Modularization and Software Architectures

UNIVERSITÄT SALZBURG

Software Architecture Analysis Method (SAAM)

UNIVERSITÄT SALZBURG

Selection of relevant criteria in SAAM (I)

- We consider the quality criterion **adaptability** as an example. A software system can be easily adaptable in certain aspects (for example structure and configuration of the user interface), while being hardly adaptable in other aspects (for example in supported data formats and conversions).

- The requirements related to a certain quality criterion must be defined therefore in relation to a certain context.

UNIVERSITÄT SALZBURG

Selection of relevant criteria in SAAM (II)

- SAAM specifies context through **scenarios**. For example, a scenario describes that a user of the software is able to configure the user interface. The user mentioned in the scenario is an example of a concerned party (***stakeholder*** in the SAAM terminology), e.g., a person who has purchased the software system. A **scenario** describes in terms of catchwords the interaction of a stakeholder with the software system.
- Scenarios are the premises of a useful software architecture analysis.

UNIVERSITÄT SALZBURG

SAAM steps

(1) **Identify and assemble stakeholders**

(2) **Develop and prioritize scenarios** (functionality scenarios, change and development scenarios, etc.)

(3) Describe candidate architectures

(4) **Classify scenarios as direct or indirect**

(5) Evaluate indirect scenarios by an analysis of each architecture, to obtain an **estimation of the coupling of the components** of the software architecture

(6) Evaluate the interactions between indirect scenarios by an analysis of the architecture, to obtain an **estimation of the cohesion of the components** of the software architecture

(7) Generate an overall evaluation

UNIVERSITÄT SALZBURG

SAAM – possible stakeholders (I)

Stakeholder	Interest
Customer	Schedule and budget Usefulness of system Meeting market expectations
End user	Functionality Usability Robustness
Developer Maintainer Integrator Application builder	Modularity (Coupling, Cohesion) Documentation System transparency / Readability (e.g., ability to locate places of change)

UNIVERSITÄT SALZBURG

SAAM – possible stakeholders (II)

System administrator	Ease in finding sources of operational problems
Network administrator	Network performance Throughput predictability
Tester	Modularity (Coupling, Cohesion) Consistent error treatment Documentation System transparency / Readability
Representative of application area	Interoperability with other systems

UNIVERSITÄT SALZBURG

Description and prioritization of scenarios

- Scenarios are suggested by stakeholders and must be representative for future requirements, changes and extensions.
- The scenarios should describe all the relevant aspects of using the system and should refer in particular to aspects of functionality, development and change.
- Approx. 10 - 20 scenarios
- At the end: prioritization by importance

UNIVERSITÄT SALZBURG

Description of candidate architectures

- Practice has shown that a simple description of the data connections (which components exchange which information) and the control connections (one component enabling another component to perform its function) is usually sufficient for a static representation of architecture.
- In addition to the static description, runtime behaviors are outlined for example by UML interaction diagrams or by natural language.
- For scenarios that concern changes it may be necessary to include source text fragments.

UNIVERSITÄT SALZBURG

Direct and indirect scenarios

- Direct scenarios are those that can be executed by the system without modification.
- Indirect scenarios are those that require modifications to the system. In this case, it is necessary to indicate also the estimated expenditure for implementing the modifications (in person-days /-months or -years).
- The ensuing SAAM steps regard only the indirect scenarios. An indirect scenario can affect one or more components by the incurred changes of architecture.

UNIVERSITÄT SALZBURG

Evaluation of indirect scenarios

- For each indirect scenario, the number of components affected by changes is indicated.
- This is the measure for coupling in the software architecture.
- Also the cost of performing the changes is estimated.

UNIVERSITÄT SALZBURG

Evaluation of interactions between indirect scenarios

- Two indirect scenarios interact if they require a change of the same component.
- The analysis of scenario interactions points out which components cover several aspects and exhibit thus a small cohesion.
  - High interaction among scenarios that are fundamentally different corresponds to low cohesion and suggests high structural complexity.
  - High interaction among fundamentally similar scenarios signals high cohesion.

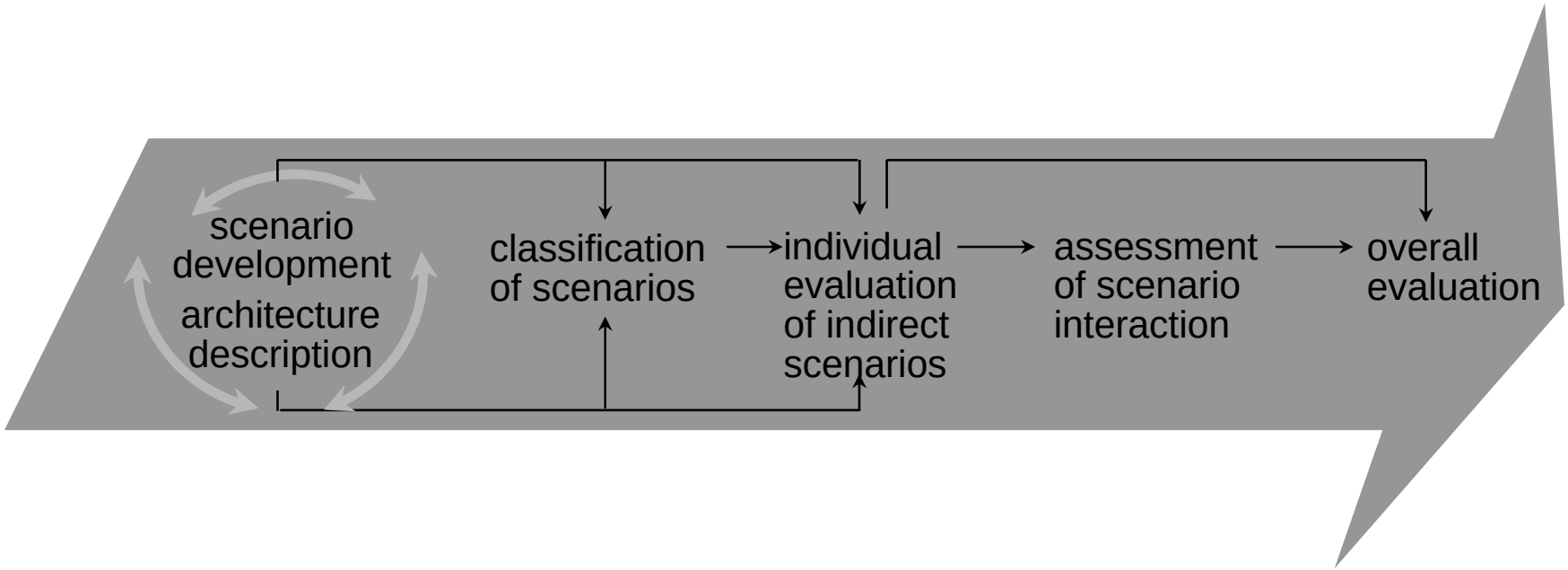
UNIVERSITÄT SALZBURG

Overall Evaluation

- The last step serves in particular to compare the different architecture candidates.
- A weight should be assigned to each scenario and the scenario interactions in terms of their relative importance, and the weighting should be used to determine an overall ranking of the candidate architectures.
- The process of performing a SAAM analysis can be used to gain a more complete understanding of the competing architectures; this understanding, rather than just a scenario-based table, is useful for performing comparative analysis.

UNIVERSITÄT SALZBURG

Interaction of SAAM Steps



UNIVERSITÄT SALZBURG

Example of SAAM application

UNIVERSITÄT SALZBURG

Revision Control System WRCS (I)

- A project in WRCS is a group of related files, which result together in a product if they are connected accordingly:
  - Source text files are those that are translated to an executable program,
  - Text documents of a book or
  - Digitized audio and video data for an advertising spot, etc.

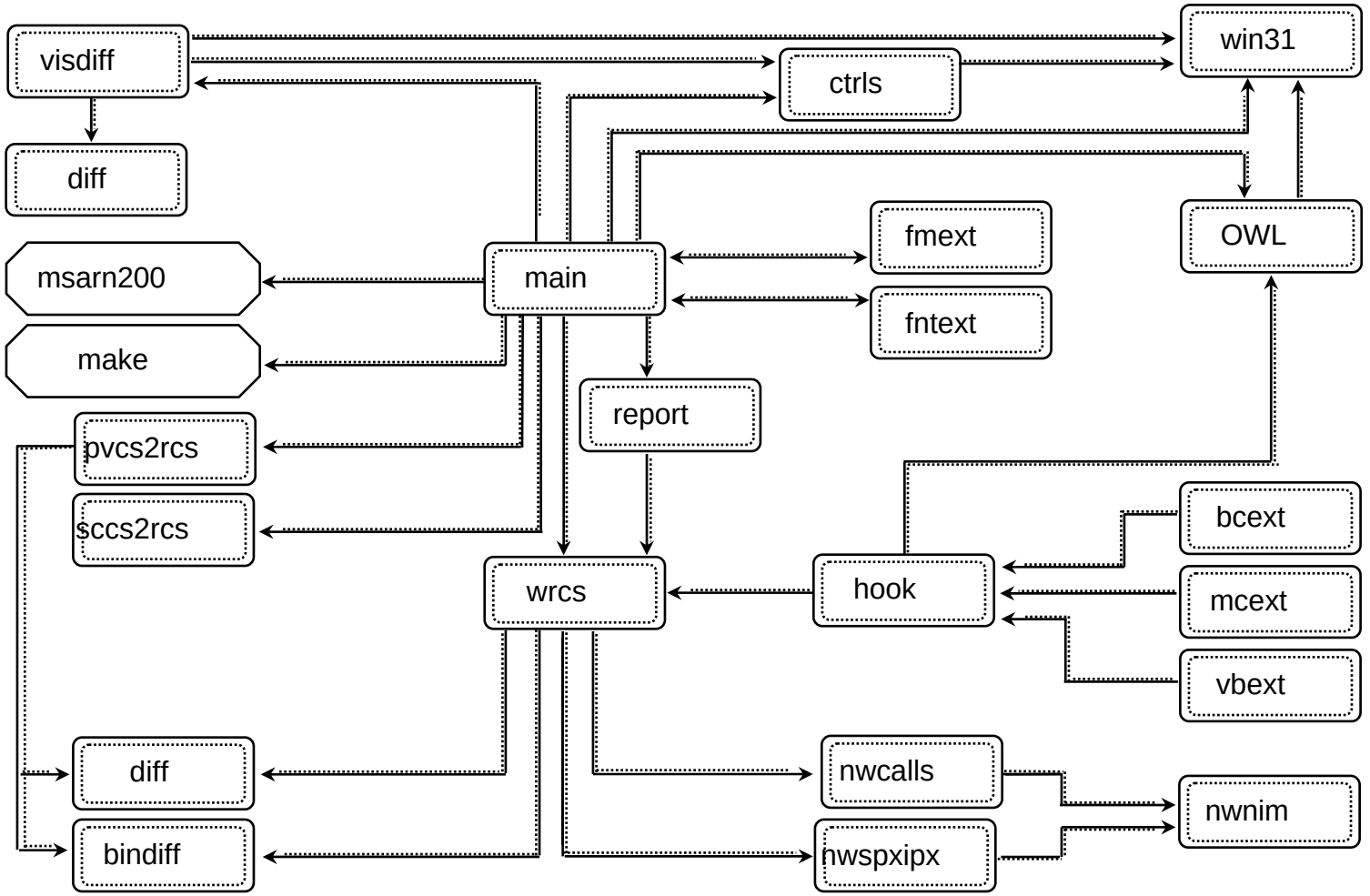
UNIVERSITÄT SALZBURG

Revision Control System WRCS (II)

- With WRCS one can
  - Trace changes of files
  - Define archives
  - Check files in and out
  - Create releases to be produced
  - Restore old versions
- The WRCS has been integrated with different development environments. The available functionality can be implemented by the respective development environments or via the WRCS's graphical user interface.

UNIVERSITÄT SALZBURG

Architectural Representation of WRCS



UNIVERSITÄT SALZBURG

Scenarios and their weights

Stakeholder	Scenario	Importance
End user	compare binary file representations	1
	configure the product's toolbar	3
Developer (Maintainer)	Make minor modifications to the user interface	6
	Port to another operating system	4
Administrator	Change access permissions for a project	5
	Integrate with a new development environment	2

UNIVERSITÄT SALZBURG

Classification in direct and indirect scenarios

Stakeholder	Scenario	Classification
End user	compare binary file representations	indirect
	configure the product's toolbar	direct
Developer (Maintainer)	Make minor modifications to the user interface	indirect
	Port to another operating system	indirect
Administrator	Change access permissions for a project	direct
	Integrate with a new development environment	indirect

UNIVERSITÄT SALZBURG

Number of components that must be changed by an indirect scenario

Indirect scenario	Number of components to change
Compare binary file representations	2
Configure the product's toolbar	3
Port to another operating system	3+
Integrate with a new development environment	4

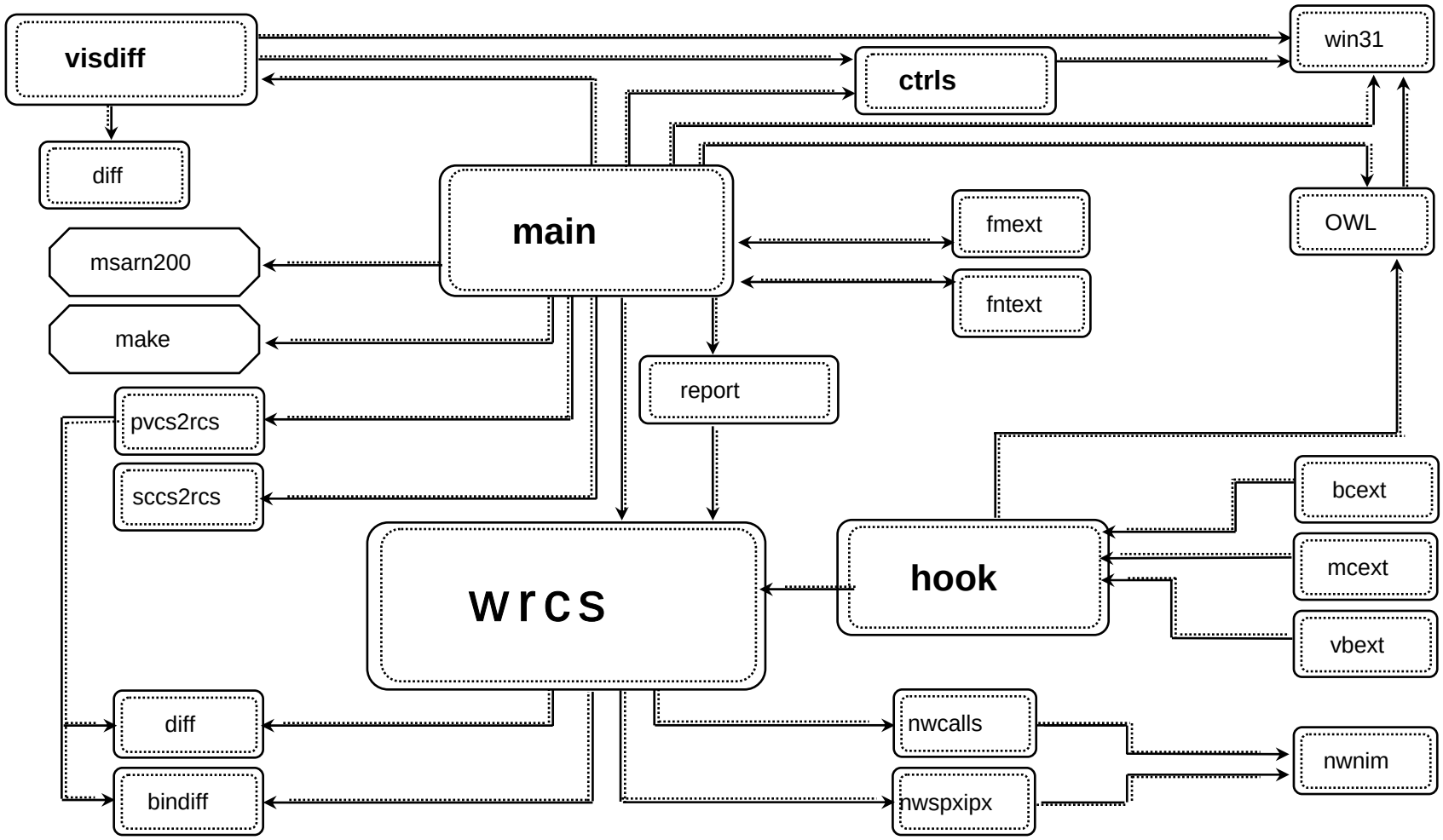
UNIVERSITÄT SALZBURG

Evaluation of interactions between indirect scenarios

Modul	Number of changes
main	4
wrcs	7
diff	1
bindiff	1
pvcs2rcs	1
sccs2rcs	1
nwcalls	1
nwspxipx	1
nwnlm	1
hook	4
report	1
visdiff	3
ctrls	2

UNIVERSITÄT SALZBURG

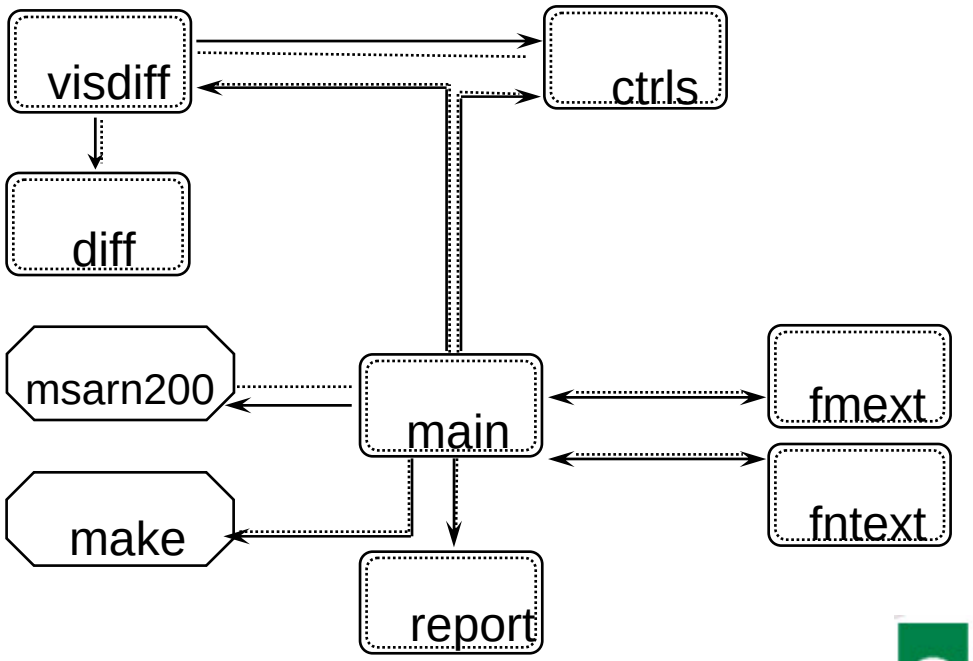
Visualization of scenario interactions



UNIVERSITÄT SALZBURG

Advantages and risks of using SAAM (I)

- The usefulness of the results depends crucially on the correct choice of the granularity of architecture description. For example, the following granularity would be problematic with WRCS:



UNIVERSITÄT SALZBURG

Advantages and risks of using SAAM (II)

- Makes possible rapid and goal-safe evaluation of the quality of a software architecture, in particular regarding changes and extensions.
- No detailed code inspections are necessary.
- An expenditure of few person days (2 to 5 days depending upon system complexity) is necessary.
- Experience and technical knowledge of the stakeholders are conditions for a successful SAAM application.
- Stakeholder participation is the socio-economic component of the SAAM and secures the efficiency of the analysis process.
- The SAAM offers a pragmatic possibility to “measure” coupling and cohesion. Based on these measurements, one can make an evaluation of the balance between coupling and cohesion of a selected modularity.

UNIVERSITÄT SALZBURG