ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20190923203841
Original scientific paper

Application for Product Functional Model Creation

Nenad BOJCETIC, Filip VALJAK, Stjepan FLEGARIC, Mario STORGA

Abstract: The research presented in this article tackles the problem of creation of functional models by students and young engineers. The problem of usage of paper-pen approach is analysed and described. As a solution the creation of the program application is introduced and described. The application introduces a visual aid for better understanding of available functions and flows. The experiment was conducted to validate the application.

Keywords: design education; design support tool; functional basis; functional modelling

1 INTRODUCTION

One of the best ways to popularize the usage of scientifically developed methods and methodologies is through education. If students or young designers are exposed to usage of a specific method or methodology, it is more likely that they will embrace it and use it during their professional work. One such method, in the conceptual design phase, is a creation of product functional models. Product functional models can be perceived as a bridge between the problem and solution space [1]. Results of functional model's research, conducted during the last sixty years, can be seen through evolution and representation variation among the models. The conducted research was mainly focused on standardization of functional vocabulary and applications of functional modelling, [2-8]. Many of the prominent functional modelling schemes, like [9, 10], and [11] are based on the [12] flow-based reasoning, supported by representations of the flows of material, energy, and information through the system.

Every functional modelling scheme introduces its vocabulary of functions and flows, and this creates lots of ambiguity [13]. The ambiguity can be an issue especially when students are asked to create their functional models. When students are given freedom to use vocabulary that is natural for them, it is tough to "read" the functional models and to understand them, both by the educators and by their fellow collegues. Eisenbart [8] states that "this ambiguity is problematic in the collaboration of different designers because it may considerably hinder communication about individual functions and expected system functionality". That is why, in students works, modelling scheme was restricted to just one. The scheme that was selected is the functional basis introduced by [9] and it will be used for the creation of functional models for student's assignments. Restricting the students' usage of different modelling schemes to just one was necessary to bring uniformity to their works so that all students' solutions can be interpreted in the same way.

This step increased the level of understanding functional models created by students (using paper-pen approach), but students were facing a problem of comprehending functional basis functions and flows of actual meaning. So, there were a significant number of misunderstandings and misused terms. To tackle this problem, we created an application to support the creation of uniform functional models based on the Hirtz functional basis representation [18]. The developed application introduces visualization aid to help students to understand better the terms used in functional models' creation. Visualization aid is implemented through the usage of images and short movies that can closely explain the term from the vocabulary. One of the advantages from student usage of developed application would be more uniform models.

The questions that govern the work presented in this paper are "does restricted usage of terms, imposed by the application, improve the readability of models?" and "does restricted terms usage, imposed by the application, gives more readable models?". To be able to give the answers to these research questions, we conducted an experiment.

In the second chapter, a brief introduction to the functional basis [9] is given following the developed application description. The fourth chapter introduces conducted experiment explaining how the experiment was conducted. The paper concludes with a discussion and the conclusion.

2 BACKGROUND AND RELATED WORK

In engineering design, function structures are used in modelling and reasoning about the product in the conceptual phase [14]. This leads to producing a model which represents the roles of its individual components that will be embodied in the final product. [11, 12, 15, 16]. During years of research on function modelling function models have significantly evolved, resulting in different representations of functions, now found in literature, ranging from the functional basis approach [9, 17, 18] to affordance-based function modelling [19, 20]. Many of the well-established function modelling schemes follow the flow-based thinking [12], underpinned by representations of the flows of material, energy, and information through the system [9-11].

In this work we are focusing on the usage of the functional basis introduced by Stone & Wood [9], where authors attempted to create a common design language to be used with functional models, focusing primarily on the mechanical and electromechanical domains. Stone and Wood through the usage of proposed "universal language" were trying to contribute specifically to the following product design areas:

- Product architecture development,

- Systematic function structure generation,
- Archival and transmittal of design information,
- Comparison of product functionality,
- Creativity in concept generation,
- Product metrics, robustness and benchmarks.

In their work, Stone & Wood [9] emphasize one crucial requirement of the functional basis, that functions must be expressed as a verb-object pair. The basic functions fill the verb spot, and the basis flows provide the object. Later work [21] was focusing on reconciliation and evolution of the functional basis. In the reconciled functional basis, they label three levels. According to Hirtz [18], "These different levels of functional specification are important for several reasons. In the design of new products, the customer needs, and thus functional requirements, are more difficult to ascertain than in a redesign or evolutionary design effort."

Class (Primary)	***Secondary***	***Tertiary***	***Correspondents***
Branch	Separate		Isolate, server, disjoin
		Divide	Detach, isolate, release, sort, …
		Extract	Refine, filter, purity, percolate, …
		Remove	Cut, dispel, disperse, diverge,
	Distribute		Diffuse, dispel, dissipate, …
Channel	Import		Form entrance, input, capture, …
	Export		Dispose, eject, emit, empty, …

Figure 1 The hierarchical relationships between levels of specification in the functional basis [9]

In general, ambiguous customer needs should result in the use of higher-level functions. A more specific customer needs to lead to the use of more specific types of functions. The first level is class (primary), the second one is secondary, and the last one is tertiary. They retained the name of the first level class because of its common usage in functional modelling literature. It is worth to mention that lower levels provide a more specific definition, thus leading to particular technologies or physical principles. The hierarchical relationships between levels of specification in the functional basis are shown in Fig. 1.

More recently, researchers have started discussing an additional benefit of function modelling, a shared understanding between collaborating designers from different disciplines [22, 23]. This is one more aspect that we intent to address by upgrading the application after we analyze a prototype.

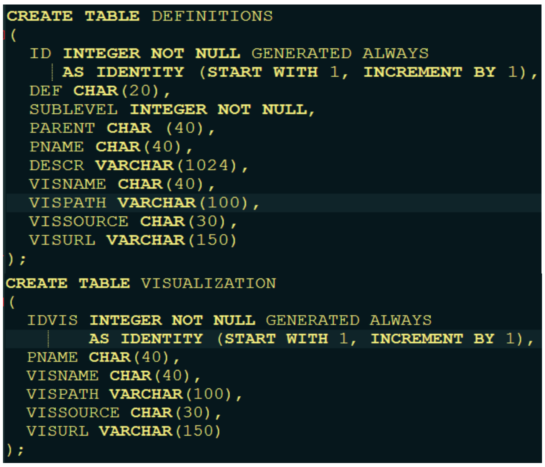
3 THE APPLICATION ARCHITECTURE AND DEVELOPMENT

Before the decision to create the presented (described) application, the extensive search for an existing solution was done in the hope to find any application for the creation of functional models. What we found is that in the last fifteen years there was no application for the creation of functional models proposed. So, we broadened our search, and we found a couple of articles describing software tools for the creation of functional models [24-26]. The most interesting article was about FunctionCAD [27]. The FunctionCAD is a functional modelling application based on integrated functional and process modelling within the Function Design Framework [25]. FunctionCAD is based on a modular architecture and utilizes a plugin interface to increase versatility and allow for future expandability. Unfortunately, neither the application nor the main library (libFCAD) is available. The functional models are, mostly, created using software solutions for the creation of various types of diagrams (like Microsoft VISIO) and specialized software solutions for the creation of SysML (Systems Modelling Language) and UML (Unified Modelling Language) diagrams or by using matrices [28, 29].

Thus, we decided to create our own application for functional models' creation. As mentioned earlier, the basic idea for the application creation was to overcome inconsistency and diversity of students' product functional models done using paper-pen approach. The inconsistency and diversity were evident regardless of whether students were using suggested vocabulary (Hirtz) or had no imposed restrictions on functions and flows naming. The application was created using Java and JavaFX (for GUI creation) with Apache Derby as an embedded database. The Apache Derby was chosen because it lends itself to be used either as a server-based database or as the embedded database. For our prototype, we used an embedded database, but we are considering the usage of centralized database that can be extended or updated from one place and that all changes are immediately available to all client users. One of the problems that we encountered was the graphic diagram representation and manipulation for prototype application. Because the students and the teachers must embrace developed application before we create a final product, we did not want to develop graphic diagram creation and manipulation algorithms from scratch. So, we searched for an existing solution that was free and promising. We found one suitable solution on the GitHub [30].

The database schemas (Fig. 2) were created, and the tables were populated using vocabulary from Hirtz et al. [18]. The existing data model (from the graph-editor application) was expanded with new classes that resemble objects from the database.

The visualization files were stored in separate folders. We considered saving them as a BLOB (Binary Large Object), but we decided against it because this will considerably enlarge the database and will slow down the application.



```
CREATE TABLE DEFINITIONS
(
   ID INTEGER NOT NULL GENERATED ALWAYS
      AS IDENTITY (START WITH 1, INCREMENT BY 1),
   DEF CHAR(20),
   SUBLEVEL INTEGER NOT NULL,
   PARENT CHAR (40),
   PNAME CHAR(40),
   DESCR VARCHAR(1024),
   VISNAME CHAR(40),
   VISPATH VARCHAR(100),
   VISSOURCE CHAR(30),
   VISURL VARCHAR(150)
);
CREATE TABLE VISUALIZATION
(
   IDVIS INTEGER NOT NULL GENERATED ALWAYS
         AS IDENTITY (START WITH 1, INCREMENT BY 1),
   PNAME CHAR(40),
   VISNAME CHAR(40),
   VISPATH VARCHAR(100),
   VISSOURCE CHAR(30),
   VISURL VARCHAR(150)
);
```

Figure 2 Database schema

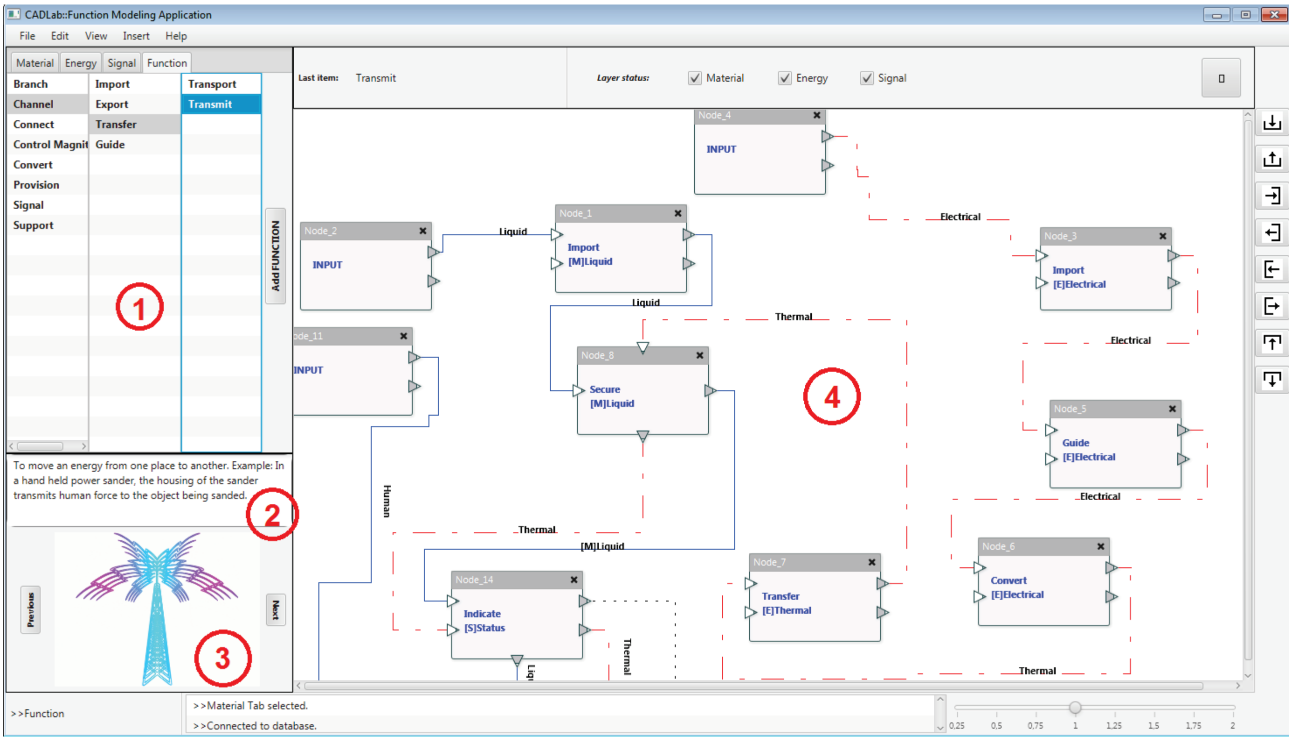
The visualization can be considered as an addition or extension of functional basis vocabulary. Functional basis vocabulary contains the explanation of a particular term, but students found them not very clear and sometimes incomprehensible and misleading. That is why we considered adding a visual aid (images and movies) beside the textual description. In the next application development iteration, we are considering storing visualization along with the functional model. Thus, a person reading an existing functional model could gain additional insight into the creator's idea.

The main application window is shown in Fig. 3. The main window is divided into the following areas:

1. Functional basis vocabulary using four primary tabs Material, Energy, Signal and Function. For every primary tab, primary, secondary and tertiary terms can be shown. Terms are shown in succession order. On application execution or particular tab selection, primary terms are shown, then after selection of primary term, the secondary is shown and so on.
2. Textual explanation of the selected term.
3. Visualization aid for the selected term. There could be more than one image or movie to visualize the selected term, and the user can iterate through them using the Previous and Next button.
4. Functional model graph. This area can be magnified or diminished according to the user needs.

3.1 The Application Usage Scenarios

Although the application can be used in various ways, in this article, one possible application usage scenario will be presented. First, the user must choose the function to be added (using the button Add FUNCTION see Fig. 3).



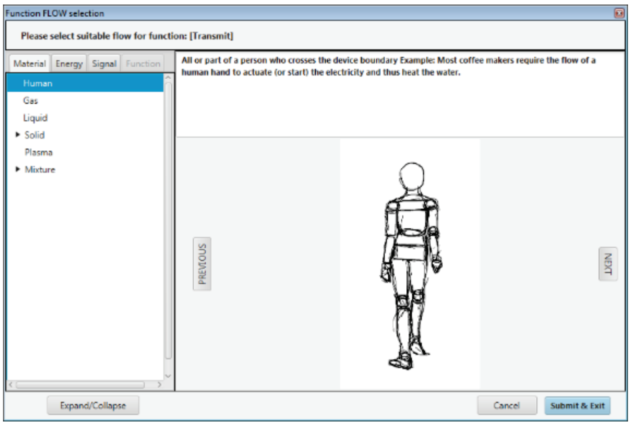
To move an energy from one place to another. Example: In a hand held power sander, the housing of the sander transmits human force to the object being sanded.

>>Function

>>Material Tab selected.
>>Connected to database.

Figure 3 The main application window

While adding a function, a new dialog will open, enabling the user to attach the flow logically associated with the function (Fig. 4).



Please select suitable flow for function: [Transmit]

All or part of a person who crosses the device boundary Example: Most coffee makers require the flow of a human hand to actuate (or start) the electricity and thus heat the water.

Figure 4 Additional dialog box enabling function flow selection

Result of this operation is a node with inscribed function and flow name (Fig. 5). The inserted node also has four connectors, two input and two output, which are automatically added.

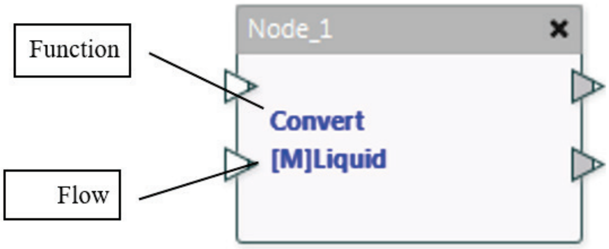


Figure 5 Created node

The connectors can be removed or added as user requires. This can be achieved by selecting Add Connector button on the pop-up dialog (Fig. 6a) showing on right mouse button click on the node (Fig. 6b). User can, also, use quick add connector buttons placed on the right of the main window. Except for the connector manipulation, this pop-up dialog (Fig. 6a) can be used to initiate function or flow change.

The function and the flow associated to the function can be changed at any time. When change is initiated, it will display the dialog depicted in Fig. 4.

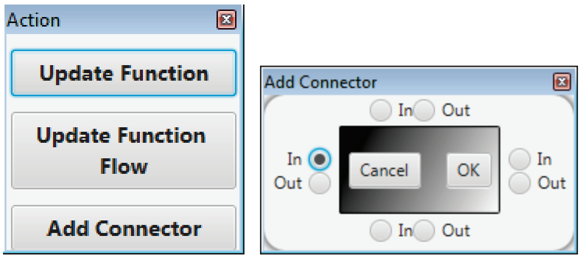


Figure 6 a) pop-up dialog; b) add connector dialog

The application usage showed that an arbitrary selection of functions and flow is not a good idea. Thus, for further application development, we are considering implementing rules for function and flow selection. The process of function adding can be repeated, or the user can add a flow between the nodes. The flow can be selected from the list boxes on the left-hand side after the flow type (tab) is selected. Then the user must click on one of the connectors on one of the nodes and drag the mouse to the connector on another node (Fig. 7).

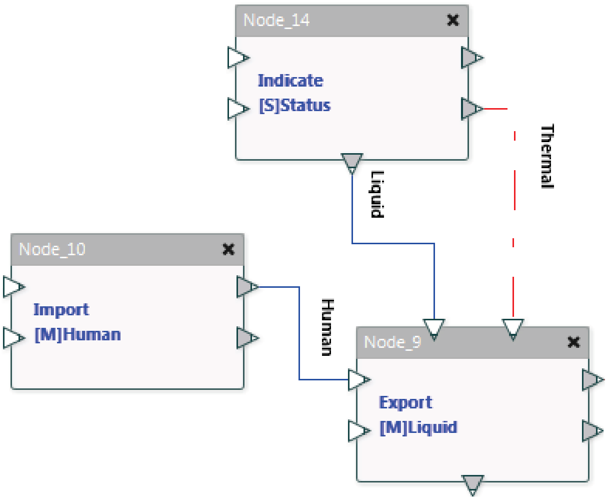


Figure 7 Connected nodes

The described steps can be repeated until the desired functional model is created. Created functional models can be saved and later opened as necessary. Files are saved in XML format as plain text.

The nodes and the connections can be freely dragged on the canvas. As an aid user can activate the grid and/or snap-to-grid option to better position the nodes and to create cleaner graphs. Also, the user can change magnification level from 0.25 to 2. For easier view manipulation a bird's view is available and can be activated by button from the main window upper-right corner (Fig. 8).

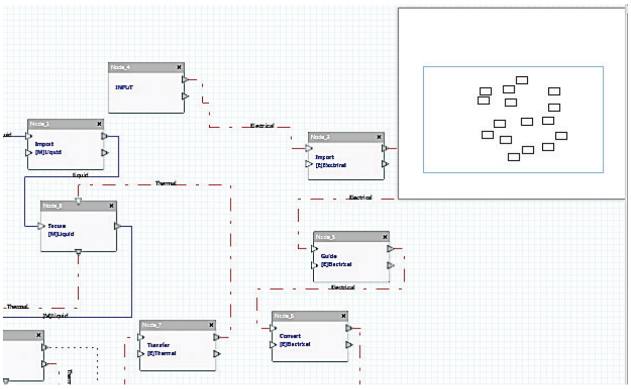


Figure 8 Bird's view example

4 THE EXPERIMENT ARCHITECTURE

The intended application usage is for all mechanical engineering students and especially for students in a product development course. Later, after the application reaches a more mature stage, the application will be presented to the young and senior engineers in the firms.

To be able to validate the created application, we conducted the experiment. What we were trying to achieve with the experiment is to get insight in student application usage. Basically, we were interested in the acceptance of the application by the students. This is because they will be the ones that will use the application through their studies and hopefully afterwards in their professional life, and we wanted to be sure that the application is tailored to their needs.

The teacher moderated the experiment, and the participants were 20 students of mechanical design (3rd to 5th year). The students had previously created product functional models and are familiar with the product development theories and functional basis. The experiment protocol was as follows:

- Introducing the basic idea to the students, explaining what are their tasks, and how to complete them.
- Creation of the given product functional model using paper-pen approach and functional basis.
- Creation of the same product functional model using the application.
- Completing the online questionnaire.
- Completing the interview with the moderator.
- Also, students are given the following guidelines:
- Function and flow names must be written.
- Flows must be distinguished from one another (use different colours or line font).
- System boundaries must be visible.
- All input and output flows must be entered.
- Flows should be checked for consistency.
- The student is randomly assigned one of the following products to work with: hairdryer, toaster, kitchen mixer, water heater, smoke detector and desktop ventilator. In Fig. 9a few paper-pen functional models are depicted.

Depicted paper-pen student's assignments speak for themselves. Before application creation was considered, teachers discussed with students several problems regarding building functional models' diagrams.

The major highlighted problem is that teacher faces many difficulties to understand and evaluate student's functional models' diagrams.

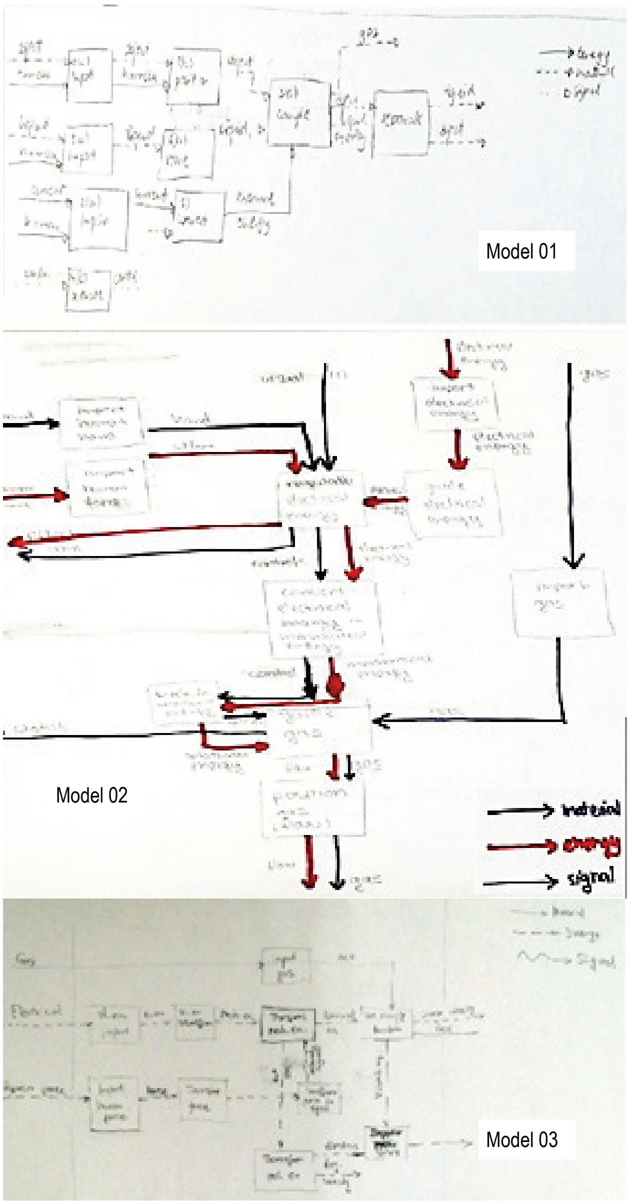


Figure 9 Example of student's paper-pen functional models

In Fig. 10 a few products functional models created using the application are depicted. The completed functional models were reviewed and compared to the paper-pen models by teachers. Teachers commented that functional models created using application were more readable, and the students did fewer mistakes on chaining functions and flows. One very important comment was that teachers were able to compare student's work done using program application more easily than paper-pen models. This analysis and more experiments were left for future work because the application is not yet "approved" for usage in courses.

After students completed their assignments, they had to complete the online questionnaire. The online questionnaire was created using Google Forms [31].

Students had to answer the following questions:

- Year of study - so we could assess the level of their previous exposure to the creation of the product functional models.
- Specialized in (mechanical design, medical engineering, IC engines and motor vehicles, mechatronics and robotics) - not all students are attending product development course, so the students that are not familiar with functional modelling are given a brief introduction.

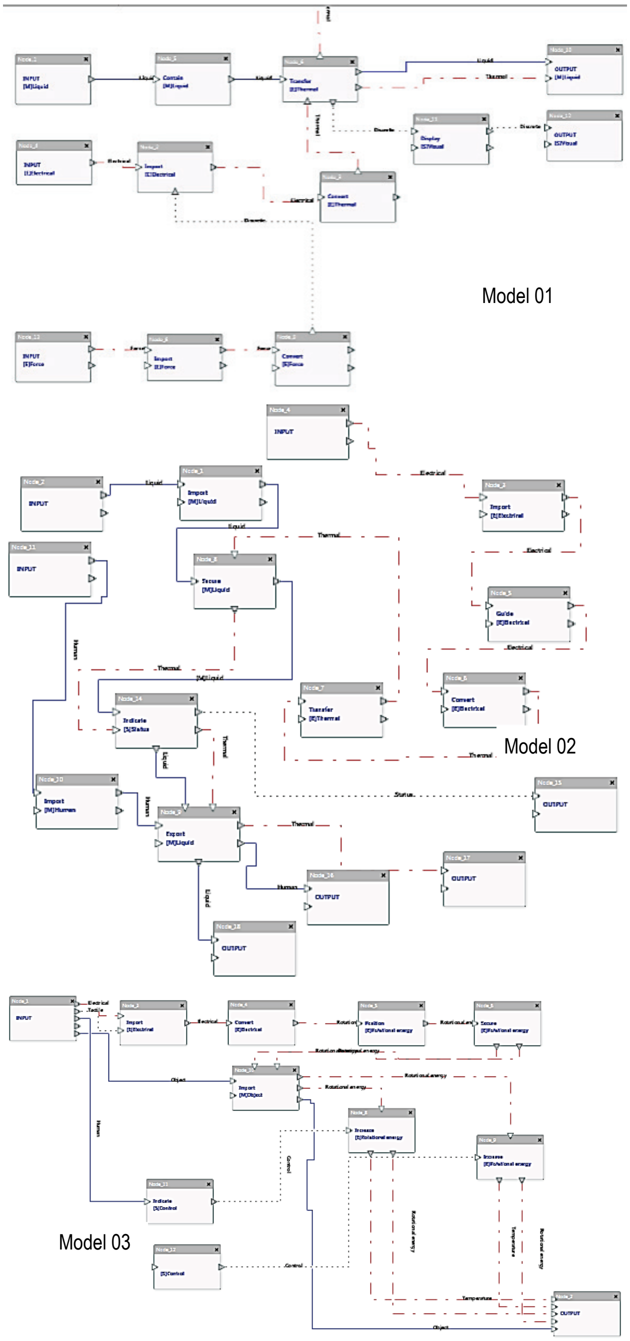


Figure 10 Student's functional models created using the application

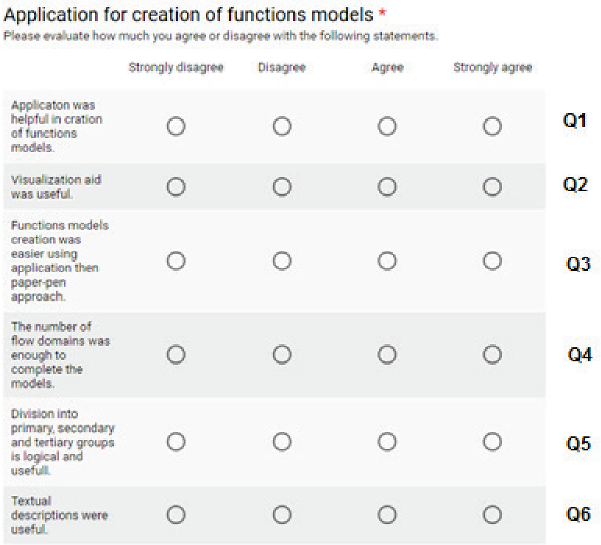
Group of questions regarding the application (Fig. 11)

- *Q*1 - this question intention is to get students impression on the helpfulness of the applications.
- *Q*2 - targets the visualization aid (images and movies), so we can decide whether this functionality is unnecessary or student benefits from its usage.
- *Q*3 - so far students have only used paper-pen approach, and we wanted to know if they prefer paper-pen over application.
- *Q*4 - from this question, we wanted to get information whether the existing number of domains is enough for models' creation or we need to consider implementing a mechanism for adding new domains.

- *Q*5 - this question targets functional basis, and its division into three levels.
- *Q*6 - functional basis comes with the textual description attached to each term, and we wanted to know if these descriptions are useful or not.

Also, the moderator conducted a short interview with each student asking the following questions:

- What additional functionalities, in your opinion, must be implemented in the application? - So we can tailor the application more to student's needs.
- Did you have any problems during your work? - We were interested in bugs and abnormalities in the application execution so they can be fixed.
- Did you find some functions or flows missing? - This question was related to the *Q*4 so we could refine student's answers.
- Overall what is your impression of functional basis vocabulary? - The functional basis was selected by the faculty staff, and we wanted to know if student support our selection or they find another functional vocabulary more adequate.



Application for creation of functions models *

Please evaluate how much you agree or disagree with the following statements.

	Strongly disagree	Disagree	Agree	Strongly agree	
Applicaton was helpful in cration of functions models.	○	○	○	○	Q1
Visualization aid was useful.	○	○	○	○	Q2
Functions models creation was easier using application then paper-pen approach.	○	○	○	○	Q3
The number of flow domains was enough to complete the models.	○	○	○	○	Q4
Division into primary, secondary and tertiary groups is logical and usefull.	○	○	○	○	Q5
Textual descriptions were useful.	○	○	○	○	Q6

Figure 11 The application related questions

5 RESULTS AND DISCUSSION

After the experiment was done, and all the data were collected, we started the analysis. The analysis was done according to the questionnaire and the conducted interviews. Majority of the students were from the final year of study and were from the mechanical design department.

In Fig. 12 the answers about the application are shown. Unfortunately, not all the students gave answers to all the questions, but most of them did.

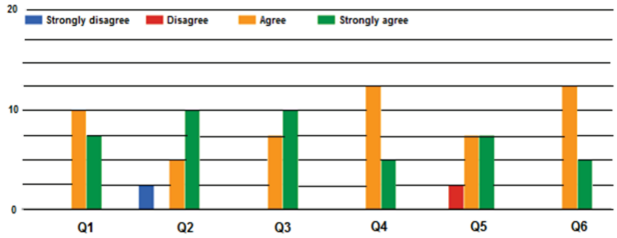


Figure 12 Answers about the application

Among all the student's answers, the most interesting for application evolution and upgrading are the following answers from interviews.

Additional functionalities:

- To be able to create regions - regions are common in the creation of functional models, and they allow logical grouping of functions; in the current application configuration regions creation is not directly enabled, but the user can create an empty node and extend its boundaries to simulate a region. We agreed that this capability must be implemented in the application.
- Increased quality of flow display - selected graphical library has several shortcomings that have to be addressed in the next iteration.
- Shortcuts to add a function - this was observation from one student, and his argument was that he had to "click" several times to be able to select ADD FUNCTION button.
- Function and function's descriptions search - five students pointed out that search capability both by the name and by the description would increase application functionality.
- More visual examples for functions, ability to search the function using wildcard characters - here a student found the number of images and movies provided by the function is too small. Application architecture enables the user to supply as many as needed images and movies for each term from the vocabulary.
- Automatic model checking and alerting in case of inconsistent flow and function combination - this requirement is under development, as a separate library, and eventually will be implemented in the finished application release.
- Automatic tips that contain last and the most used vocabulary terms - seven students requested this functionality. We are considering implementing this.
- To be able to control the colour and shape of the flow lines - eleven students requested the ability to control the colour and shape of the flow lines. Because of the limitations of the underlying graphical library this functionality cannot be implemented but if the decision is made to change the graphical library, then it will be considered.
- Zoom using mouse scroll button - ten students requested this functionality and it will be implemented in the next release.
- One button to add input and output functions without the drop-down menu - two students found current implementation where the addition of input and output functions positioned under a drop-down menu is too complicated and their suggestion was to add button instead. Currently we are not considering changing this functionality.
- There should be a way to add prefix or suffix to the function or flow so that something like "Convert liquid to steam" could be inserted - almost all students required this functionality. After a discussion with the teachers, there were two sides, one for adding this functionality, and the other for not adding it. The arguments for adding were that this functionality will increase the readability of functional models and the argument against was that this functionality will

decrease the readability. So, the decision was made that, for now, the functionality will not be implemented, and that further discussions and research should be conducted.

Problems:

- Flow lines are not updated - fifteen students reported this problem. The problem is with graphical library shortcomings.
- It was hard for me to find a function for measurement, if it exists - because of the nature of the product for which students crated functional models a couple of functions were needed, but they did not exist in the functional basis. That is why we are considering enabling the addition of new functions and flows, but we are still arguing about the best way for implementation.
- Smaller spaces between dots in flow lines - eleven students reported this problem. The problem is with graphical library shortcomings.
- When the application works for a long time, and the graph has many functions, the application slows down - ten students reported a problem with the application degrading speed. This problem was with application memory management and it was corrected.
- "For me, the English language was a problem because I had a tough time understanding the meaning of a function or flow; there should be the ability to change the language" - eleven students reported this as a problem. The ability to select the working language will be implemented in the next application release.

6 CONCLUSION

In this article, the authors described the application for creation of functional models based on the functional basis. Also, the decisions and the problems encountered during development were described and explained. The primary issue that needed to be resolved with application development was a problem with paper-pen functional models created by the students. Particularly paper-pen models readability and understanding. After several discussions among teachers, we decided to give an answer to two research questions "does the restricted usage of terms, imposed by the application, improve the readability of models?" and "do the restricted terms of usage, imposed by the application, give more readable models?"

To be able to get the answers, we conducted a moderated experiment. After the experiment finished, we interviewed participants, and they had to fill the questionnaire. The results are described and discussed in the fifth section. After the analysis of the results, we were able to get the answers to the questions. Almost all students (one student preferred paper-pen approach) found the application helpful and easy to use. Eleven students were struggling with the language, but they argued that visualization aid was useful to better understand and select the right term from the vocabulary. Also, most students were in favour of visualization aid implementation adding that the used visualization aid had to be a part of the functional model file. The second question, "does restricted terms usage, imposed by the application, give more readable models?" was more related to the teachers. The teachers that were involved in the experiment (reviewing the students' models) argued that functional models created using application were more readable and that the students made fewer mistakes. One significant comment was that teachers were able to compare student's work more easily than when comparing paper-pen models.

After all the conducted analysis and discussions, we concluded that the developed application was suitable for usage in our courses after the found issues were addressed. Also, the next steps in application development were defined. First, we are in the process of creation of a new graphics library that will replace existing, second, the functional model consistency checking ability will be implemented. Also, we are considering implementing the ability to save the model in different formats, for example SysML or UML.

7 REFERENCES

[1] Patel A., Kramer, W., Summers, J. D., & Shuffler-Porter, M. (2016). Function Modeling: A Study of Sequential Model Completion Based on Count and Chaining of Functions. *Proceedings of IDETC16*, 1-11. https://doi.org/10.1115/DETC2016-59860

[2] Pahl, G. & Beitz, W. (1988). *Engineering Design: A Systematic Approach*, Springer-Verlag, Berlin.

[3] Keuneke, A. M. (1991). Device representation: The significance of functional knowledge. *IEEE Expert, 6*(2), 22-25. https://doi.org/10.1109/64.79705

[4] Welch, R. V. & Dixon, J. R. (1992). Representing function, behavior and structure during conceptual design. *Design Theory and Methodology -DTM'92-,* 11-18.

[5] Iwasaki, Y., Fikes, R., Vescovi, M., & Chandrasekaran, B. (1993). How things are intended to work: Capturing functional knowledge in device design. *Proceedings of IJCAI'93*, 1516-522.

[6] Stone, R. B. & Wood, K. L. (2000). Development of a functional basis for design. *Journal of Mechanical Design 122*, 359-370. https://doi.org/10.1115/1.1289637

[7] Pahl, G., Beitz, W., Feldhusen, J., & Grote K.H. (2007). *Engineering Design: A Systematic Approach*, 3rd ed. London: Springer. https://doi.org/10.1007/978-1-84628-319-2

[8] Eisenbart, B., Gericke, K., & Blessing, L. (2013). An analysis of functional modeling approaches across disciplines. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 27*, 281-289. https://doi.org/10.1017/S0890060413000280

[9] Stone, R. B. & Wood, K. L. (2000). Development of a functional basis for design. *Journal of Mechanical Design 122*, 359-370. https://doi.org/10.1115/1.1289637

[10] Ulrich, K. T. & Eppinger, S. D. (2003). *Product Design and Development*, 3rd ed. New York: McGraw-Hill/Irwin.

[11] Ullman, D. G. (2000). *The Mechanical Design Process*, 4th ed. New York: McGraw-Hill.

[12] Pahl, G., Beitz, W., Feldhusen, J., & Grote, K. H. (2007). *Engineering Design: A Systematic Approach*, 3rd ed. London: Springer. https://doi.org/10.1007/978-1-84628-319-2

[13] Vermaas, P. (2011). Accepting Ambiguity of Engineering Functional Descriptions. *Proceedings of 18th International Conference on Engineering Design, ICED*, 98-107.

[14] Eppinger, S. D. & Browning, T. R. (2012). *Design Structure Matrix Methods and Applications*, Cambridge, MA: MIT Press. https://doi.org/10.7551/mitpress/8896.001.0001

[15] Hubka, V. & Eder, W. E. (1988). *Theory of technical systems*. https://doi.org/10.1007/978-3-642-52121-8

[16] Ulrich K. T. & Eppinger, S. D. (1995). *Product design and development*, McGraw-Hill.

[17] Kurfman, M. A., Stone, R. B., Wie, M. Van, Wood, K. L., & Otto, K. N. (2000). Theoretical Underpinnings of Functional

Modeling: Preliminary Experimental Studies. *Proceedings of DETC2000, ASME*.

[18] Hirtz, J., Stone, R. B., & McAdams, D. (2002). A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts. *Research in Engineering Design, 13*, 65-82. https://doi.org/10.1007/s00163-001-0008-3

[19] Maier, J. R. A. & Fadel, G. M. (2009). Affordance based design methods for innovative design, redesign and reverse engineering. *Research in Engineering Design, 20*(4), 225-239. https://doi.org/10.1007/s00163-009-0064-7

[20] Gaffney, E. S., Maier, J. R., & Fadel, G. M. (2007). Roles of Function and Affordance in the Evolution of Artifacts. *ICED 2007*, 1-12.

[21] Hirtz, J., Stone, R. B., McAdams, D., Szykman, S., & Wook, L. K. (2001). Evolving a Functional Basis for Engineering Design. *Proceedings of DETC01 2001 ASME Design Engineering Technical Conferences*.

[22] Erden, M. S., Komoto, H., Van Beek, T. J., D'Amelio, V., Echavarria, E., & Tomiyama, T. (2008). A review of function modeling: Approaches and applications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 22*, 147-169. https://doi.org/10.1017/S0890060408000103

[23] van Beek, T. J. & Tomiyama, T. (2009). Integrating conventional system views with function-behaviour-state modelling. *Proceedings of CIRP Design Conference*.

[24] Hundal, M. S. (1990). A Systematic Method for Developing Function Structures, Solutions and Concept Variants. *Mech. Math. Theory, 25*(3), 243-256. https://doi.org/10.1016/0094-114X(90)90027-H

[25] Nagel, R. L., Hutcheson, R. S., Stone, R., McAdams, A. D., & Donndelinger, J. (2008). Function Design Framework (FDF): Integrated Process and Function Modeling for Complex System Design. *Proceedings of ASME International Design Engineering Technical Conference, DETC2008-49369*. https://doi.org/10.1115/DETC2008-49369

[26] Kirchman, C. F. & Fadel G. M. (1998). Classifying Functions for Mechanical Design. *Transactions of the ASME, 482/120*.

[27] Nagel, R. L., Perry, L. K., Stone, B. R., & McAdams, A. D. (2009). FunctionCAD: A Functional Modeling Application Based on the Function Design Framework. *Proceedings of ASME International Design Engineering Technical Conference & Computers and Information in Engineering Conference, IDETC/CIE 2009*. https://doi.org/10.1115/DETC2009-87010

[28] Karakašić, M., Zadnik, Ž., Kljajin, M., & Duhovnik, J. (2010). Functional Structure Generation within Multi-Structured Matrix Forms. *Technical Gazette, 17*(4), 465-473.

[29] Zadnik, Ž., Cok, V., Karakašić, M., Kljajin, M., & Duhovnik, J. (2011). Modularity Solutions within a Matrix of Function and Functionality (MFF). *Technical Gazette 18*(4), 471-478.

[30] Retrieved from: https://github.com/tesis-dynaware/graph-editor, accessed on 2018-03-01.

[31] Retrieved from: https://docs.google.com/forms/u/0/, accessed on 2019-05-06.

Contact information:

Nenad BOJCETIC
(Corresponding author)
Faculty of Mechanical Engineering and Naval Architecture,
Ivana Lucica 5, 10000 Zagreb, Croatia
E-mail: nbojcetic@fsb.hr

Filip VALJAK
Faculty of Mechanical Engineering and Naval Architecture,
Ivana Lucica 5, 10000 Zagreb, Croatia
E-mail: fvaljak@fsb.hr

Stjepan FLEGARIC
Faculty of Mechanical Engineering and Naval Architecture,
Ivana Lucica 5, 10000 Zagreb, Croatia
E-mail: sflegaric@fsb.hr

Mario STORGA
Faculty of Mechanical Engineering and Naval Architecture,
Ivana Lucica 5, 10000 Zagreb, Croatia
E-mail: mstorga@fsb.hr