- [5] Yablochnikov E.I., Fomina Y.N., Salomatina A.A., Computer Technologies in Product Life Cycle, SPbSU ITMO, SPb, 2010, 180 p. (in Russian).
- [6] Shilolitsky O., How to re-invent CAD / PDM integration, 2014 URL: <http://beyondplm.com/2014/05/19/how-to-re-invent-cad-pdm-integration>.
- [7] Koucky S., Essentials of managing product design data, 2001. URL: <https://www.machinedesign.com/archive/article/21816798/essentials-of-managing-product-design-data>.
- [8] Filinskikh A.D., Analysis of the state of project management systems at Russian enterprises, Bulletin of the Belgorod State Technological University named after V. G. Shukhov, No. 1, 2011, pp. 162-167. (in Russian).
- [9] Gujarathi G.P., Ma Yongsheng, Parametric CAD/CAE integration using a common data model, *Journal of Manufacturing Systems*, 30(3), August 2011, pp. 118-132. doi:10.1016/j.jmsy.2011.01.002.
- [10] Lee K. Fundamentals of CAD (CAD/CAM/CAE), St. Petersburg: Piter, 2004, 560 p. (in Russian).
- [11] Grieves M.W., Product lifecycle management: the new paradigm for enterprises (2005) *International Journal of Product Development*, 2 (1-2), pp. 71-84. doi: 10.1504/ijpd.2005.006669.
- [12] Zakovryashin A.I., IPI technology for creating high-tech products, *Electronic journal "Trudy MAI"*, No. 49, 2011, URL: <http://trudymai.ru/upload/iblock/c01/ipi-tekhnologiya-sozdaniya-naukoemkikh-izdeliy.pdf> (in Russian).
- [13] ISO 10303-1-1994. Industrial Automation Systems and Integration-Presentation and exchange of product data-Part 1: Overview and basic principles, Geneva: ISO, 1994, 17 p.
- [14] Levin A.I., Davydov A.N., Barabanov V.V., Concept for the development of CALS technologies in Russian industry, Moscow: Research Center of CALS-technologies "Applied Logistics", 2002, 130 p. (in Russian).
- [15] Shalumov A. S., Nikishin S.I., Noskov-Kovrov V. N., Introduction to CALS-technologies, Textbook. Manual, Kovrov state technol. akad., 2003, 184 p. (in Russian).
- [16] Tutt W., Graph Theory, Translated from English: Mir, Moscow, 1988, 424 p. (in Russian).
- [17] Filinskikh A.D., Sosnina O.A., Boityakov A.A., Hierarchical space of geometric model parameters, *Bulletin of Belgorod State Technological University n.a. V.G. Shukhov*, No 2, 2015, pp.131-134. (in Russian).
- [18] Boityakov A.A., Generalized model of data transfer between CAD- and PDM-systems using frames, *Bulletin of Belgorod State Technological University n.a. V.G. Shukhov*, No 3, 2015, pp. 115 - 119. (in Russian).
- [19] Golitsyna T.D, Issues of integration of product data management systems (PDM) and CAD, *Scientific and Technical Bulletin of SPbSU ITMO - Information Technologies*, No 6, 2009, pp. 543- 547. (in Russian).
- [20] Bertsun V.N., Mathematical modeling on graphs. Part 2, Tomsk State University Press, Tomsk, 2013. 88 p. (in Russian).

- [21] Filinskikh A.D., Byashev A.Kh., Analysis of parametric and graphic information transfer based on experimental data, Bulletin of Belgorod State Technological University n.a. V.G. Shukhov, No 2, 2012, pp. 164-166. (in Russian).
- [22] Kalinina N.A. Models and Procedures of Hierarchical Network Representation of the Subject Area to Support Knowledge Acquisition Processes, D. thesis for the degree of Candidate of Technical Sciences. Nizhny Novgorod State Technical University, N. Novgorod, 2018, 180 p. (in Russian).