LOGICAL OPERATOR USAGE IN STRUCTURAL MODELLING

Ieva Zeltmate [(a)]

[(a)] Riga Technical University, Faculty of Computer Science and Information Technology
Department of System Theory and Design

ieva.zeltmate@gmail.com

ABSTRACT
Structural modelling is an approach that is developed to support expert`s knowledge acquisition and representation in the process of complex system analysis, design and modelling. The framework of several integrated structural models is used to capture and manage knowledge about specific system morphology, operations, and behaviour under certain operating conditions. To minimize expert's workload, once acquired knowledge must be automatically transformed between related structural models. Methods for structural model transformations are used to keep important system characteristics and conectedness between system components and functionality. In this paper special cases of structural model transformations are described. In models additional elements called logical operators are used. The goal of logical operator usage is to depict in structural models possible combinations and operation compatibilities that exist between real system components. The formal method that allows transform and use logic operators between models is described and examples of structural model transformations are given.

Keywords: structural modelling, transformations, functional, behaviour, parameters

1. INTRODUCTION

System structural modelling is a type of modelling that can be used in system design and analysis, to acquire expert`s knowledge about system structure and organisation (Grundspenkis 1999; Kresken 1996; Oliva 2004). System structural modelling becomes important when it is necessary to understand and to change complex system and also predict its behaviour (Grundspenkis 1999). Usually, when system analysis is performed, a domain expert knows that system components and relationships among them are logically connected. However, when a system model is created, not all knowledge of the system structure may be presented. Sometimes it is because the model is not expressive enough to represent all necessary knowledge, but sometimes all knowledge about system is not known. Relationships between elements in the model show the connections between objects and logical operation sequences in the system in terms of normal functioning (Grundspenkis 1999; Oliva 2004). Although a system structural model rarely answers how the relationships are connected or how elements collaborate or exclude one another. To address these questions it is important to represent coherence between element`s relationships. Therefore additional elements, called logical operators, are used in Structural modelling approach (Grundspenkis 1997; Zeltmate and Grundspenkis 2008). Logical operators are extra characteristics (Capiluppi 2007) integrated in system structural models (Zeltmate and Grundspenkis 2008). Using logical operators allows to describe system's conformity to rules and order of activities. System representation with logical operators clearly defines existing paths, relationships, and flows in the models that describe real world systems.

Structural modelling (SM) is systematic and domain model–based approach that can be used in system structural modelling. The goal of SM is to support consecutive, structural system analysis, design and reasoning about systems using acquired knowledge. There are two different and integrated paradigms in structural modelling: a) morphological structure model (MSM) and b) functional structure model (FSM). MSM lacks characteristics which are necessary for diagnostic and predictive reasoning. Therefore functional representation and transformation algorithm to transform acquired knowledge from MSM into FSM were created. Functional structural model can be derived in the space of functions, behaviour, and parameters (Grundspenkis 1997; Grundspenkis 1999). Additional attributes and specific cases for model transformations are acquired using continuous SM approach. Representation of logic of the relationships in structural models offers a qualitative, more detailed knowledge about system and allows identifying relationship compatibilities.

In Structural modelling, the logic is defined in a morphological structural model and transformed to functional structural model. To minimize an expert's workload and time during knowledge representation process, it is essential to maintain and transform newly acquired logical operators automatically. Logical operator visualisation and interpretation in the MSM as well transformations from the MSM to the FMS in a space of functions were overviewed previously by (Zeltmate and Grundspenkis 2008). In this paper, I consider transformations from MSM to FMS in the space of behaviour and parameters and corresponding examples. The transformations in the space of behaviour are analogous to the FMS transformations in the space of functions.

The paper overviews specific structural model transformation cases in the space of behaviour and parameters where the logical operands are used. This paper has five sections which are organized as follows. In section two concepts of structural models are presented to introduce with formal aspects of structural modelling approach and used notations. In section three graph transformation concepts and transformation examples are given. The section four includes an example of a cooling system of an internal combustion engine representation. Three structural models and corresponding transformation cases are shown. In conclusions a short summary of paper is given and tasks for future work described.

2. THE CONCEPTS OF STRUCTURAL MODELS

In Structural modelling approach structural and functional aspects of a system are unified in one framework. Using several types of interconnected models allows to observe and represent a system from different viewpoints (Grundspenkis 1999; Abu-Hanna and Jansweier 1994; Oliva 2004). Structural models are visualized as directed graphs (Tutte 2001) where the system components are depicted as nodes and relationships between components as directed edges. Hence nodes represent structural components of models. Structural component can be any collection of elements that is acquired in the process of conceptual system decomposition. MSM represents the set of objects and relations between objects at the system decomposition level selected by experts. In the morphological structural model, three structural components are used to encapsulate the domain knowledge: objects, flows, and contacts (see Figure 1).

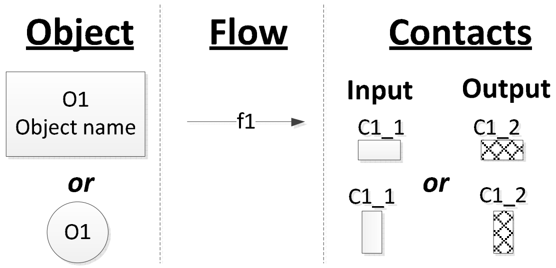

Figure 1: MSM components

Objects represent small and independent units of the domain knowledge. Although in the MSM objects are nodes, each object can be decomposed and inspected in detail. A flow is an interaction path between one object's output and another object's input (Grundspenkis 1997). Flows are used to represent functional and causal dependencies in system. Flows are identified by a small letter 'f' followed by a number. Each flow can perform one or more actions and execute one function. The knowledge about a flow includes the function's name in the appropriate frame. This enables the connection between structural models and automates model transformations. An object's contacts include the information about the Input/Output state, the name of the flow, the function, and behaviour states. Contact names are acquired in the knowledge acquisition process. A contact's context or meaning does not depend on its vertical or horizontal position. If interpreted in an application domain, abstract objects correspond to components of a given system, and contacts represent their inputs and outputs (Grundspenkis 1997).

Structure, functions and behaviour are interconnected in a system and influence each other. That is why it is necessary to create and view corresponding structural models as interconnected. In SM the transformation is formal method that relates system morphological and functional aspects. The morphological structure model can be transformed into a functional structural model in function, behaviour, and parameter space (Grundspenkis 1997; Grundspenkis 1999; Zeltmate and Grundspenkis 2008). This means that transformation provides iterative represented knowledge transfer from MSM to FSM. In order to perform transformation several steps and rules must be considered (Grundspenkis 1997)

The behaviour state and parameters affect the action and time, during the function execution, and describe the system dynamics. From the structural point of view, behaviour state specifies constraints or boundaries, within which the function is implemented under certain circumstances. The behaviour state is a qualitative characteristic of a flow and describes the way how flow acts while it maintains steady stream from one object to another. Flow executes some function and implements some process in a system. The parameters are variables with certain values that specify the function behaviour. The topology of the functional structural model in the behaviour space (FSM BS) is derived from the MSM automatically (Grundspenkis 1997). The basic structural components for the behaviour representation are: behaviour state, behaviour, and flow (see Figure 2).

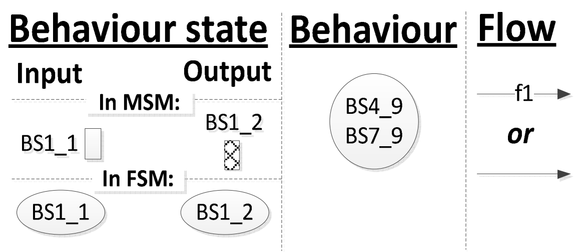

Figure 2: FSM BS components

Behaviour states (see Figure 3) can be represented in both structural models (MSM and FSM). They encapsulate the knowledge about the behaviour of a system that operates under "normal" conditions. When an object performs an action with a specific function, a determined behaviour is applied. The behaviour expresses itself in the action, when the function is executed. It can be set at the output of an object in the model and assessed at the input of another object. Comparing the behaviour states at the output and input, it is possible to diagnose system functioning faults (Grundspenkis 1997; Capiluppi 2007). Accordingly, behaviour states for each system component are identified and described in the component's (object's) contacts. The representation of the behaviour states in the MSM is similar to the representation of contacts. Each contact has more than one possible behaviour states, but it can be only in one state at a given moment. That state can be recorded only in one structural model.

In FSM behaviour states are visually represented using ellipses. Behaviour states specify the flow at in inputs and outputs of a given object. If a system functions correctly, the behaviour at the beginning of flow (output) is the same as at the end of flow (input). A formal model transformation behaviour states in FSM BS into pairs. The transformed states define the behaviour of the function (see Figure 3). First, the input behaviour state is represented, followed by the input behaviour state. The behaviour includes behaviour state set that describes the flow representing the connection between two behaviour states.

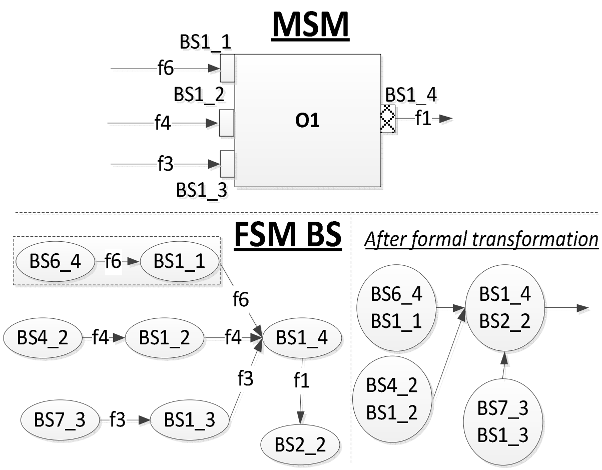

Figure 3: Behaviour representation in MSM and FSM

Each behaviour state has a parameter set. A parameter has a value or a set of values that characterize the state of the behaviour. A parameter can be linked to other parameters. When the system is described, external and internal relationships between the parameters are identified. Identifying parameters and relationships between them is the main task of building an explicit functional structure model in the parameters space (FSM PS). The FSM PS is used in the structural modelling approach to represent system dynamics and enables diagnostic reasoning (Grundspenkis 1997). Logical operators for external relationships are acquired automatically from the MSM, but for the internal relationships the logic must be acquired and specified by an expert. Three components are used to represent parameters in the FSM PS when the system functions correctly: parameter (P1_1), parameter set (PS1_1), and flow (f1; see Figure 4). When the system functions with faults (Grundspenkis 1997, Capiluppi 2007), an additional element–defect–is used. Defect is a cause which can create fault in object behaviour and as a consequence a failure of system functioning.

In a space of parameters (Grundspenkis 1997) the decomposition of the parameter set is considered. First step in the FSM PS development is to acquire a set of parameters from the MSM (or FSM BS), which is done automatically. At the next step, each set is decomposed. An expert represents the knowledge about the parameters and relations between them in a frame hierarchy. In the FSM PS diagrams, the parameters are identified by a letter ‘P’, followed by symbols from the parameter set (for example, 6_4) that correspond to the object number and the parameter set, followed by an additional number for each parameter.

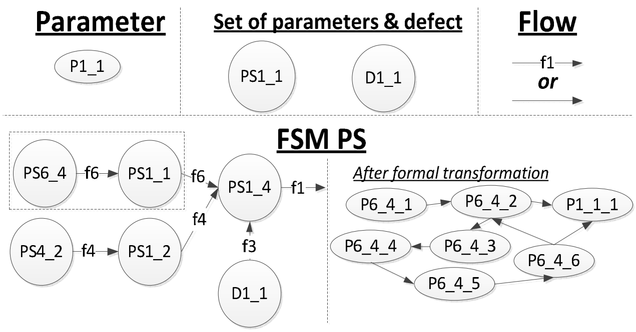

Figure 4: Parameters and corresponding elements

At least one member of the parameter set has a relationship with external parameters. An external parameter is a parameter, which can be found in another behaviour state for the same or another object. Accordingly, an internal parameter is a parameter, which is a part of the object's behaviour state. When the parameter space is decomposed, we first describe the parameters, which are connected with an external parameter in the output contact behaviour state (OBS). The parameters, which are connected to an external parameter in the input contact behaviour state (IBS), are described last.

If system functions faulty then acquired structural model differs from the structural models for a normally functioning system. The FSM PS must be extended using element defect in the model representation. If the defect which have caused faults does not seriously affect the system's functioning (that is, does not cause system failures), then only the FSM PS differs. More serious functioning problems (say, a lost system component or relationship) also appear in other structural models. The defects are usually caused by misbehaving objects located in higher or lower decomposition levels. These objects can be the system’s components or external objects. A defect is identified similarly to the parameter set, because it can be caused by the object parameters that logically appear in the OBS. A defect is identified by a letter ‘D’, followed by the number of the object that is affected by this defect, followed by the symbol '_' and the number of defect. Defects can be detected by measuring the parameter values. If the values changed and are outside the defined and allowed ranges then the system functions differently. Unexpected changes in the structure models can be observed in two situations: 1) when an expert has not represented all the knowledge about the system or the system has an alternative functioning case and 2) when the system operates with faults. If the sources of the fault are known, the expert can represent about the timing and origins of it in the frame hierarchy. This knowledge can be used in the case of the system malfunctioning to diagnose the cause of fault. Using the FSM PS an event tree can be created to find the defects which caused faults. Change of parameter value is called an event. A causal order of two events is interpreted as a causal relationship. Causal relationship chain allows to generate event tree structure (Grundspenkis 1997; Grundspenkis 1999). Event tree represents cause and consequence relationships. Since in system exists organization, change in one element impact other related elements. Organization is defined as relationship between two system elements A and B that can be a condition and can impact element C properties, characteristics, values or state (Ashby 2004). Because not all parameters are observed and measured in real systems (especially complex ones), event trees play an important role in diagnostics. Diagnosis allows to detect causes why system behaviour has changed. Logical operators are essential elements in event trees. Transformations are performed to acquire functional representation of system and transfer logic between structural models. Once acquired and represented logic can be transferred also to event trees. Next, I describe possible concepts for the graph transformations.

3. GRAPH TRANSFORMATION CONCEPTS

The logic in graphs between arcs is represented using operators (AND; OR). Logical operators in models are used for visual interpretation of structural and causation equations. The notation allows to map flows and to determine causes and expected results for certain actions and changes. The logic and flows combined representations can be interpreted in the "if-then" rule format. The logical operators determine what function, what behaviour, and what parameters are active for the incoming or outgoing flows. Specific symbols represent logical operators in structural models (Zeltmate and Grundspenkis 2008). If there are not complex relationships between flows that require complex logical expressions/rules with parentheses, then the notation is quite simple (see Figure 5.a.–5c.).

A square on the flow or on the relationship/link between flows specifies a logical AND operator and a triangle specify an OR. Symbols ',' and ';' separate flow identifiers and draw attention to the operators AND and OR. Shapes and symbols on or above a flow refer to other flows affected by the current flow (see Fig 5.a and 5.b). For each case that maps a structural component's relationships, a rule can be designed. If this notation is implemented in a software tool that supports structural modelling approach then automatically acquired rules can be used to reason about the system. A shape on a relation between flows refers to the incoming flows, and symbols above the shape between the flow names refer to the outgoing flows (Figure 5.c). Figure 5.a corresponds to the case when the flow f1 affects the flow f2. If the current object has only one input and one output flow then there is no need for logical operators. In Figure 5.b, operator OR (a triangle) shows how the outgoing flows f2 and f3 are affected by the flow f1. Similarly, a square and a comma (Zeltmate and Grundspenkis 2008) correspond to the operator AND. In Figure 5.b the link connecting the flows f1 and f2 and the square shows that two flows operator affect two other flows. A line between flows with a vertical crossing dash and parentheses correspond to brackets in logical expressions like it is shown on Figure 5.d.

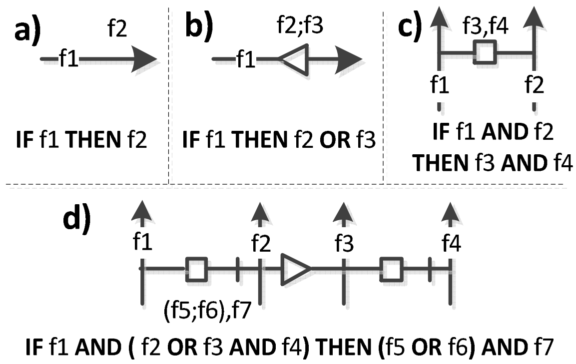

Figure 5: Logical operator repesentation

The described notation represents logical links between objects in structural models, their relationships, functions, and behaviour. In the case of normal functioning, four basic combinations of input and output flows exist (Zeltmate and Grundspenkis 2008): one to one; one to many; many to one; many to many. There are also situations when only output or only input flows for the object are detected, e.g., when there is a transient object in the system that accumulates or produces something (energy, matter, information). If an object does not consume resources and there are no incoming flows then the corresponding outgoing flows drain after some time. An example of such object is an electric battery. In a malfunctioning system, an object with only incoming or only outgoing flows indicates the presence of a fault, such as the disappearance of or change in some relationships (and perhaps objects). These cases and the corresponding transformations must be also described to diagnose system defects. Transformations from MSM to the functional structural models are similar in the space of behaviour and parameters and in the space of functions. Flows and logic are common components of all structural models. Thus, the knowledge about these components can be shared between models. Structural model transformations (Grundspenkis 1997) enable knowledge reuse in different contexts. Further in this section several examples are given. The chosen representation form shows possible input and output flow combinations, using logical operators.

The first combination case is simple and no specific notations are used to understand the logical causation relationship (see Figure 6). However the logical operator transfer in FSM BS and FSM PS differs from operator transfer in FSM FS which have been presented previously (Zeltmate and Grundspenkis 2008). The logical expressions for the case (a) are as follows:

1. **IF BS3_2 THEN BS1_1; IF BS1_2 THEN BS2_1**. An output behaviour state at output (for example, BS1_2) corresponds to exactly one input behaviour state (for example, BS2_1). Since these behaviour states are connected with one flow, they are usually equal, and the parameters that specify the behaviour can be equal, too. If the parameters are not equal, their values are at least in a predefined range.
2. **IF (BS3_2, BS1_1) THEN (BS1_2, BS2_1)**. If the flow f1 implements the behaviour of object O3 then the flow affects the behaviour of object O1.
3. **IF PS1_2 THEN PS2_1.** If one parameter set is used then it is connected to another parameter set to provide the required functionality. (Both sets belong to normally functioning objects) If two parameter sets belong to one flow then the names of the parameters in the sets are identical and the connection between the sets is strong: even though the two parameter sets belong to two objects, they can differ only in the parameters' values and in automatically assigned id like P1_2_1.

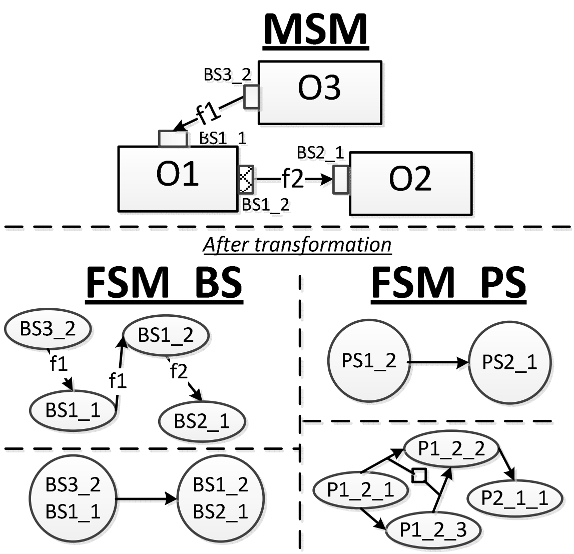

Figure 6: MSM and FSM BS, FSM PS, case (a)

If two parameter sets belong to two flows, the connection between them is weaker and the parameters in the sets can be different. I will not show the decomposition of FSM PS in other combination cases because logical operators for the parameter internal relationships are set by expert in the same manner as for the MSM.

The second combination covers three different cases (Zeltmate and Grundspenkis 2008). In the first two cases there is one input flow and several output flows connected with an AND (b) or an OR (c) d of logical operator. In the third case (d) there is one input flow and several output flows connected with logical AND and OR operators (See Figure 7). The logical expressions for the case (d) are listed below:

1. **IF (BS1_2, BS2_1) THEN ((BS2_2, BS4_1) OR (BS2_3, BS5_1)) AND (BS2_4, BS6_1).** The behaviour realized by the object O1 affects the object O2 behaviour making O2 behave in two different ways.
2. **IF PS1_2 THEN PS2_1 AND IF PS2_1 THEN (PS2_2 OR PS2_3) AND PS2_4.** When the system functions normally, there is an expression which is true for the system: IF PS1_2 THEN (PS4_1 OR PS5_1) AND PS6_1. The parameter set determined at the beginning of the flow f1 affects the parameter sets for the flows f2, f3s and f4.

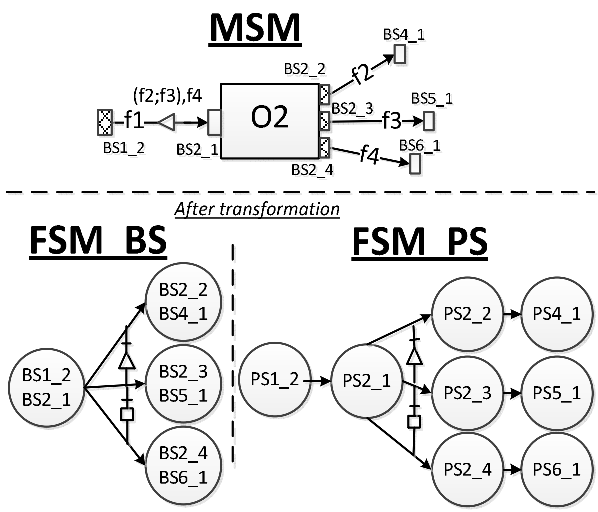

Figure 7: MSM and FSM BS, FSM PS, case (d)

The third combination can be divided into three different cases (Zeltmate and Grundspenkis 2008), too. These cases are common in situations when several input flows affect one output flow. In this paper the example of case (f) is represented (see Figure 8). The logical expressions for the case (f) are as follows:

1. **IF ((BS4_2, BS2_1) OR (BS5_2, BS2_2)) THEN (BS2_3, BS3_1).** Object O2 behaviour can be affected by the other two object (O4

and O5) behaviours. Because of the OR operator, only one of the objects (O4 or O5) affects O2.

2. **IF PS4_2 THEN PS2_1; IF PS5_2 THEN PS2_2; IF (PS2_1 OR PS2_2) THEN PS2_3.** When the system functions normally, then the following expression holds: IF (PS4_2 OR PS5_2) THEN PS3_1. The parameter sets at the beginning of the flows f2 or f3 can affect the parameter set for the flow f5.

In the case of a defect both logical operators (AND and OR) can be used between the flows. The OR operator means that the defect affects the functionality of the system only occasionally. The AND operator means that the defect always affects the functionality of the system.

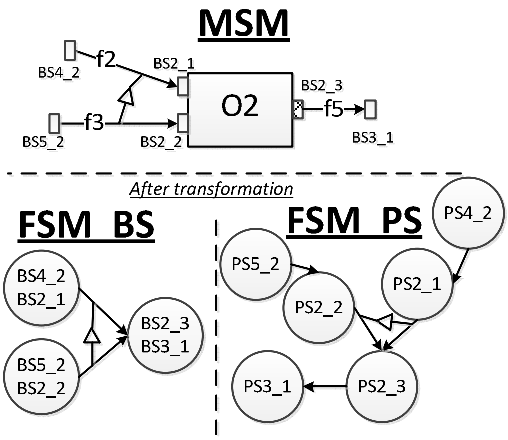

Figure 8: MSM and FSM BS, FSM PS, case (f)

The fourth combination of flows is the most general one. It covers the case of several input flows and several output flows, all possibly linked with various logical operators. This combination case has several sub–cases (Zeltmate and Grundspenkis 2008). I present one of them in Fig 9.

The following logical expressions hold for the case (m):

1. **IF (BS2_2, BS1_1) THEN (BS1_4, BS5_1) AND IF (BS2_2, BS1_1) AND (BS3_2, BS1_2) THEN (BS1_5, BS6_1) OR (BS1_6, BS7_1).** The behaviour implemented by the objects O2 and O3 affects the object O2 behaviour.
2. This complex case has several expressions:

- (IF PS2_2 THEN PS1_1) AND (IF PS3_2 THEN PS1_2) AND (IF PS1_1 AND PS1_2 THEN PS1_5 OR PS1_6) AND (IF PS1_5 THEN PS6_1) AND (IF PS1_6 THEN PS7_1)
- (IF PS2_2 THEN PS1_1) AND (IF PS1_1 THEN PS1_4) AND (IF PS1_4 THEN PS5_1)

3. Also two other expressions are true for the system in the case of normal functioning:

- (IF PS1_1 AND PS1_2 THEN PS6_1 OR PS7_1)
- (IF PS1_1 THEN PS5_1)

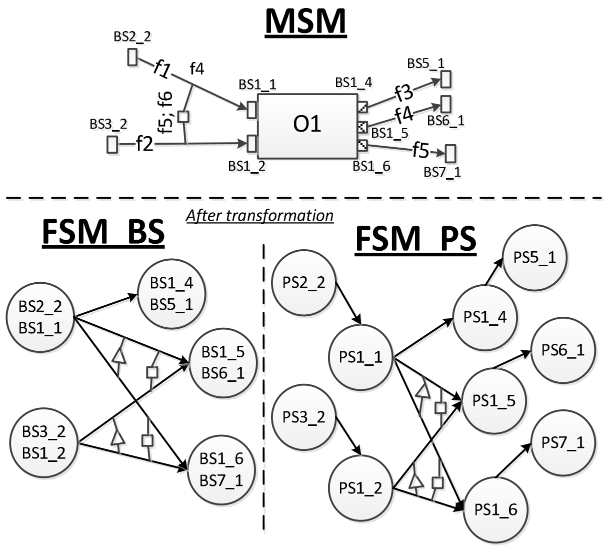

Figure 9: MSM and FSM BS, FSM PS, case (m)

Although structural models represent the instantaneous system structure, one can use them to analyse the transient system behaviour, reason about it, and diagnose problems in the case of the system malfunctioning. In the next section, I discuss an example of using logical operators in system modelling approach–a cooling system of an internal combustion engine (ICE) (Grundspenkis 1997; Zeltmate and Grundspenkis 2008).

4. EXAMPLE: ICE COOLING SYSTEM

The MSM of the ICE cooling system consists of 12 objects, 14 flows between them, and logical operators and links between the flows (Fig. 10). The MSM can be represented in two different ways: (a) as a list of contacts for each object or (b) as a set of behaviour states for each object. The difference between the representations is clear in a frame hierarchy, when the model is explored in detail: A contact includes all information about the flow and the behaviour state, but the behaviour state is just a part of the contact that includes a parameter set. To give a better idea about the structural model relations and possible transformations, Fig. 10 shows behaviour states.

No explicit logical operators between relationships in the model imply a logical AND operator. The cooling system's MSM has seven rules (with or without using the logical OR operator). Using OR changes the meaning of some rules (see Table 1) and actually describes the correct system behaviour.

Table 1: Rules in the morphological structure model

Rules (without OR usage):	Rules (with OR usage):
1.IF f1 AND f13 THEN f2	1. IF f1 AND f13 THEN f2 (case e)
2.IF f2 THEN f3 AND f4	*2. IF f2 THEN f3 OR f4 (case c)*
3.IF f3 THEN f5	3. IF f3 THEN f5 (case a)
4.IF f5 AND f8 THEN f6 AND f14	4. IF f5 AND f8 THEN f6 AND f14 (case h)
5.IF f6 THEN f7	5. IF f6 THEN f7 (case a)
6.IF f10 AND f11 AND f12 THEN f8	6. IF f10 AND f11 AND f12 THEN f8 (case e)
7.IF f9 AND f7 AND f4 THEN f1	7. IF f9 AND (f7 OR f4) THEN f1 (case g)

The MSM alone cannot be used to determine the influence of defects, because the defects can happen in the lower or higher level of the hierarchy. If a flow ends and there are no more incoming or outgoing flows from this object, then one of the following is true:

1. the object is connected to other objects beyond the system boundary and the environment is not shown in the MSM;
2. the system is not represented correctly; for example, an abstraction level shows objects or flows that belong to other decomposition levels;
3. there is a defect in the system caused by the expert's improper system understanding.

Determining all structural models and all rules for all cases in a system manually takes a lot of time and is costly. Using several structural models and a transformation methodology (Grundspenkis 2002) makes automated knowledge transfer possible (this includes the knowledge about the logical operators). Using logical operators in the FSM in the space of behaviour is similar to that in the space of functions (see Figure 11).

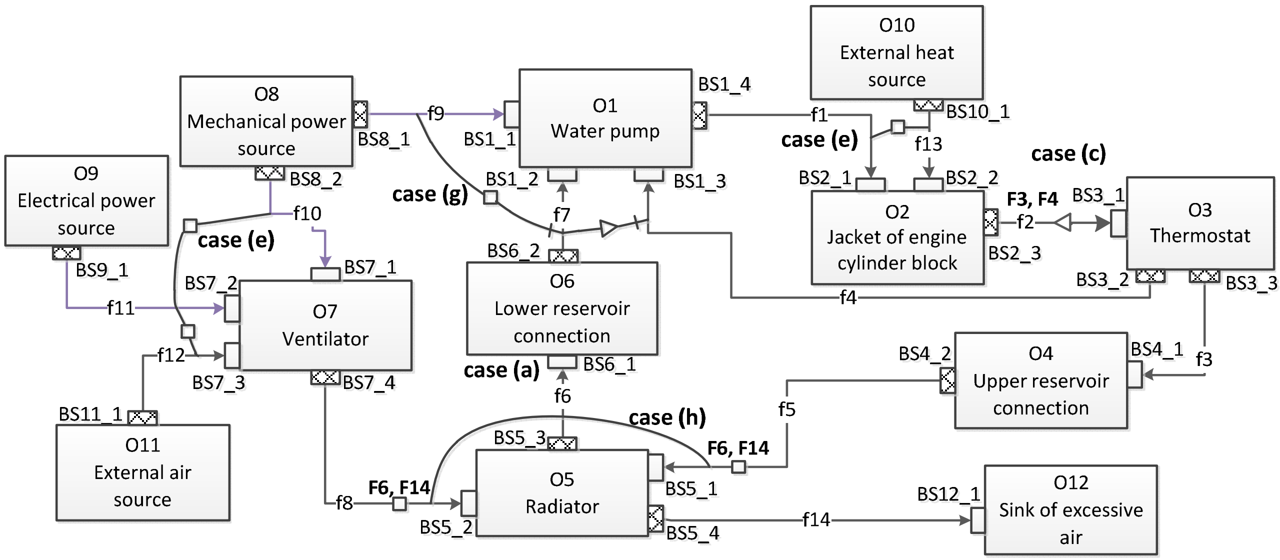

Figure 10: The morphological structure of an ICE cooling system

Each node in the diagram represents the behaviour state of the outgoing flow (above) and the behaviour state of the incoming flow (below). In the case of the normally functioning system these two states are strongly connected: they are either equal or at least similar. The behaviour state order in a node determines the direction of the flow. One can notice that there is no explicit connection between nodes (BS8_1, BS1_1) and (BS8_2, BS7_1). However, there should be a connection, because all behaviour states of an object are somehow interconnected. It is implied that there is a logical AND operator between these nodes.

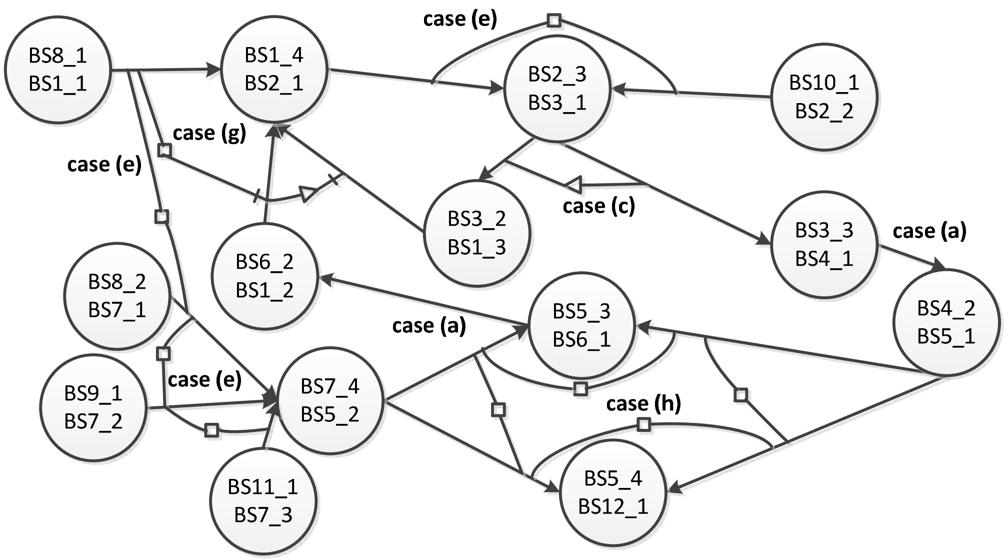

Figure 11: The functional structure model of an ICE cooling system in the space of behaviour

The next structural model that can be produced by the transformations is a functional structural model in the space of parameters. This structural model is similar to the FSM in the state of behaviour before grouping the behaviour states (see Figure 12).

Additional rules can be obtained in the parameter space. An example of an additional rule is: IF PS8_1 AND (PS6_2 OR PS3_2). An FSM PS represents causation relationships between parameter values that are connected with external parameters. Two parameter sets for each flow fully reflect the changes in the system in the case of a defect or rule changes. In the former case, there will be at least two flows: one caused by the defect and the other caused by a system object—usually connected with a logical AND operator. It is possible for some flows to change direction because of a defect. This change of direction can be detected through the change in parameter sets if logical operators are used: if the current values of the parameters differ from the original values, then one can create an event tree (Grundspenkis 1997) and determine the cause of the problem. If no logical operators are used, it is not possible to obtain correct rules that describe the system's functions, and build the causation chain. The functional structure model in the space of parameters makes it possible to speculate about the consequences of various parameter changes, such as made on purpose or in the case of faults.

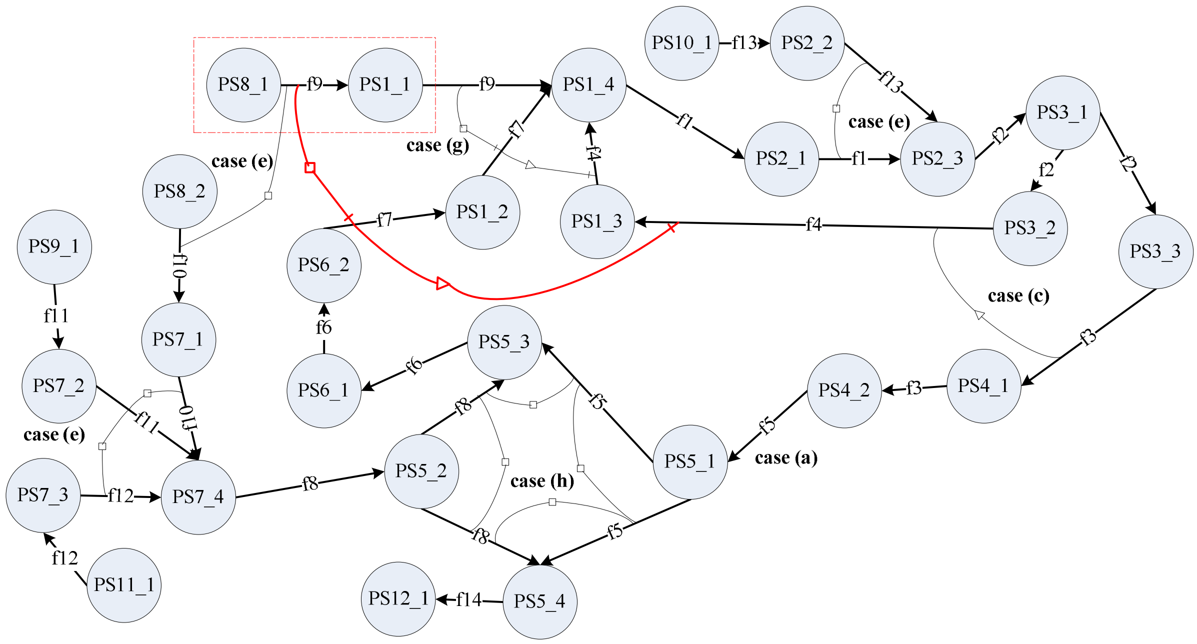

Figure 12: The functional structure model of an ICE cooling system in the space of parameters

5. CONCLUSION

Structural modelling is a flexible systematic approach to representing a normally functioning system, subject to changes, using a frame hierarchy. The knowledge about the system is stored in a knowledge base that is used to construct structural models. Combining several interconnected structural model makes it possible to explore different system aspects. Model transformations reduce time required for knowledge acquisition and representation by transferring logical operators between structural models. The system representation is built by experts and used to explore the system and to reason and make decisions about it. It is important to provide a convenient graphical notation for the knowledge that supports decision-making processes. Logical operators in structural models improve the understanding of system structural components and define relationships between them, which is necessary for understanding the principles of the system functioning.

In this paper, I discussed structural model transformations using logical operators in the space of behaviour and in the space of parameters. The transformations with logical operators and automatic logical transformations improve the quality of analysis and decision making.

The arrangement of components according to the system rules is essential for reasoning. It should be noted that structural components and functions, including the system rules, are organized in a certain order subject to certain restrictions, making, e.g., some combinations of flows impossible. This partially eliminates possible misunderstanding that can arise in the decision process if only partial knowledge is available. Appropriate tools are needed to automate decision process, structural model construction, and transformations. The future work is connected with the transformation algorithm implementation in the tool which supports structural modelling approach.

REFERENCES

Abu-Hanna A. and Jansweier W. N. H., 1994. Modelling Domain Knowledge Using Explicit Conceptualization. IEEE-Expert 9(5), pp. 53-64.

Ashby W.R., 2004. Principles of self – organizing system, in E:CO Special Double Issue Vol. 6 No. 1/2, reprint, (ed. Goldstein J.), pp.102-126
Capiluppi M., 2007, Fault Tolerance in Large Scale Systems: hybrid and distributed approaches, Ph.D. Thesis, p. 122
Grundspenkis J., 1997. Causal domain model driven knowledge acquisition for expert diagnosis system development, In: *Lecture Notes of the Nordic-Baltic Summer School on Applications of AI to Production Engineering*, Kaunas, Lithuania, (Wang K. and H. Pranevicius H., eds.), Kaunas University of Technology Press, pp. 251-268.
Grundspenkis J., 1999. Reasoning Supported by Structural Modelling. In: *Intelligent Design, Intelligent Manufacturing and Intelligent Management*, (Wang K. and Pranevicius H., eds.), Kaunas University of Technology Press, Technologija, pp. 57-100.
Grundspenkis J., 2002. Reasoning in structural model-based diagnosis. In proceedings of the 4th Int. Conf. on Quality, Reliability and Maintenance, QRM2002,Oxford, Professional Engineering Publishing, London, UK, pp. 295-298.
Kresken T., 1996, System Modelling. Technology Briefing Report, Sintef, Renaissance Esprit Project 22010, p. 11.
Oliva R., 2004. Model structure analysis through graph theory: partition heuristics and feedback structure decomposition. System Dynamics Review. Vol. 20, Issue: 4, pp. 313 – 336.
Tutte W.T., 2001. *Graph Theory*, Cambridge University Press, New York, p 333.
Zeltmate I., Grundspenkis J., 2008. Formal Method Of Functional Model Building Based On Graph Transformations. In proceedings of the International Workshop on Modeling & Applied Simulation, International Mediterranean and Latin American Modeling Multiconference, pp. 140-147

AUTHORS BIOGRAPHY

Ieva Zeltmate is researcher at the Department of System Theory and Design of Faculty of Computer Science and Information Technology in Riga Technical University. At the moment she is working on the final version of her thesis. Her scientific interests are associated with system analysis, design and development of knowledge representation systems, structural modelling, graphs, networks, complex and intelligent systems and knowledge representation within frames.