Segmentation and Conceptual Chunking in Embedded Real-Time System Design

Bernhard Rumpler, Martin Schlager
TTTech Computertechnik AG
Schoenbrunner Strasse 7, 1040 Vienna, Austria
{rumpler,schlager}@tttech.com

Ever growing complexity is one of the major problems in embedded real-time system design as the complexity may lead to latent design faults. A prominent example for this is automotive electronics, where design faults are a major cause for recalls. Another important aspect for growing complexity is the current trend towards computer system architectures, where application subsystems from different vendors and with different criticality levels are integrated within the same hardware. Examples for such architectures in the automotive and avionics domains are AUTOSAR [1], Integrated Modular Avionics (IMA) [2] and DECOS [3]. Considering complexity factors of computer system architectures is fundamental to be able to create comprehensible systems where design faults are less likely to remain undetected.

Cognitive complexity is the amount of cognitive resources that are needed for a task. If the complexity of a task is high, this becomes manifest in increased time we require for this task, and in a higher error rate. It is thus important to keep the complexity of a system low in order to allow fast development and low design failure rates. According to *relational complexity theory* [4], the two major strategies humans apply when they have to deal with complex tasks are *segmentation* and *conceptual chunking*. Segmentation means to split a complex task into a number of subtasks that do not exceed our processing capacity. Then, these subtasks can be processed serially. Conceptual chunking means to create representations of problems that reduce their dimensionality, i.e., by generating some kind of abstraction, thereby making some relational information inaccessible. This paper considers the fundamental characteristics of time-triggered (TT) and event-triggered (ET) real-time systems in the context of relational complexity theory.

In a purely ET approach all actions of the system are in fact reactions that are caused by events. In the TT approach all actions of a system are triggered by the progression of real time. This basic characteristic has important implications on how we think about such systems: The instants of real time where actions of the systems are triggered are a focal point for segmentation and conceptual chunking in TT systems. In ET systems it is the occurrence of events that provides a basis for conceptual structuring. In this paper it is explained why it is often easier to fully understand the behavior of a TT system, compared to an ET system. Some concepts of TT systems are conceptually less complex than those of ET systems. State messages, for example, are conceptually simpler than event messages – this can be explained with relational complexity theory.

Furthermore, the provision of state message based data sharing interfaces (*temporal firewalls*) has even further characteristics that simplify the segmentation of the system. When considering the technical characteristics here we can draw interesting parallels with psychological concepts: The state message interfaces that were originally introduced to increase the technical autonomy of components (e.g., to avoid control error propagation) also serve as segmentation points for understanding the system. A pure state message interface allows us to consider a component in isolation as it makes no references to other components. Thus, a state message interface usually has very low relational complexity, compared to most event message based interfaces. It is not only a borderline to avoid flow control propagation, but it also serves to increase the conceptual separability of the components which is important for segmentation and conceptual chunking.

Another important characteristic of a TT system is the provision of a global sparse time base. This sparse time simplifies the interpretation of time-stamped events and allows the construction of various fault tolerance mechanisms at the architectural level that would otherwise not be possible, e.g., replica-deterministic components. When implementing these fault tolerance services at the application level this results in system characteristics that have been identified as highly complex in various psychological experiments.

The goal of this work is to support *design comprehension* by analyzing various approaches of system design paradigms and by identifying their influence on the complexity of various design tasks. An example of this is the DECOS approach, where a complex system is built from well-defined and comprehensible components that minimize the system-level complexity.

There are various approaches and practices in embedded real-time system design that are intended to reduce the system complexity. However, there exists just little scientific work that is really based on the theoretical background of cognitive psychology. This paper aims at bridging this gap by providing some cognitive support theories for low complexity in embedded real-time system design.

[1] AUTOSAR GbR. AUTOSAR – Technical Overview V2.0.1, June 2006

[2] Aeronautical Radio, Inc., 2551 Riva Road, Annapolis, Maryland, 21401. ARINC Specification 651: Design Guide for Integrated Modular Avionics, November 1991.

[3] R. Obermaisser, P.Peti, B.Huber, and C. El Salloum. DECOS: An Integrated Time-Triggered Architecture. *E&I Journal (Journal of the Austrian Professional Institution for Electrical and Information Engineering)*, 3, March 2006

[4] Graeme S. Halford, William H. Wilson, and Steven Phillips. Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. *Behavioral and Brain Sciences*, 21:803—832, 1998