



Functional Requirements Specifications of Automotive Software Systems

Sabine Rittmann

International Workshop on Automotive
Requirements Engineering (AuRE 2006)

Minneapolis/St.Paul, 09/11/2006



Software & Systems Engineering

The ten Centres of Competence



Software
Architecture &
Components



Model Driven
Software
Engineering



IT Security



Embedded
Systems



System
Development
Processes



Theorem Proving



Mobility &
Context Aware
Architecture



RE Requirements
Engineering



Tool Support



Software
Maintenance



Project partners

sd&m
software design & management



BMW Group

Microsoft

4SOFT



BOSCH

skynamics

SIEMENS



BSI

ETAS

Deutsche Bank



HypoVereinsbank



msg
systems

VALIDAS
validated quality



O₂

Information Society
Technologies

...T...Systems

Department of Informatics
Chair IV: Software & Systems Engineering

TUM
TECHNISCHE
UNIVERSITÄT
MÜNCHEN

3



Today...

- ...current/intermediate research results from the mobilSoft project (<http://www.mobilsoft.info>)
- 2 OEMs + 7 suppliers + 4 research institutes
- Concepts and processes for the service-based / function-oriented, seamless, model-based development of future automotive software systems
- You are welcome to discuss and criticize!

TUM
TECHNISCHE
UNIVERSITÄT
MÜNCHEN

Department of Informatics
Chair IV: Software & Systems Engineering

4



Motivation – RE for automotive systems

- State of the art: Requirements are usually obtained in an **informal** form (text, UML Use Cases, ...)
- Problems:
 - imprecise, ambiguous, contradictory descriptions of the system under consideration
 - Gap between the RE phase (with its **informal, textual** system descriptions) and the design phase (with its **formal** models)



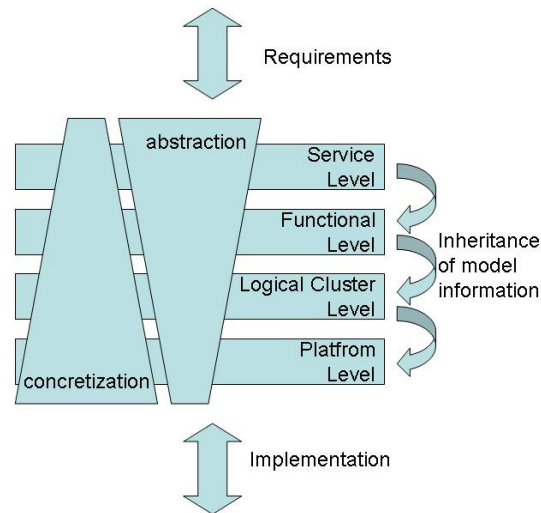
Idea

- Formalization of informal, textual requirements (model-based requirements engineering)
- Here: Only **functional** requirements are looked at
- **Service** = formal description of a functional requirement
- **Formal service description language (ForSeL)**
- Consequently:
 - Due to formalism: precise meaning
 - Contradictions can be detected
 - Abstract model of the system behavior (issues concerning distribution, technical realisation are not considered)
 - Enabling development according to different levels of abstraction (see following slide)



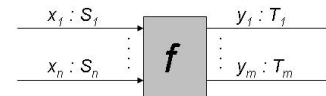
Abstraction Levels

- Service Level: Black box behavior
 - Functional Level: White box behavior
 - Logical Cluster Level: Distributable units
 - Platform Level: Hardware
- } Behavior
 } Structure



Basis: The FOCUS theory

- ForSeL (formal service description language) is formally founded by FOCUS
- FOCUS
 - Stream-based system theory: a system is a function relating input streams to output streams
 - Adequate means for describing the behavior of reactive, distributed systems
 - Here: Black box view onto a system
 - set of data messages M ("actions" in our setting)
 - M^* (M^∞) denotes the set of finite (infinite) streams (sequences over M)
 - $M\omega = M^* \cup M^\infty$.
 - A stream can be represented by the function s where $s(t)$ contains the message processed in the stream s at the point of time t .
 - Example: $s = \langle m_1, m_2, m_3, m_4 \rangle$ or $s' = \langle m, m, m, m, m, \dots \rangle$





Basic Elements: Actions

- Action = relevant behavior that can be observed at the system border, e.g. „toggle switch pressed“, „Window moves up“
- Basic elements of which a specification is comprised
- System = relating (*input*) actions to (*output*) actions
 $s \in Act^{\omega}$
- Action A can have exactly one of the following assignments (at the point of time t):
 - $A=1$: A is active at the point of time t, i.e. $A \in s.t$.
 - $A=0$: A is not active at the point of time t, i.e. $\neg A \in s.t$
 - $A=\uparrow$: A starts at the point of time t, i.e. $A \uparrow \in s.t$, consequently $\neg A \in s.(t-1)$ and $(A \vee A \downarrow) \in s.(t+1)$ (*start event*)
 - $A=\downarrow$: A stops at the point of time t, i.e. $A \downarrow \in s.t$, consequently $A \in s.(t-1)$ and $(\neg A \vee A \uparrow) \in s.(t+1)$ (*stop event*)



Basic Elements: Actions

- Actions can be clearly divided into
 - *environment actions* $Env \subset Act$ and
 - *system reactions* $Sys \subset Act$,
 whereas $Act = Env \cup Sys$ and $Env \cap Sys = \emptyset$.



- Service =
 - Formalized **functional** requirement
 - **Partial** behavior
- Service is given by $R = [P] (T \rightarrow S)$
 - **Precondition** P $P = (A_1 = b_1) \wedge (A_2 = b_2) \wedge \dots \wedge (A_n = b_n)$
with $A_i \in \text{Act}, b_i \in \{0,1\}, i = 1, \dots, n$
 - **Triggering event** T $T = (B_1 = e_1) \wedge (B_2 = e_2) \wedge \dots \wedge (B_n = e_n)$
with $B_i \in \text{Act}, e_i \in \{\uparrow, \downarrow\}, i = 1, \dots, n$
 - **System reaction** S $S = (C_1 = e_1) \wedge (C_2 = e_2) \wedge \dots \wedge (C_n = e_2)$
with $C_i \in \text{Sys}, e_i \in \{\uparrow, \downarrow\}, i = 1, \dots, n$
- Semantics of $R = [P] (T \rightarrow S)$ $[[[P](T \rightarrow S)]] =$
 $\{s \in \text{Act}^\omega \mid \forall t \in \mathbb{N} :$
 $((P \in s.t \wedge T \in s.t) \Rightarrow (\exists t' > t : S \in s.t'))\}$



- Conjunction of formal requirements
 $\text{Spec} = R_1 \otimes R_2 \otimes R_3 \otimes \dots \otimes R_n$
- Let $R_1 = [P_1](T_1 \rightarrow S_1)$ and $R_2 = [P_2](T_2 \rightarrow S_2)$
 then their composition is defined as follows:
 $[[[P_1](T_1 \rightarrow S_1) \otimes [P_2](T_2 \rightarrow S_2)]] =$
 $\{s \in \text{Act}^\omega \mid \forall t \in \mathbb{N} :$
 $((P_1 \in s.t \wedge T_1 \in s.t) \Rightarrow (\exists t' > t : S_1 \in s.t')) \wedge$
 $((P_2 \in s.t \wedge T_2 \in s.t) \Rightarrow (\exists t' > t : S_2 \in s.t'))\}$



Properties of the composition operator

- Identical reaction pattern:

$$[[[P_1](T \rightarrow S) \otimes [P_2](T \rightarrow S)]] = [[[P_1 \vee P_2](T \rightarrow S)]]$$
- Identical precondition and reaction:

$$[[[P](T_1 \rightarrow S) \otimes [P](T_2 \rightarrow S)]] = [[[P]((T_1 \vee T_2) \rightarrow S)]]$$
- Identical precondition and identical triggering event:

$$[[[P](T \rightarrow S_1) \otimes [P](T \rightarrow S_2)]] = [[[P](T \rightarrow (S_1, S_2))]]$$
- Commutative and associative

(proofs: cf. Paper)

- No precondition:

$$[[T \rightarrow S]] = \{s \in \text{Act}^\omega \mid \forall t \in N : (T \in s.t \Rightarrow (\exists t' > t : S \in s.t'))\}$$



Refinement

- Deductive design of requirements specifications
- Collection of refinement rules
- S2 is a **behavioral refinement** of S1 if and only if the observable behavior of S2 is also an observable behavior of S1 $[[S2]] \subseteq [[S1]]$
- Behavioral refinement is stepwise reduction of non-determinism
- Due to the definition of our notion of refinement (set inclusion) we obtain the following properties:
 - **Transitivity** (stepwise application of refinement rules)
 - **Compositionality** (local refinement $\rightarrow$ global refinement)



Refinement

- The refinement of a service can be achieved by:

- Enhancing the service domain of a reaction pattern

- Alternative preconditions

$$[P_1](T \rightarrow S) \otimes [P_2](T \rightarrow S) = [P_1 \vee P_2](T \rightarrow S)$$

- Alternative triggering events

$$[P](T_1 \rightarrow S) \otimes [P](T_2 \rightarrow S) = [P]((T_1 \vee T_2) \rightarrow S)$$

- Restricting the service reaction

$$[P](T \rightarrow S_1) \otimes [P](T \rightarrow S_2) = [P](T \rightarrow (S_1, S_2))$$



Notation

	R_1	...	R_n
Precondition P:			
E_1			
...			
E_n			
S_1			
...			
S_n			
Triggering Event T:			
E_1			
...			
E_n			
S_1			
...			
S_n			
Reaction S:			
S_1			
...			
S_n			



Methdological Support

- Identify the set of relevant actions
- Classify the set of actions (Sys and Env)
- Fill in the table entries (iteratively) and perform consistency checks
 - Determine reaction patterns
 - Determine preconditions
 - Perform plausibility checks
- Extension of specification terms / actions
 - Addition
 - Deleting



Example

	2a	2b	2c	2d
P				
SWO				
SWC				
CD				
CL			WO	
WET				
WEB			WO	0
WO		0	0	0
WC			CD	1
T				
SWO				
SWC				
CD	↑	↑	↑	↑
CL				
WET				
WEB				
WO				
WC				
S				
WO	↑	↑	↑	↑
WC				



Future Work

- Interrupt / abortion of services
- Translation of textual requirements into formal ForSeL expressions by ‚standard‘ text patterns
- Totalization of behavior (as services are partial)
- Larger scale case studies
- Enrichment with quantitative timing information
- Distinction into different preconditions [SOSE 2006]



Questions???

- ... More questions? [rittmann „at“ in.tum.de](mailto:rittmann_at_in.tum.de)