

SADS Exercises – Architecture Decisions



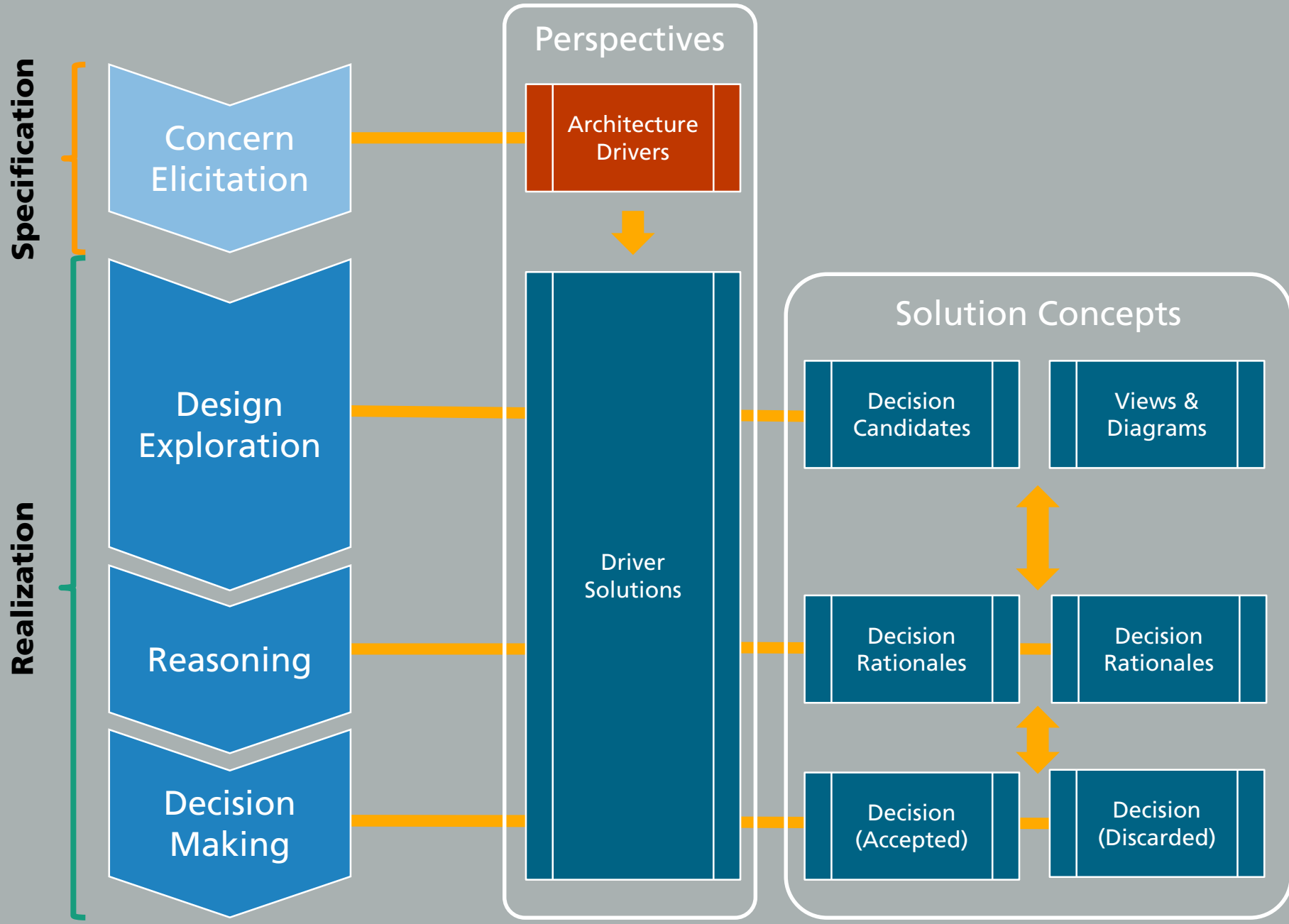
TU Kaiserslautern, SS2018

**Lecture “Software and System
Architecture (SSA)”**

Dr. Pablo Oliveira Antonino
pablo.antonino@iese.fraunhofer.de

Dr. Andreas Morgenstern
andreas.morgenstern@iese.fraunhofer.de

Architecture Decision Making



Architecture Driver Template

Categorization		Responsibilities	
Driver Name	<i>Concise short name</i>	Supporter	<i>Stakeholders supporting the driver</i>
Driver ID	<i>Unique identifier</i>	Sponsor	<i>Stakeholders paying for the driver</i>
Status	<i>[Open, Elicited, Under Design, Designed, Under Realization, Realized, Done]</i>	Author	<i>Responsible for filling this template</i>
Priority	<i>[High - Medium - Low]</i>	Inspector	<i>Stakeholders reviewing this driver</i>

Description		Quantification
Environment	<i>Context and/or initial situation applying to this driver</i>	<ul style="list-style-type: none"> ▪ <i>Measurable effects applying to the environment</i>
Stimulus	<i>The event, trigger or condition arising from this driver</i>	<ul style="list-style-type: none"> ▪ <i>Measurable effects applying to the stimulus</i>
Response	<i>The expected reaction of the system to the driver event (black box view putting no constraints on the design)</i>	<ul style="list-style-type: none"> ▪ <i>Measurable effects applying to the response</i> ▪ <i>Measurable indicators that the driver has been achieved by the architecture</i>

Architecture Driver Example

Categorization		Responsibilities	
Driver Name	Application startup time	Supporter	Carla Customer
Driver ID	AD.01.PERFORMANCE	Sponsor	Mike Manager
Status	Realized	Author	Arnold Architect
Priority	High	Inspector	Alfred Architect

Description		Quantification
Environment	The application is installed on the system and has been started before at least once. The application is currently closed and the system is running on normal load.	<ul style="list-style-type: none"> Previous starts ≥ 1
Stimulus	A user starts the application from the Windows start menu.	
Response	The application starts and is ready for inputting search data in less than 1 second. The application is ready for fast answers to search queries after 5 seconds.	<ul style="list-style-type: none"> Initial startup time $< 1s$ Full startup time $< 5s$

Driver Solution Template

Driver Name	<i>Concise short name</i>	
Driver ID	<i>Unique identifier</i>	
Steps	<ol style="list-style-type: none"> <i>1. Logical flow to explain driver solution (white box view explaining the design)</i> <i>2. The glue between design decisions (accepted and discarded)</i> <i>3. Putting all related design decisions in a combined and larger context</i> 	
Related Design Decisions	<p>ACCEPTED</p> <ul style="list-style-type: none"> <i>▪ Link to design decision (detailed description) to enable traceability</i> 	<p>DISCARDED</p> <ul style="list-style-type: none"> <i>▪ Link to design decision (detailed description) to enable traceability</i>
Pros & Opportunities		
<ul style="list-style-type: none"> <i>▪ Points in favor</i> <i>▪ Anticipations of future</i> 		
Cons & Risks		
<ul style="list-style-type: none"> <i>▪ Points against</i> <i>▪ Unknown or open aspects</i> 		
Assumptions & Quantifications		
<ul style="list-style-type: none"> <i>▪ Assumption made about the driver solution (or parts of it)</i> <i>▪ Measurable effects applying to the driver solution (or parts of it)</i> 		
Trade-Offs		
<ul style="list-style-type: none"> <i>▪ Trade-offs to other design decisions, quality attributes, solutions concepts, architecture drivers</i> <i>▪ Potentially impacted if this solution changes</i> 		
Manifestation Links	<i>Links to models, diagrams, additional documentation</i>	

Driver Solution Example

Driver Name	Application startup time
Driver ID	AD.01.PERFORMANCE.

Steps	<ol style="list-style-type: none"> 1. Application always stores preprocessed index-structures on updates of searchable items 2. On startup, loading of search data is moved to a separate thread 3. The UI is started and ready for user input while loading of search data is ongoing 4. After loading the search data, searches can be done without the user noticing that search was not available before
Related Design Decisions	<ul style="list-style-type: none"> ▪ DD.01 Decoupled loading of search data ▪ DD.12 Preprocessed index-structures of search data

Pros & Opportunities
<ul style="list-style-type: none"> ▪ Very fast startup time, application directly usable by user

Cons & Risks
<ul style="list-style-type: none"> ▪ More effort in realization ▪ Loading in separate thread requires synchronization and makes implementation more difficult

Assumptions & Quantifications
<ul style="list-style-type: none"> ▪ Data can be loaded in 5s ▪ User rarely sends a search in less than 4s after start is completed

Trade-Offs
<ul style="list-style-type: none"> ▪ Maintainability, understandability

Decision Rationale Template

Decision Name	<i>Concise short name</i>
Design Decision ID	<i>Unique identifier</i>
Explanation	<i>Explanation of the decision rationale</i>

Pros & Opportunities

- *Points in favor*
- *Anticipations of future*

Cons & Risks

- *Points against*
- *Unknown or open aspects*

Assumptions & Quantifications

- *Assumption made about the driver solution (or parts of it)*
- *Measurable effects applying to the driver solution (or parts of it)*

Trade-Offs

- *Trade-offs to other design decisions, quality attributes, solutions concepts, architecture drivers*
- *Potentially impacted if this solution changes*

Manifestation Links

Links to models, diagrams, additional documentation

Decision Rationale Example

Decision Name	Decoupled loading of search data
Design Decision ID	DD.01
Explanation	Loading the search data is done in a separate thread. The application's UI can be started and used for typing in search queries before the search data is actually loaded.

Pros & Opportunities

- Data loading time does not add on startup time

Cons & Risks

- Loading in separate thread requires synchronization and makes implementation more difficult

Assumptions & Quantifications

- Data can be loaded in 5s

Trade-Offs

- Maintainability, understandability

Manifestation Links

Task



- For each driver specified, *identify* and *describe* the design decisions and rationales to address them.

