

# A Structure-Behavior Coalescence Method for Human-Computer Interaction System Requirements Specification

Yu-Chen Yang<sup>1</sup>(✉), Yi-Ling Lin<sup>1</sup>, and William S. Chao<sup>2</sup>

<sup>1</sup> Department of Information Management, National Sun Yat-Sen University, Kaohsiung, Taiwan  
{ycyang, yllin}@mis.nsysu.edu.tw

<sup>2</sup> Association of Chinese Enterprise Architects, Taipei, Taiwan  
architectchao@gmail.com

**Abstract.** For a system requirements method to specify a system as an integrated whole of that system's multiple views, it must be able to integrate the system structure and system behavior when specifying a system.

Current system requirements methods such as data-oriented, function-oriented, control-oriented and object-oriented, more or less, fail to specify a system as an integrated whole of that system's multiple views because they are not able to integrate the system structure and system behavior when specifying a system.

In this paper, we present a structure-behavior coalescence (SBC) method for human-computer interaction (HCI) system requirements specification (SyRS). Structure-behavior coalescence method includes three fundamental diagrams: (a) architecture hierarchy diagram, (b) component operation diagram and (c) interaction flow diagram. SBC method provides a sophisticated way to integrate the system structure and system behavior when used for HCI system requirements specifications.

**Keywords:** Structure-behavior coalescence method · System requirements specification · Human-Computer interaction

## 1 Introduction

A system is complex that it consists of multiple views such as structure view, behavior view, function view, data view, etc. The system requirements method specifies the multiple views of a system possibly using two different approaches. The first one is the non-integration approach and the second one is the integration approach.

The non-integration approach respectively picks a model for each view as shown in Fig. 1, the structure view has the structure model; the behavior view has the behavior model; the function view has the function model; the data view has the data model. These multiple models are heterogeneous and unrelated of each other, thus there is no way to put them into a conformity model [11, 17].

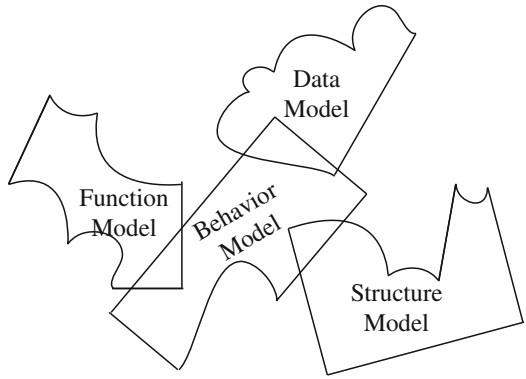


Fig. 1. The Non-integration approach

The integration approach, instead of picking many heterogeneous and unrelated models, will use only one single integration model as shown in Fig. 2. The structure, behavior, function and data views are all integrated in this one single integration model which represents an integrated whole of that system’s multiple views [5–8].

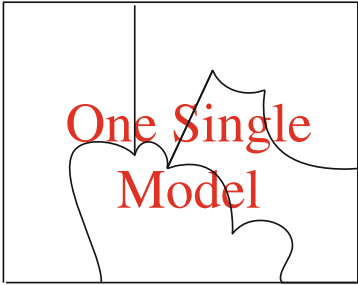
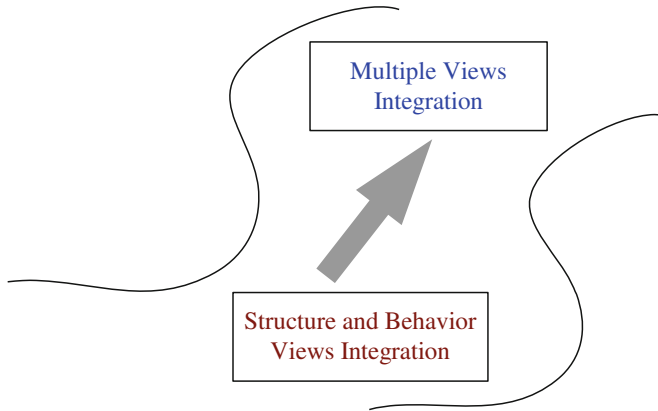


Fig. 2. The integration approach

Among those multiple views, the structure view focuses on the system structure which is described by components and their composition while the behavior view concentrates on the system behavior which involves interactions among external environment’s actors and components. Since structure and behavior views are the two most distinguished ones among multiple views, integrating the structure and behavior views becomes the most appropriate method for integrating multiple views of a system as shown in Fig. 3.

For requirements being able to describe a system as an integrated whole of that system’s multiple views, an ideal method for human-computer interaction (HCI) [18] system requirements specification should be based on a set of interacting components forming an integrated system structure and system behavior. The purpose of this paper is to explore this principle in depth. The whole paper is organized as follows. Section 1 is the introduction. Current methods such as data-oriented, function-oriented, control-oriented and object-oriented for requirements failing to describe a system as an

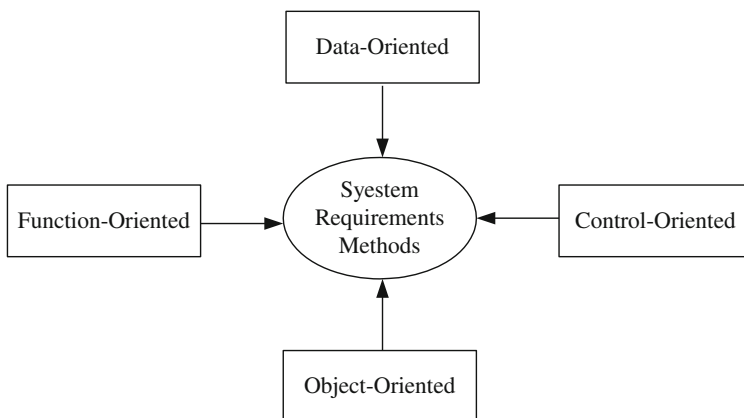


**Fig. 3.** Structure and behavior views integration facilitates multiple views integration

integrated whole of that system's multiple views are discussed in Sect. 2. Section 3 examines in detail the structure-behavior coalescence (SBC) method which integrates the system structure and system behavior of a system. Advantages of using structure-behavior coalescence as a method for HCI system requirements specification are delineated in Sect. 4. Section 5 is a summary.

## 2 Related Works

In general, current system requirements methods for HCI system requirement specification fall into three general categories: data, function, control and objects [14, 15], as shown in Fig. 4. Each of these methods, more or less, fails to describe a system as an integrated whole of that system's multiple views.



**Fig. 4.** Current system requirements methods

Data-oriented methods for HCI system requirements specification stress the system state as a data structure. Jackson System Development (JSD) [4] and Entity Relationship Modeling (ERM) [9] are primarily data-oriented. Data-oriented methods concentrate only on data and completely neglect to integrate the system structure and system behavior. Therefore, data-oriented methods for HCI system requirements specification belong to the non-integration approach and will never become an ideal requirement method.

Function-oriented methods for HCI system requirements specification take the primary view of the way a system transforms input data into output data. Each transformation from input data into output data demonstrates a function of the system. A system may contain many such kinds of functions which represent the function view of the system. Classical Structured Analysis (SA) [10] fits into the category of process-based methods, as do Structured Analysis and Design Technique (SADT) [19] and Structured Systems Analysis and Design Method (SSADM) [1]. Process-based methods concentrate only on the function view and completely neglect to integrate the system structure and system behavior. Just like data-oriented methods, function-oriented methods for HCI system requirements specification belong to the non-integration approach and will never become an ideal system requirements method.

Control-oriented methods for HCI system requirements specification emphasize synchronization, deadlock, exclusion, concurrency, and process activation of a system. Petri Net [22] and Flowcharting [2] are primarily control-oriented. Control-oriented methods concentrate only on the control view and completely neglect an integrated structure and behavior views which grasps the essential properties of a system. Just like data-oriented and function-oriented methods, control-oriented methods for HCI system requirements specification belong to the non-integration approach should not and will never become an ideal requirement method.

Object-oriented methods for HCI system requirements specification describe the system as classes of objects and their behaviors. Object-oriented Analysis (OOA) [3], fitting into the category of object-oriented methods, looks at the problem domain, with the aim of producing a conceptual model of the information that exists in the area being analyzed. The result of object-oriented analysis is a description of what the system is behaviorally required to do, in the form of a conceptual model. That will typically be presented as a set of use cases and a number of activity diagrams. Object-oriented methods stress both the structure view and the behavior view, but not an integrated structure and behavior views. Object-oriented methods do not emphasize to integrate the system structure and system behavior. Like data-oriented, function-oriented and control-oriented methods, object-oriented methods for HCI system requirements specification belong to the non-integration approach and will never become an ideal system requirements method.

### 3 The Approach

Structure-behavior coalescence (SBC) method for HCI system requirement specification uses three fundamental diagrams to accomplish the HCI system requirements specification. These diagrams are: (a) architecture hierarchy diagram, (b) component operation diagram, and (c) interaction flow diagram.

#### 3.1 Architecture Hierarchy Diagram

Structure-behavior coalescence (SBC) method for HCI system requirement specification uses an architecture hierarchy diagram (AHD) to specify the multi-level (hierarchical) decomposition and composition of a human-computer interaction system.

As an example, Fig. 5 shows that Multimedia KTV is composed of Song\_Selection and Songs; Songs is composed of Song\_1 and Song\_2. Among them, Multimedia KTV and Songs are aggregated systems while Song\_Selection, Song\_1 and Song\_2 are non-aggregated systems.

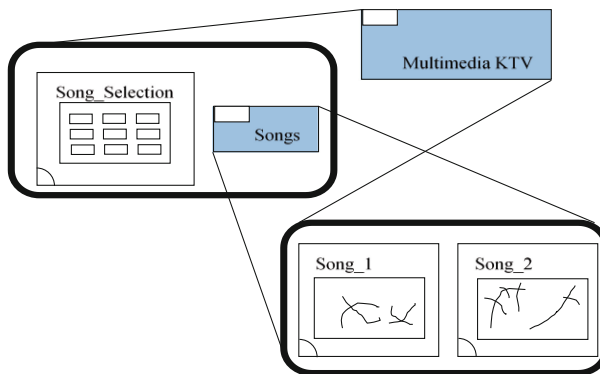


Fig. 5. AHD of the multimedia KTV

#### 3.2 Component Operation Diagram

Structure-behavior coalescence (SBC) method uses a component operation diagram (COD) to specify all components' operations in a human-computer interaction system.

An operation provided by each component represents a method of that component [3]. Figure 6 shows a COD of the Multimedia KTV. In the figure, component Song\_Selection has two operations: Select\_Song\_1 and Select\_Song\_2; component Song\_1 has two operations: Broadcast\_Song\_1 and Sing\_Song\_1; component Song\_Selection has two operations: Broadcast\_Song\_2 and Sing\_Song\_2.

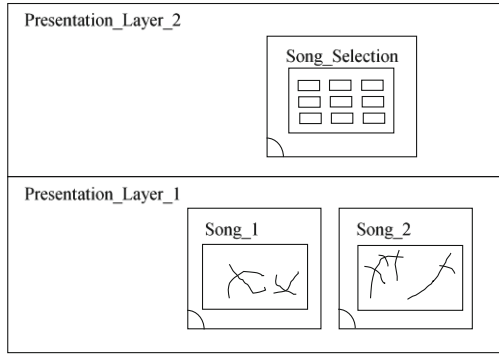


Fig. 6. FD of the multimedia KTV

For a human-computer interaction system, we use a component operation diagram (COD) to design all components' operations. Figure 7 shows a COD of the Multimedia KTV. In the figure, component Song\_Selection has two operations: Select\_Song\_1 and Select\_Song\_2; component Song\_1 has two operations: Broadcast\_Song\_1 and Sing\_Song\_1; component Song\_Selection has two operations: Broadcast\_Song\_2 and Sing\_Song\_2.

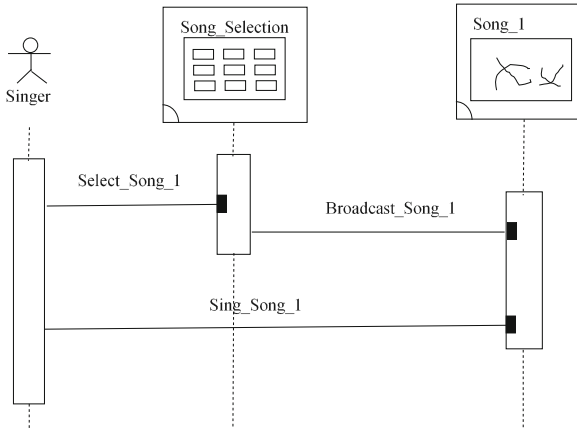


Fig. 7. IFD of the “KalaOK\_Song\_1” behavior

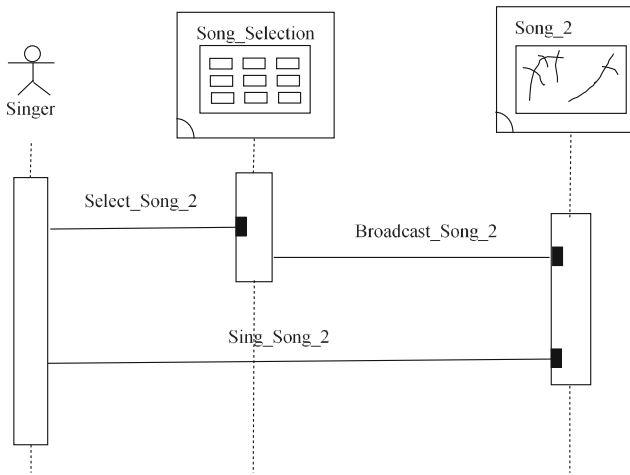
### 3.3 Interaction Flow Diagram

Structure-behavior coalescence method uses interaction flow diagram (IFD) to define all individual behavior of a human-computer interaction system. In a human-computer interaction system, if the components, and among them and the external environment's actors to interact, these interactions will lead to the system behavior. That is, “interaction” plays an important factor in integrating the systems structure and systems behavior for a human-computer interaction system.

The overall behavior of a human-computer interaction system consists of many individual behaviors. They tend to be executed concurrently [16, 20, 21]. Each individual behavior represents an execution path. We use an interaction flow diagram (IFD) to define this individual behavior. For example, the overall Multimedia KTV's behavior includes two behaviors: KalaOK\_Song\_1 and KalaOK\_Song\_2.

Figure 7 shows the IFD of the KalaOK\_Song\_1 behavior. First, actor Singer interacts with the Song\_Selection component through the Select\_Song\_1 operation call interaction. Next, component Song\_Selection interacts with the Song\_1 component through the Broadcast\_Song\_1 operation call interaction. Finally, actor Singer interacts with the Song\_1 component through the Sing\_Song\_1 operation call interaction.

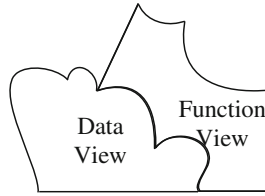
Figure 8 shows the IFD of the KalaOK\_Song\_2 behavior. First, actor Singer interacts with the Song\_Selection component through the Select\_Song\_2 operation call interaction. Next, component Song\_Selection interacts with the Song\_2 component through the Broadcast\_Song\_2 operation call interaction. Finally, actor Singer interacts with the Song\_2 component through the Sing\_Song\_2 operation call interaction.



**Fig. 8.** IFD of the “KalaOK\_Song\_2” behavior

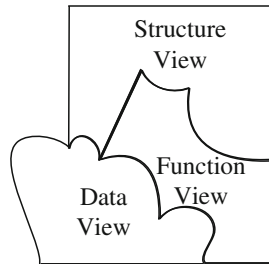
## 4 Result and Discussions

In the SBC method for HCI system requirement specification, an operation formula includes (a) operation name, (b) input parameters, and (c) output parameters. Since input/output parameters represent the data view and an operation represents the function view, so data and function views are well integrated in the SBC method for HCI system requirement specification as shown in Fig. 9.



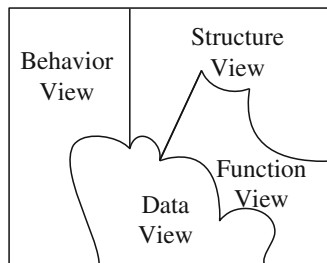
**Fig. 9.** Data and function views integration

In the SBC method for HCI system requirement specification, a component operation diagram specifies all components' operations in a system. These all components' operations represent the function view of the system. Since components' operations belong to the function view and components belong to the structure view, so data, function and structure views are well integrated in the SBC method for HCI system requirement specification as shown in Fig. 10.



**Fig. 10.** Data, function and structure views integration

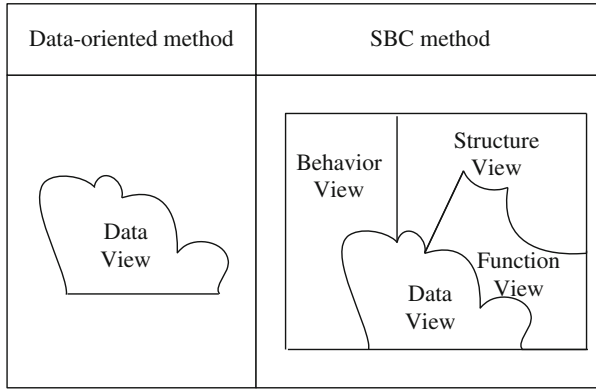
Also in the SBC method for HCI system requirement specification, an interaction flow diagram is constructed by specifying those "interactions" among the external environment's actors and the components. Since external environment's actors and components belong to the structure view and interaction flow diagrams belong to the behavior view, so data, function, structure and behavior views are well integrated in the SBC method for HCI system requirement specification as shown in Fig. 11.



**Fig. 11.** Data, function, structure and behavior views integration

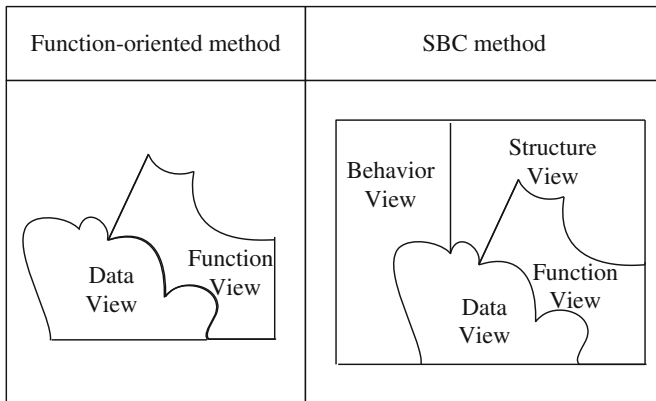


Let us compare the data-oriented method with the SBC method. As shown in Fig. 12, the data-oriented method for HCI system requirement specification concentrate only on the data view and the SBC method for HCI system requirement specification has the data, function, structure and behavior views all integrated.



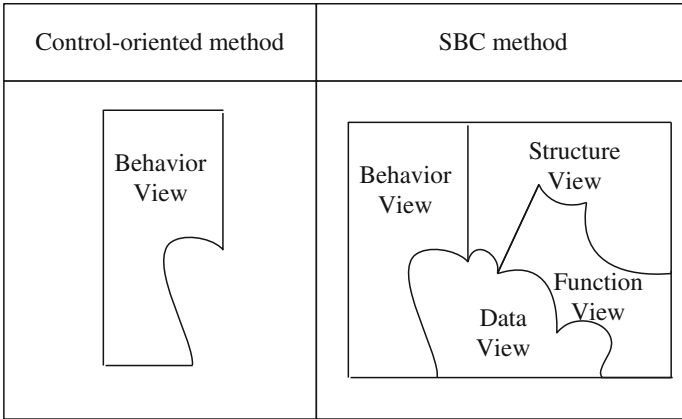
**Fig. 12.** Data-oriented method vs. SBC method

Let us compare the function-oriented method with the SBC method. As shown in Fig. 13, the function-oriented method for HCI system requirement specification concentrate only on the data and function views and the SBC method for HCI system requirement specification has the function, structure and behavior views all integrated.



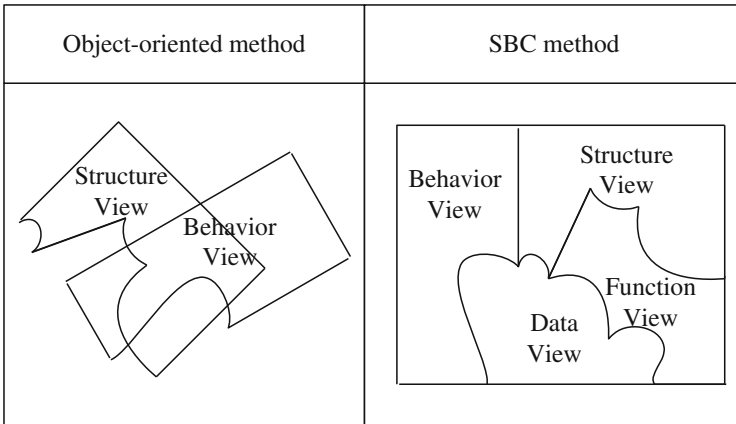
**Fig. 13.** Function-oriented method vs. SBC method

Let us compare the control-oriented method with the SBC method. As shown in Fig. 14, the control-oriented method for HCI system requirement specification concentrate only on the behavior view and the SBC method for HCI system requirement specification has the function, structure and behavior views all integrated.



**Fig. 14.** Control-oriented method vs. SBC method

Let us compare the object-oriented method with the SBC method. As shown in Fig. 15, the object-oriented method for HCI system requirement specification has the structure and behavior views unrelated and the SBC method for HCI system requirement specification has the function, structure and behavior views all integrated.



**Fig. 15.** Object-oriented method vs. SBC method

## 5 Conclusions

For a system requirements method to specify a human-computer interaction system as an integrated whole of that system’s multiple views, it must be able to integrate the system structure and system behavior when specifying a system.

Current human-computer interaction system requirements methods such as process-based, behavioral and object-oriented, more or less, fail to specify a human-computer

interaction system as an integrated whole of that system's multiple views because they are not able to integrate the system structure and system behavior when specifying a human-computer interaction system.

The characteristics of the SBC method for HCI system requirement specification lie in its integrating the structure and behavior views hence it is able to integrate the multiple views of a human-computer interaction system. Therefore, for HCI system requirements specification, SBC method is more advanced than the process-based, behavioral and object-oriented methods.

## References

1. Ashworth, C.: SSADM: A Practical Approach, 1st edn. McGraw-Hill Book Company Ltd., London (1990)
2. Bashe, C.: IBM's Early Computers. The MIT Press, Cambridge (1986)
3. Booch, G.: Object-oriented Analysis and Design with Applications, 3rd edn. Addison-Wesley, Boston (2007)
4. Cameron, J.R.: The Jackson Approach to Software Development. IEEE Computer Society Press, Silver Spring (1989)
5. Chao, W.S., General Systems Theory 2.0: General Architectural Theory Using the SBC Architecture. CreateSpace Independent Publishing (2014)
6. Chao, W.S.: Systems Thinking 2.0: Architectural Thinking Using the SBC Architecture Description Language. CreateSpace Independent Publishing (2014)
7. Chao, W.S.: A Process Algebra For Systems Architecture: The Structure-Behavior Coalescence Approach. CreateSpace Independent Publishing Platform (2015)
8. Chao, W.S.: An Observation Congruence Model For Systems Architecture: The Structure-Behavior Coalescence Approach. CreateSpace Independent Publishing Platform (2015)
9. Chen, P., et al.: The Entity-Relationship Model - Toward a Unified View of Data. ACM Transactions on Database Systems **1**(1), 9–36 (1976)
10. DeMarco, T.: Structured Analysis and System Specification. Prentice Hall, Upper Saddle River (1979)
11. Dennis, A., et al.: Systems Analysis and Design, 4th edn. Wiley, New York (2008)
12. Date, C.J.: An Introduction to Database Systems, 8th edn. Addison Wesley, Boston (2003)
13. Elmasri, R.: Fundamentals of Database Systems, 6th edn. Addison Wesley, Boston (2010)
14. Grady, J.O.: System Requirements Analysis, 2nd edn. Elsevier, Amsterdam (2013)
15. Hatley, D.J., et al.: Process for System Architecture and Requirements Engineering, 1st edn. Addison-Wesley, New York (2000)
16. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall, Upper Saddle River (1985)
17. Kendall, K., et al.: Systems Analysis and Design, 8th edn. Prentice Hall, Boston (2010)
18. MacKenzie, I.S.: Human-Computer Interaction: An Empirical Research Perspective. Morgan Kaufmann, San Francisco (2013)
19. Marca, D.A., et al.: SADT: Structured Analysis and Design Technique. McGraw-Hill, New York (1988)
20. Milner, R.: Communication and Concurrency. Prentice-Hall, Upper Saddle River (1989)
21. Milner, R.: Communicating and Mobile Systems: the  $\pi$ -Calculus, 1st edn. Cambridge University Press, New York (1999)
22. Reisig, W.: A Primer in Petri Net Design. Springer, Heidelberg (1992)