Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

UNIVERSITY OF APPLIED SCIENCES UTRECHT

Software Architecture Education

A Practical Approach with HUSACCT

Leo Pruijt, HU University of Applied Science, Utrecht

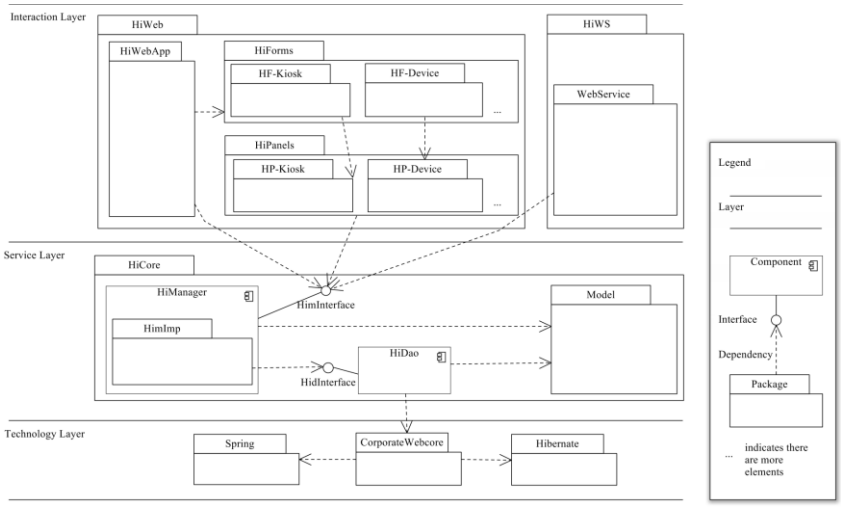
Agenda

- Software Architecture
- SA Education
- Architecture Reconstruction with HUSACCT
- Architecture Compliance Checking with HUSACCT
- Questions

Software Architecture (SA)

- Software architecture is of major importance to achieve
  - the business goals
  - functional and non-functional requirements
- However …
  - Architecture models are often obsolete or incomplete
  - Architectural models tend to be of a high-level of abstraction
    - Hard to understand by students and practitioners
    - Deviations of the software architecture arise easily during the development and evolution of a system

SA Example (Schiphol Group)



Module View

Five types of modules are visible in the example
- Subsystems, Layers, Components, Interfaces, External Systems

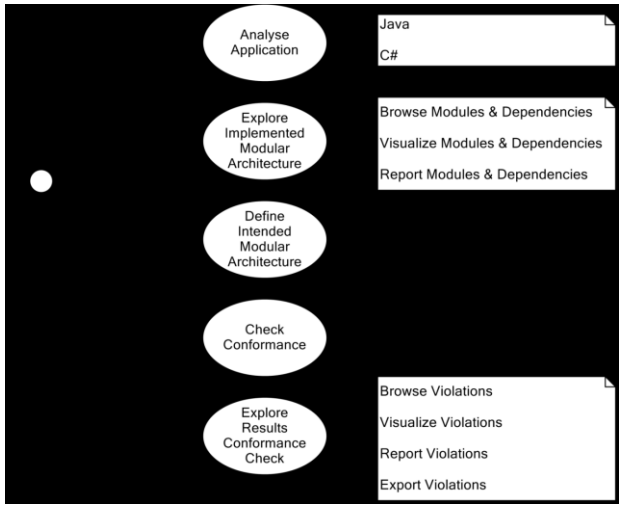
Rules of different types are used in the example
- Some of them specific for a certain type of module
- E.g., 'Back call ban' and 'Skip call ban' are specific for Layers

Type of Rule	Example (E)
Is not allowed to use	HiPanels is not allowed to use HiWS.
Back call ban	Service Layer is not allowed to use the Interaction Layer.
Facade convention	Component HiManager may be accessed only via HimInterface.
Is only allowed to use	HiForms is only allowed to use HiPanels.
Is the only module allowed to use	CorporateWebcore is the only module allowed to use Hibernate.

SA Education at the HU

- Base SA on foundation of Analysis & Design
  - Preceding courses (in year 1 and 2) on
    - Business processes, Use cases
    - Domain model, Object interaction, Database design
- Relate architecture models to code
  - Architecture Reconstruction with HUSACCT
  - Architecture Compliance Checking with HUSACCT
  - In year 2 (basic) and 3 (advanced)
- Apply in student projects
- Apply on practical cases (open source, visit companies)

HUSACCT: Hogeschool Utrecht
Software Architecture Compliance Checking Tool

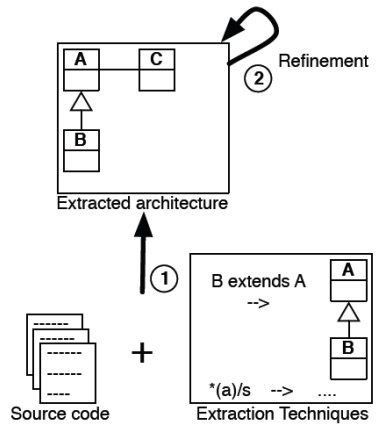


Outstanding Characteristics

- HUSACCT is free-to-use & open source
  - Download, video & manual at http://husacct.github.io/HUSACCT/
- Support of rich sets of Module and Rule Types
  - 5 common Module Types with different semantics
    - Subsystem, Layer, Component, Interface, External system
  - 11 common Rule Types
- Extensive Semantic Support of the Module and Rule Types, e.g.:
  - Automatic creation of default rules, according to the Module Type
  - Type of Module determines which Rule Types are selectable
- Configurable support
  - Enable/Disable rules, Exception rules, Default rule configuration

Architecture Reconstruction

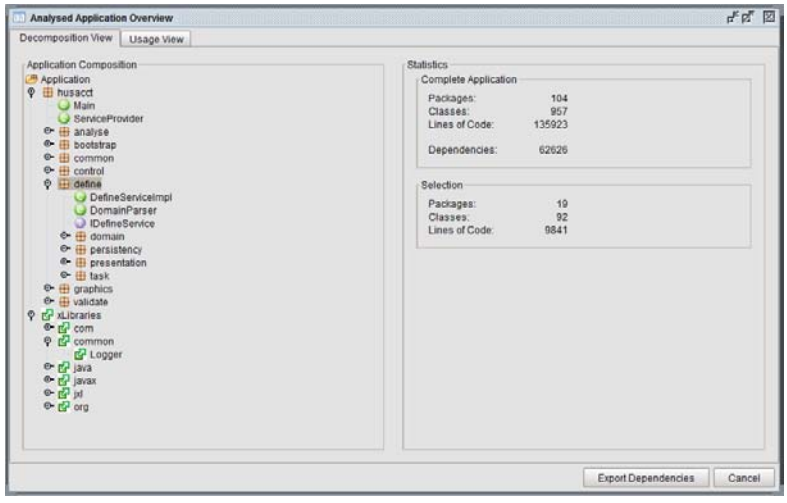
Architecture reconstruction: *comprehending* and *documenting* an architecture based on the source code (and other information)



Source: Ducasse & Polle (2009)

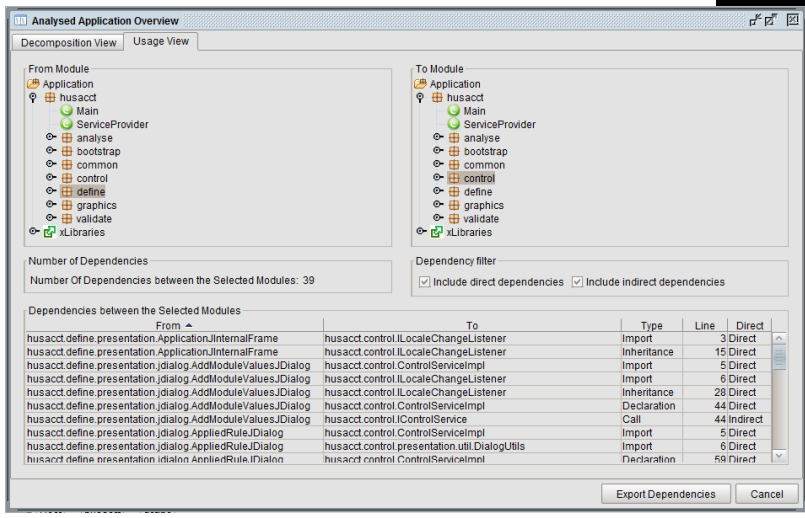
Analyse Implemented Architecture

Decomposition View: How is the software structured?



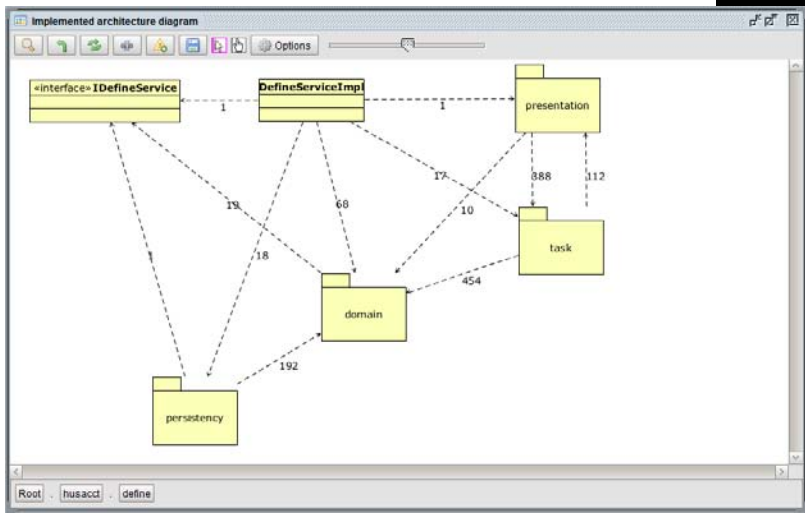
Analyse Implemented Architecture

Usage View: Which dependencies are present?



Analyse Implemented Architecture

Implemented Architecture Diagram

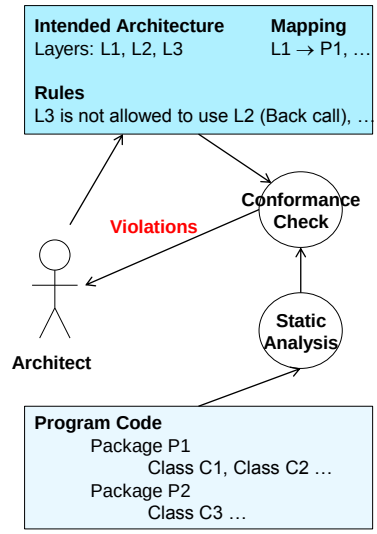


Architecture Compliance Checking (ACC)

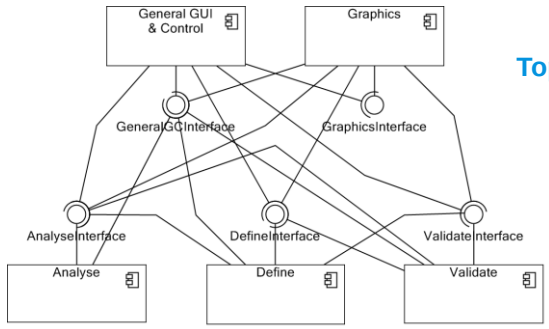
ACC verifies the conformance of
- Implemented program code to
- High-level models of architectural design

Currently …
- Adoption of ACC in practice is limited
- Tool support of the common sets of module & rule types is limited

Research Goals …
- Improve tool support
- Promote ACC in practice & education



ACC Example: Modular Architecture HUSACCT_1.0

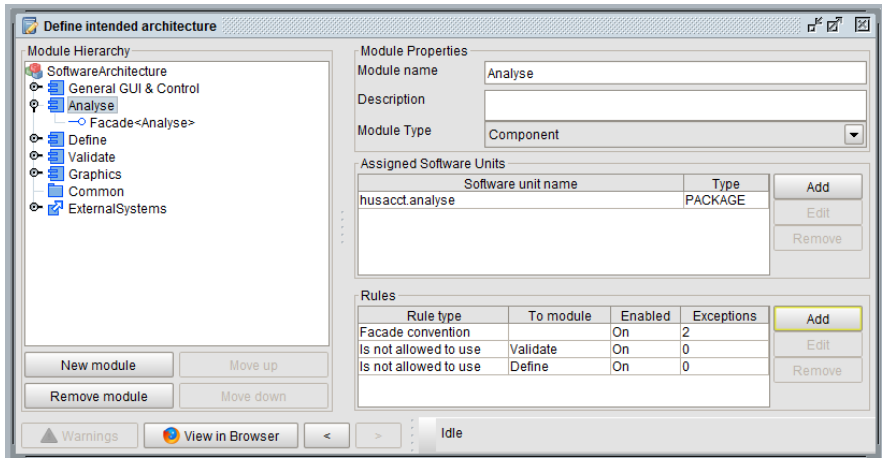


Top-level Components

Rules

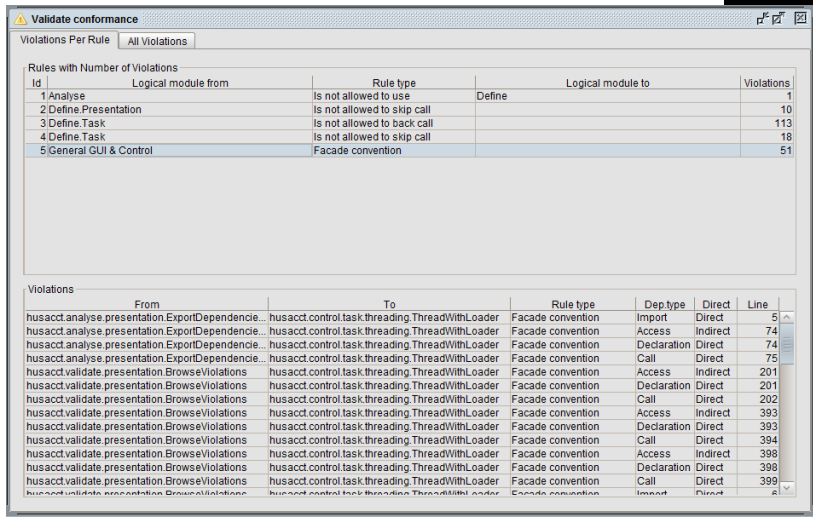
From-Module	Constraint	To-Module
Analyse	is not allowed to use	Define
Analyse	is not allowed to use	Validate
General GUI & Control	Is the only module allowed to use	Graphics
All five components	Facade convention	

Define Intended Architecture



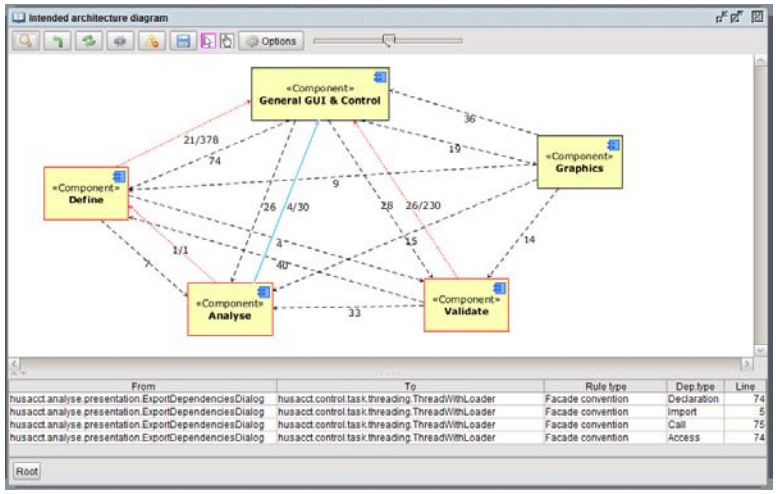
Validate Conformance

Results of the Conformance Check

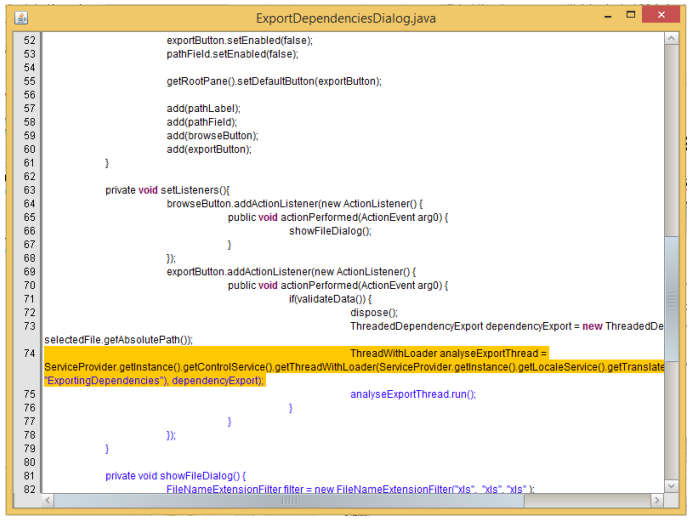


Intended Architecture Diagram

Top-level components, with dependencies and violations

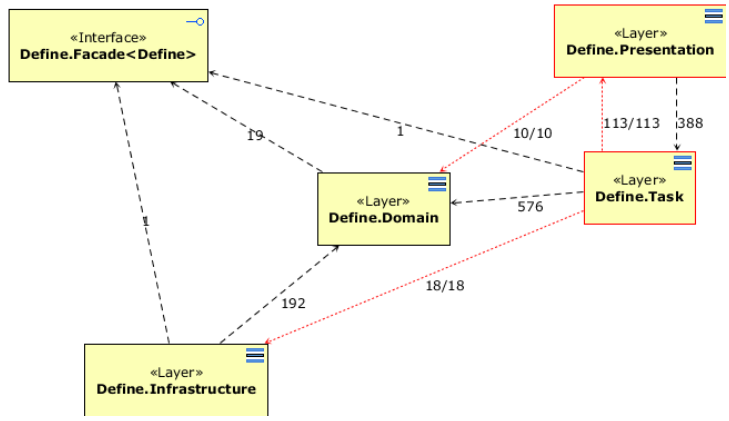


Code Viewer Double-click on dependency



Intended Architecture Diagram

Layers within Define, with dependencies and violations



Finally

- More information:
  - Ask me: Leo Pruijt, leo.pruijt@hu.nl
  - Watch the video at http://husacct.github.io/HUSACCT/
  - Read the published papers
  - Contact me for course materials
- Thank you for your attention!
- Questions?

1) HUSACCT: Architecture Compliance Checking with Rich Sets of Module and Rule Types. *2014 IEEE/ACM Int. Conf. on Automated Software Engineering, ASE 2014-09, Vasteras, Sweden*

2) Architecture Compliance Checking of Semantically Rich Modular Architectures: A Comparative Study of Tool Support. *Int. Conf. on Software Maintenance, ICSM 2013-09, Eindhoven, NL*

3) A Metamodel for the Support of Semantically Rich Modular Architectures in the Context of Architecture Compliance Checking. *SAEroCon workshop, WICSA 2014-04, Sydney, Austratlia*

4) On the Accuracy of Architecture Compliance Checking Support: Accuracy of Dependency Analysis and Violation Reporting. *Int. Conf. on Program Comprehension, ICPC 2013-05, San Francisco, USA*