Software architecture reconstruction
Krikhaar, R.

Citation for published version (APA):
Krikhaar, R. (1999). Software architecture reconstruction

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Glossary

4 + 1 View Model:
A specific architecture view model that partitions an architecture into 4 views (logical view, development view, process view, physical view) plus an additional view (scenarios) that combines the four views (see Section 1.2.1).

abstraction:
An activity that raises extracted information to a higher level of abstraction, e.g. to an architectural level (see Section 2.3).

ArchiSpect:
A concept of the SAR method that describes how an aspect of architecture of an existing system can be reconstructed (see Section 2.5.2).

architectural pattern:
A recurring solution to a problem relating to architecture (see Section 1.5).

architecture:
The main structures of a system, also used as a shorthand for software architecture (see Chapter 1).

architecture conformance:
The situation in which the implementation of a system conforms to the architecture (see Section 6.1).

architecture improvement:
The process of improving the architecture of an existing system (see Section 2.4).

architecture verification:
The process of verifying an implementation by comparing it with its architecture to assess architecture conformance (see Section 6.1).
AV model:
An architectural view model that partitions an architecture into five different views (logical view, module view, code view, execution view, physical view) plus an extra view (scenarios) that combines the five views (see Section 1.2.3).

binary relation:
A set of tuples (\{..., (x, y), ...\}) representing a certain relation, e.g. calls for function calls within a system (see Section 3.3).

Building Block method:
A dedicated software architecture method that stems from telecommunication system development (see Section 2.2).

component:
A generic name for a piece of software; this term is sometimes used to refer to a piece of software at a certain level of decomposition.

decomposition hierarchy:
The hierarchy of software entities of a system including their containment relationship (see Section 4.7).

decomposition level:
A certain level in the decomposition hierarchy (see Section 4.7).

extraction:
An activity involving the retrieval of information from source code, design documentation and/or domain experts (see Section 2.3).

Files:
An example of the notation used in this thesis for sets (see Section 3.6.2).

forward architecting:
The discipline of creating new architectures for software systems (see Section 2.2).

impact analysis:
The process of simulating a possible idea for modification in a software model to analyse, in advance, the possible consequences of its application to actual source code (see Section 5.1).

importsFiles, Files:
An example of the notation used in this thesis for multi-relations: importsFiles, Files represents the multi-relation: imports ⊆ Files × Files (see Section 3.6.2).
An example of the notation used in this thesis for binary relations: \( \text{imports} \ Files, Files \) represents the binary relation: \( \text{imports} \subseteq Files \times Files \) (see Section 3.6.2).

InfoPack:
A concept of the SAR method that describes how to extract (architecture-relevant) information from existing software (see Section 2.5.2).

lifting:
An RPA operation involving a relation and a part-of relation resulting in a relation at a higher level of abstraction (see Section 3.4).

lowering:
An RPA operation involving a relation and a part-of relation resulting in a relation having a finer coarse of granularity (see Section 3.4).

multi-relation:
A bag of tuples \((a, b)\), represented as a set of triples \((a, b, n)\), where \(n\) represents the number of occurrences, called the weight (see Section 3.5).

multi-set:
A bag of entities, represented as a set of tuples \((a, n)\), where \(n\) represents the number of occurrences, called the weight (see Section 3.5).

part-of relation:
A relation that describes a partition, i.e. a division of a set of entities into various non-overlapping (named) parts (see Section 3.4).

presentation:
The activity of showing (architectural) information to developers and architects in an appropriate way e.g. by means of diagrams, tables and/or text (see Section 2.3).

re-architecting:
The process of modifying the software architecture of an existing system (see Section 2.3).

repository:
A data-store containing software-related information (see Section 2.3).

reverse architecting:
Reverse engineering of software architectures (see Section 2.3).
reverse engineering:
The process of analysing a subject system to identify the system’s components and their relationships and create representations of the system in another form or at a higher level of abstraction (see Section 2.3).

RPA:
Relation Partition Algebra, an algebra based upon sets, relations and partitions (see Chapter 3).

SAR:
Software Architecture Reconstruction, a method for reconstructing software architectures (see Chapter 2).

SAR level:
A level of the Software Architecture Reconstruction method (see Section 2.5.1).

set:
A collection of objects, called elements or members (see Section 3.2).

SNH model:
An architectural view model that partitions an architecture into five different views: conceptual architecture, module interconnection architecture, execution architecture, code architecture and hardware architecture (see Section 1.2.2).

software architecture:
A heavily overloaded term, which covers at least the main structures of a software system (see Section 1.2).

software architecture reconstruction:
The process of recovering an existing software architecture, improving an existing software architecture and/or verifying the architecture of an existing system (see Section 2.5).

software architecting:
The process of creating software architectures (see Section 1.2).