ConQAT

Ein Toolkit zur kontinuierlichen Qualitätsanalyse

Proseminar IT Kennzahlen und Softwaremetriken

22.07.2010

Alexander Ried

Gliederung

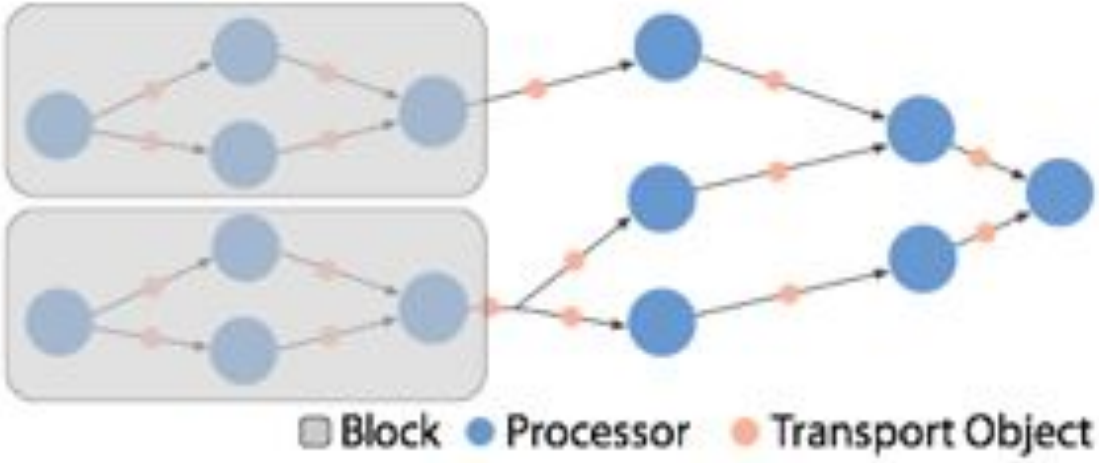
- Motivation
- Grundlagen
- Designüberblick
- Architecture Conformance Analysis
- Clone Detection
- Trends und rückwirkende Datenerhebung
- Ausblick: Integration in continuous build system
- Quellen

Motivation

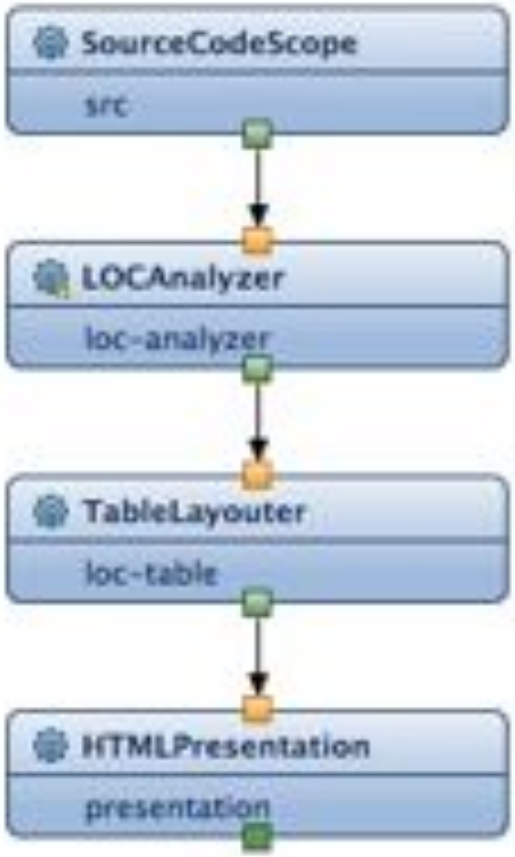
- Software-Qualität hat entscheidenden Einfluss auf Produktivität von Wartung und Weiterentwicklung
- Kontinuierliche Überwachung von Qualitätsmerkmalen wirkt dem Verfall entgegen
- Qualitätskriterien ändern sich im Projektverlauf, Überwachung muss flexibel sein
- Autonomer Betrieb
- Aggregation
- Flexibilität
- Antwort auf konkrete Fragestellung, keine vollständige Analyse

Grundlagen

- Filter (Processor)
  - Verarbeitet Daten
  - Pro Verarbeitungsschritt ein Filter
  - Kombination mehrerer Prozessoren zu *Blöcken*
- Pipe (Transport Object)
  - Datenübertragung zwischen Filtern
- Driver
  - Prozessornetzwerk initiieren



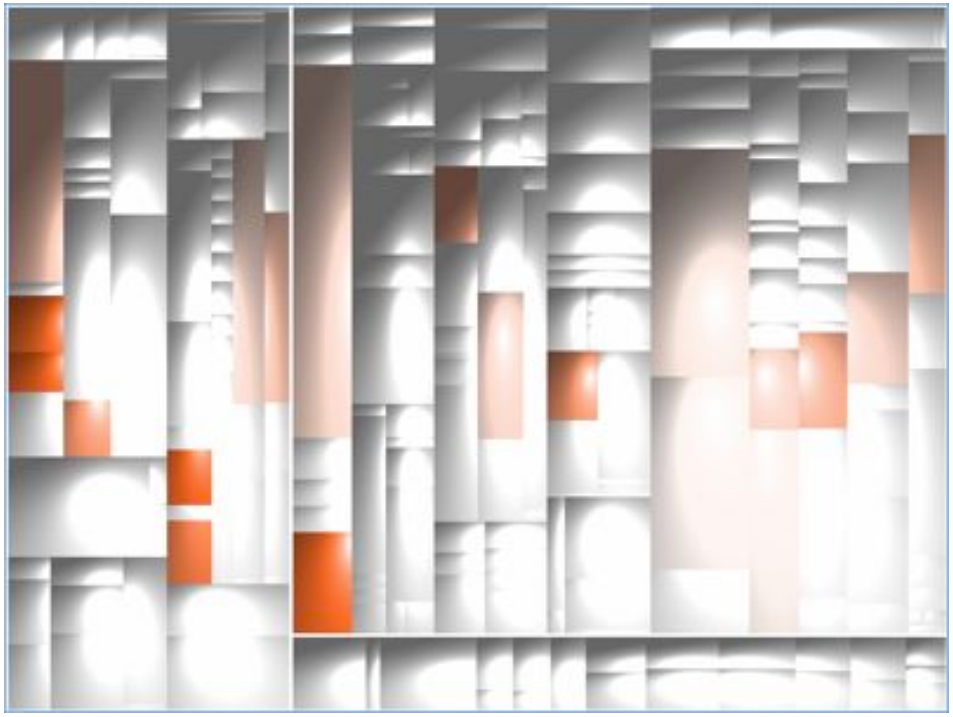
ConCAT-Analyse als Graph [DE10]



ConCAT-Beispielanalyse [Co10]

Grundlagen (2)

- Artefakt: Repräsentation von beliebiger Information
- Tree map
  - Darstellung von Baumstrukturen als geschachtelte Rechtecke
  - Gewichtung über Farben



Tree map-Beispiel für Codeclones [Co10]

Designüberblick

Modularer Aufbau:

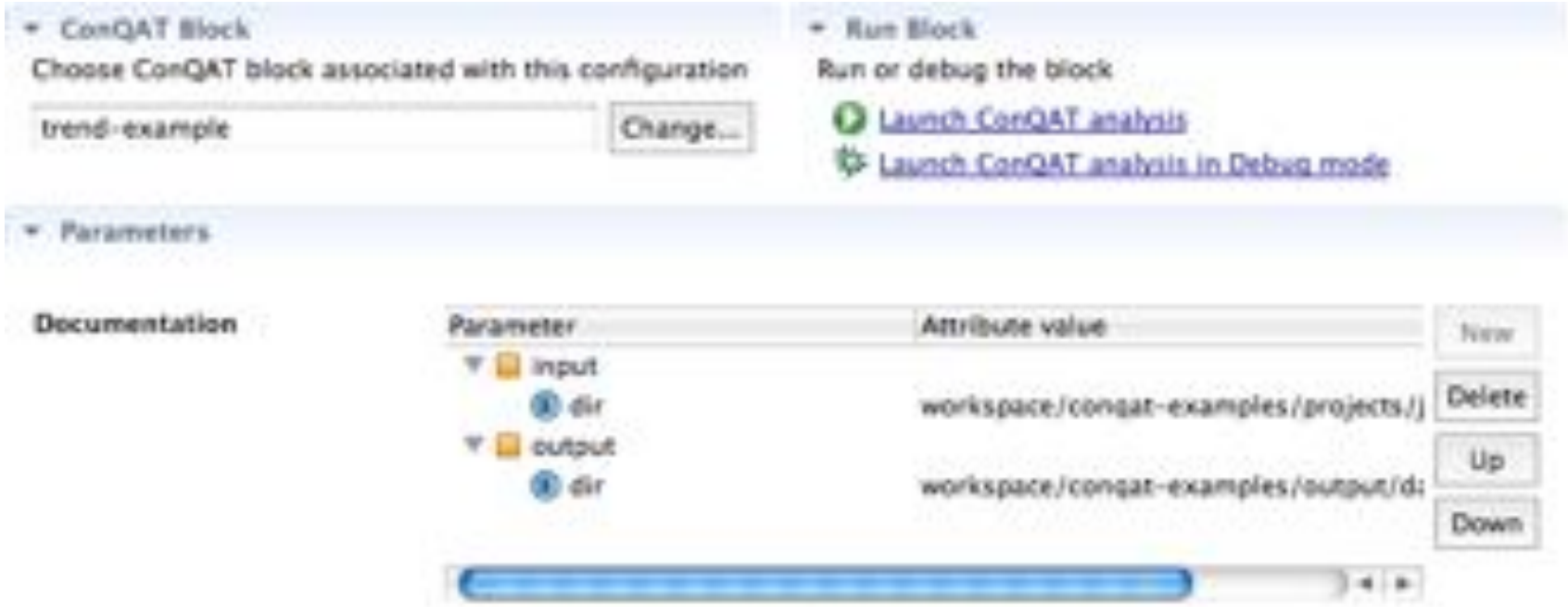
- Core-Komponente (Driver, 1.5MB)
  - Type and multiplicity inference
  - Bundles initialisieren
- Bundles stellen Funktionalität bereit (Filter, zzt. 3.8MB):
  - edu.tum.cs.conqat.architecture
  - edu.tum.cs.conqat.bugzilla
  - edu.tum.cs.conqat.clonedetective
  - edu.tum.cs.conqat.database
  - edu.tum.cs.conqat.filesystem
  - edu.tum.cs.conqat.java

Initialisierung der Bundles [DE10]

Designüberblick (2)

ConQAT-Eclipse-Plugins

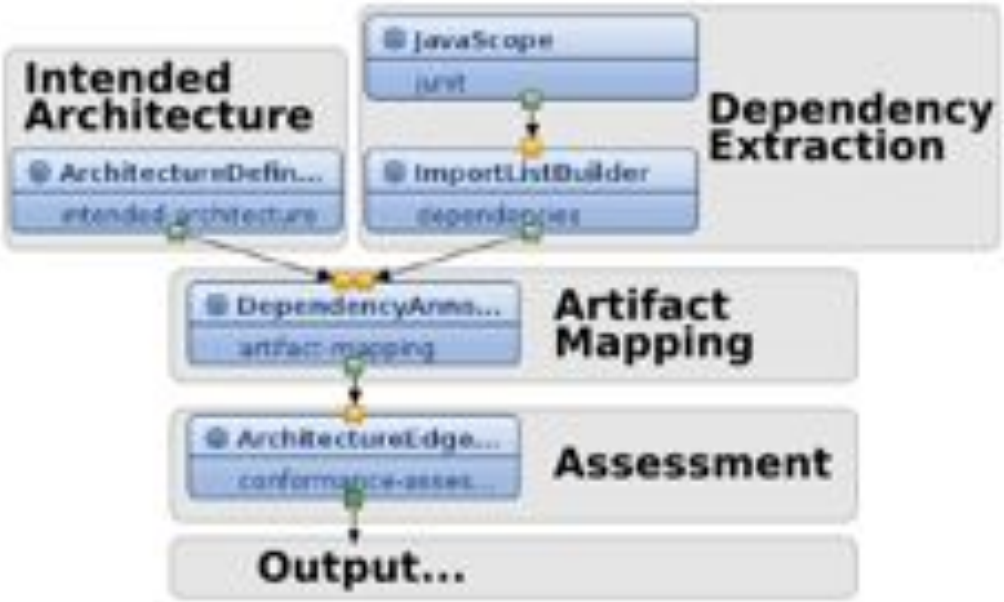
- ConQAT Block Editor
- ConQAT Run Config (Run und Debug)
- ConQAT Library View
- ConQAT Clone Detection



ConQAT Run Config-Beispiel [Co10]

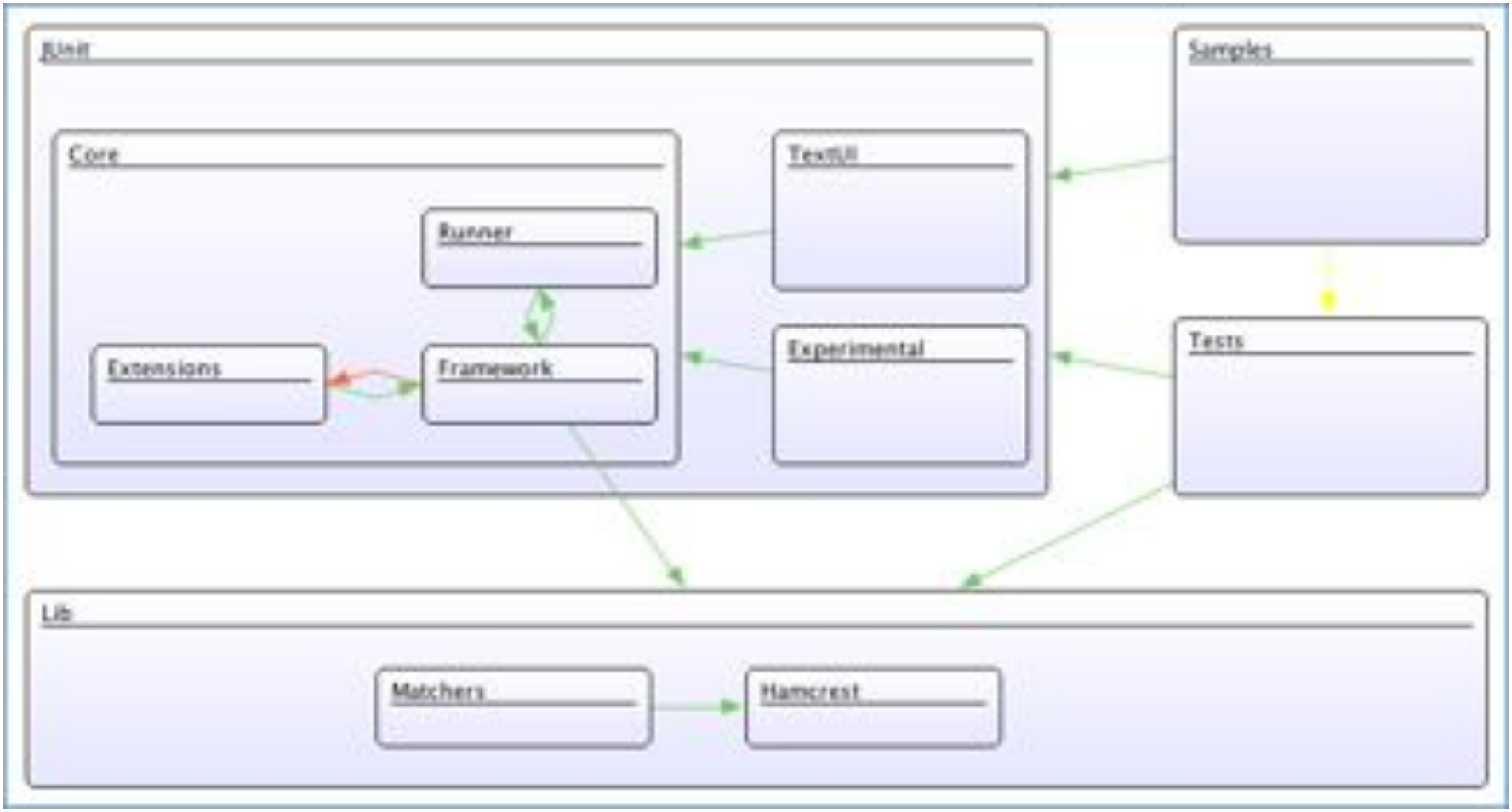
Architecture Conformance Analysis

- Architektur beschreibt
  - Einzelne Komponenten des Systems
  - Komponentenzugehörigkeit von Klassen
  - Abhängigkeiten unter den Komponenten
- Erhöht Wartbarkeit und Einstiegsfreundlichkeit
- Ohne Gegenmaßnahmen Verfall der Architektur



Beispiel Architekturanalyse-Block [DE10]

Architecture Conformance Analysis (2)



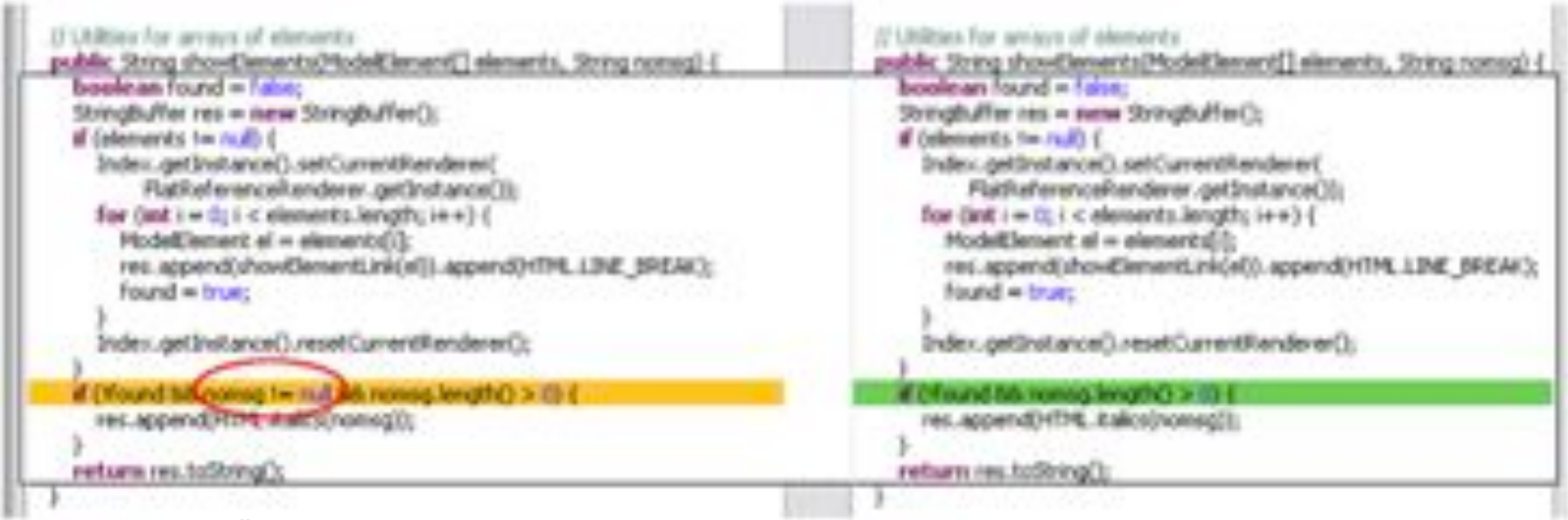
Beispielauswertung Junit [Co10]

Clone Detection

- Clones sind aus zwei Gründen schädlich: [Ju09]
  - Erhöhte Wartungskosten
  - Inkonsistente Änderungen können unerwartete Bugs erzeugen
- Messbarer Einfluss:
  - Clones divergieren
  - Änderungen an mehreren Stellen
- Duplikate müssen nicht zeichenweise übereinstimmen: Kommentare und unterschiedliche Variablennamen haben keinen Einfluss auf Logik
- Anzahl der Änderungen zwischen zwei Codestücken muss unter Grenzwert liegen (inconsistent clones)

Clone Detection (2)

- Text clone detection (Wort oder Zeile)
  - Findet nur identlische Blöcke
- Source code clone detection
  - Erfordert Kenntnis der Sprachsyntax
  - Code wird normalisiert und dann verglichen
  - Filtern von gewollten Clones nach Hashwerten über Blacklist



Inkonsistente Änderung an Clones [De10]

Trends und rückwirkende Datenerhebung

- Manche Kenngrößen sollen als *Trends* analysiert werden
- ConQAT ermöglicht Anbindung an Datenbank zum persistieren und laden von früheren Daten
- Wird eine neue Analyse hinzugefügt kann sie automatisch auf Snapshots aus dem Quell-Repository angewandt werden, um Daten zu sammeln

Ausblick: Integration in continuous build system

- Um große Software automatisch zu analysieren bedient man sich häufig eines continuous build systems
- Regelmäßiger kompletter Build-Prozess:
  - Zeitnahes Feedback für Entwickler
  - Stets eine aktuelle, gebaute Version verfügbar
- Integration von ConQAT-Analyse über Apache Ant
- Beispiele bieten guten Einstieg in Funktionalität
- Unterschiedliche Programmierumgebungen können zu Problemen führen

Quellen

- *[Co10] ConCAT Source (inkl. Samples)*
- *[De10] Dießenböck F., Feilkas M., Heinemann L., Hummel B., Jürgens E.: ConQAT Book, 2010*
- *[DH10] Dießenboeck F., Heinemann L., Hummel B., Juergens E.: Flexible Architecture Conformance Assessment with ConQAT, 2010*
- *[Ju09] Elmar Juergens, Florian Deissenboeck, Benjamin Hummel, Stefan Wagner: Do Code Clones Matter?, ISACA 2009*
- http://de.wikipedia.org/wiki/Pipes_und_Filter, 20.07.2010
- http://www.cs.umd.edu/hcil/treemap-history/, 20.07.2010