Structured Multi-view Modeling by Tabular Notation

Xiuna Zhu
Institut für Informatik
Technische Universität München
Boltzmannstr. 3, D-85748, Germany
Email: zhux@in.tum.de

Dongyue Mou
Fortiss GmbH, Guerickestr. 25,
D-81735 München, Germany
Email: mou@fortiss.org

Daniel Ratiu
Fortiss GmbH, Guerickestr. 25,
D-81735 München, Germany
Email: ratiu@fortiss.org

***Abstract*—The growth of software complexity and high degree of dependencies between functionalities motivates the use of models during requirements engineering. Hence, readability and comprehensibility of currently requirements specification techniques should be increased. Additionally, multi-view modeling and tabular expression are widely accepted techniques in requirements documentation. We present a tool that allows structured multi-view modeling of the behavior of the system by means of tabular notation. Our tool provides various table patterns to support different behavior views, which leverage the advantages of tabular specification, e.g., unambiguous, precise, and easier to read, analyses and communicate. Our aim is to reduce the complexity in the development of software systems.**

***Index Terms*—Tabular Specification, Multi-View Modeling, Model-based Requirements Engineering**

I. Introduction

Nowadays software systems are employed in in every domain. However, the growth of their complexity and high degree of dependencies between functionalities demands adequate specification approaches. Variety of approaches, such as model-driven requirements engineering, have been proposed to face the difficulties in the development of software systems. Multi-view modeling is considered among them as a technique to reduce the complexity [1].

Tabular expressions are used in the process of requirements documentation by clarifying the logical formulas [2]. Therefore, software engineers often use tables to specify the behavior of components in a system, but only as documentation. We advocate that tables can be used beyond the documentation. Different kinds of tables can be used to specify system interfaces and behaviors. And we can analyses their functional properties by analyse the tables. In so doing, tabular specification is no more only considered as documentation but exploited as a model of the system.

Due to tabular description techniques' advantages, e.g., they are unambiguous, precise, and intuitive, they are promising for practical system development [3]. Several types of tables have been defined in previous researches of tabular notation for state machine based models [4], [5]. This paper focuses on tabular specification, namely we take tables as pragmatic specification formalism of systems' behavior. For one thing, by the definition of the tabular notation for modeling and translating from tables into formulas of predicate logic and vice versa, we provide a bridge between the conciseness of readable specification and the preciseness of mathematical methods. For another, by manipulation of the tabular specification, we attempt to obtain the composition of the tabular specification.

To achieve the tool support for tabular specification, we have focused our study on following main research questions.

Q1) How to specify the functional requirements by tabular specification? This research question is concerned with how to use tables as a pragmatic specification formalism for a both precise and readable specification of system's functional requirements. Providing multiple consistent views of the system on different levels of abstraction was discussed and the definition of the tabular notation was provided.

Q2) How to clarify formal semantics of tabular specification? The formal system model should be used to define the semantics of the employed description techniques.

Q3) How to cross the functional boundary of single component to obtain a systematical view of the functionalities? To achieve the composition and decomposition of the tabular specification, we discuss on how to compose tabular-specified models and ensure the consistency and completeness of the composite model.

Q4) How to derive the optimization of system architecture from the tabular specification? To analysis the property of the system which specified by tabular specification, we can via the analysis of the property of tables. By translating tables into logical formulas, which define precise semantics for them, the dependency relation between outputs and inputs will be achieved. By dependencies encapsulated and automatically resolved, the structure features and refinement of system can be obtained, and the optimization of system architecture by manipulation on tabular specification can be achieved.

II. Various Tabular Patterns for Three Views

The proposed tabular modeling tool is the implementation and extension of the previous work [6]. One goal is that our tool supports the modeling process in a user-friendly manner. Various table patterns with manifold structures are provided. Therefore, user can choose the adaptive format as table pattern or customize the structure of the tables.

In our case, a system can be structured into a set of logical entities, which are connected via communication channels.

978-1-4799-3032-6/14/$31.00 © 2014 IEEE

RE 2014, Karlskrona, Sweden

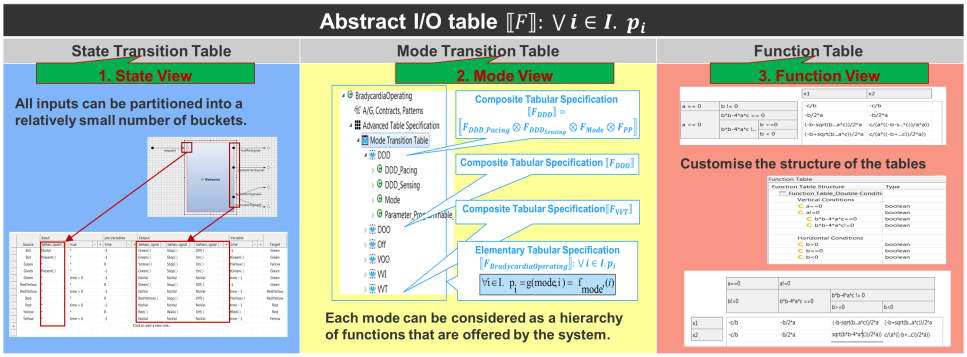

Figure 1. Supported Table Patterns for Tabular Specification

Each table can specify a functionality of system, which is a combination in form of a predicate of logical entities. The structured tabular models are classified to two types: *Elementary Tabular Specification* and *Composite Tabular Specification*. Tabular specification is suitable for describing the structured composition and decomposition of such systems by the manipulation on tables which are used to describe the system.

As shown in Fig.1, three modeling views are supported for elementary tabular specification. To achieve those views, we provided three basic table patterns: state transition tables, mode transition tables, and function tables. Here, modes represent the logical states of a functionality. Each mode can be considered as a hierarchy of functionalities that are offered by the system. Each table pattern has a general specific table structure, but the structure of the table can be customized arbitrary. So the plastically of table structure can be kept in the proposed tabular specification.

III. Illustrative Case Study

Unlike requirement documents written in natural language, which allow limited semantic and syntactic checks, we captured and modeled the requirements precisely by proposed tabular specification tool. It is integrated in model-based development tool AutoFOCUS3. Many case studies are carried out with tabular specification in AF3 modelling environment, and tool demos are available at [7]. We illustrate the functionality of our tool by the use of state transition tables for a traffic lights control system, mode transition tables for the pacemaker challenge project, and function tables for a quadratic equations solver.

Except specifying the requirements in a well-structural format, the tool concentrates on the questions if the tabular models match the customer's requirements. The problems which is not easy to find in the requirements document, like if the requirement is complete and consistent, if there is a conflict existing between two requirements, can be detected by tool.

IV. Conclusion

This work presents the following contributions to the current researches in formal specification:

1) **From tabular expression to tabular specification** Based on the FOCUS modeling theory, we define the notion of tabular specification. Instead of using graphical notation to describe the behavior of interactive systems, we define the tabular notation for state machine-based specifications to obtain a tabular specification of system behavior. By manipulation of the tables, such as composition and projection, we yield tabular specified components and the composition of the components.
2) **Tool support for various table patterns** The tabular specification tool supports tabular specification, simulation, property analysis, and code generation. Various table patterns with manifold structures are provided and supported, which can help shortening the cells content and bring tables to a nice form.
3) **Structuring features for complex systems** The large number of states is a defect which affects application of current specification techniques to describe the behavior of practical systems. Structured tabular specification was proposed to describe the behavior of the hierarchically tables. The illustrative case studies showed that in structural multi-view tabular specification the number of states can be significantly reduced and the complexity of behavior can reduced.

References

[1] M. Broy, "Software and system modeling: Structured multi-view modeling, specification, design and implementation." in *Conquering Complexity*, M. Hinchey and L. Coyle, Eds. Springer, 2012, pp. 309–372.

[2] R. Janicki and A. Wassyng, "Tabular expressions and their relational semantics," *FUNDAMENTA INFORMATICAE*, vol. 67, no. 4, pp. 343–370, 2005.

[3] M. K. Zimmerman, K. Lundqvist, and N. G. Leveson, "Investigating the readability of state-based formal requirements specification languages," in *ICSE*, W. Tracz, M. Young, and J. Magee, Eds. ACM, 2002, pp. 33–43.

[4] I. Bourguiba and R. Janicki, "Table-based specification techniques," in *Computers Industrial Engineering, 2009. CIE 2009. International Conference on*, july 2009, pp. 1520 –1525.

[5] M. Herrmannsdörfer, S. Konrad, and B. Berenbach, "Tabular notations for state machine-based specifications," *Crosstalk*, vol. 21, no. 3, pp. 18–23, 2008.

[6] X. Zhu, "A formal model for servie-based behavior specification using stream-based i/o tables," in *FACS2013*, Nanchang, China, 2013.

[7] Fortiss. (2013, Sep.) Homepage of the autofocus 3. [Online]. Available: http://af3.fortiss.org