Werkzeugunabhängigkeit bei der Modellierung

Schwierigkeiten und mögliche Lösungsansätze

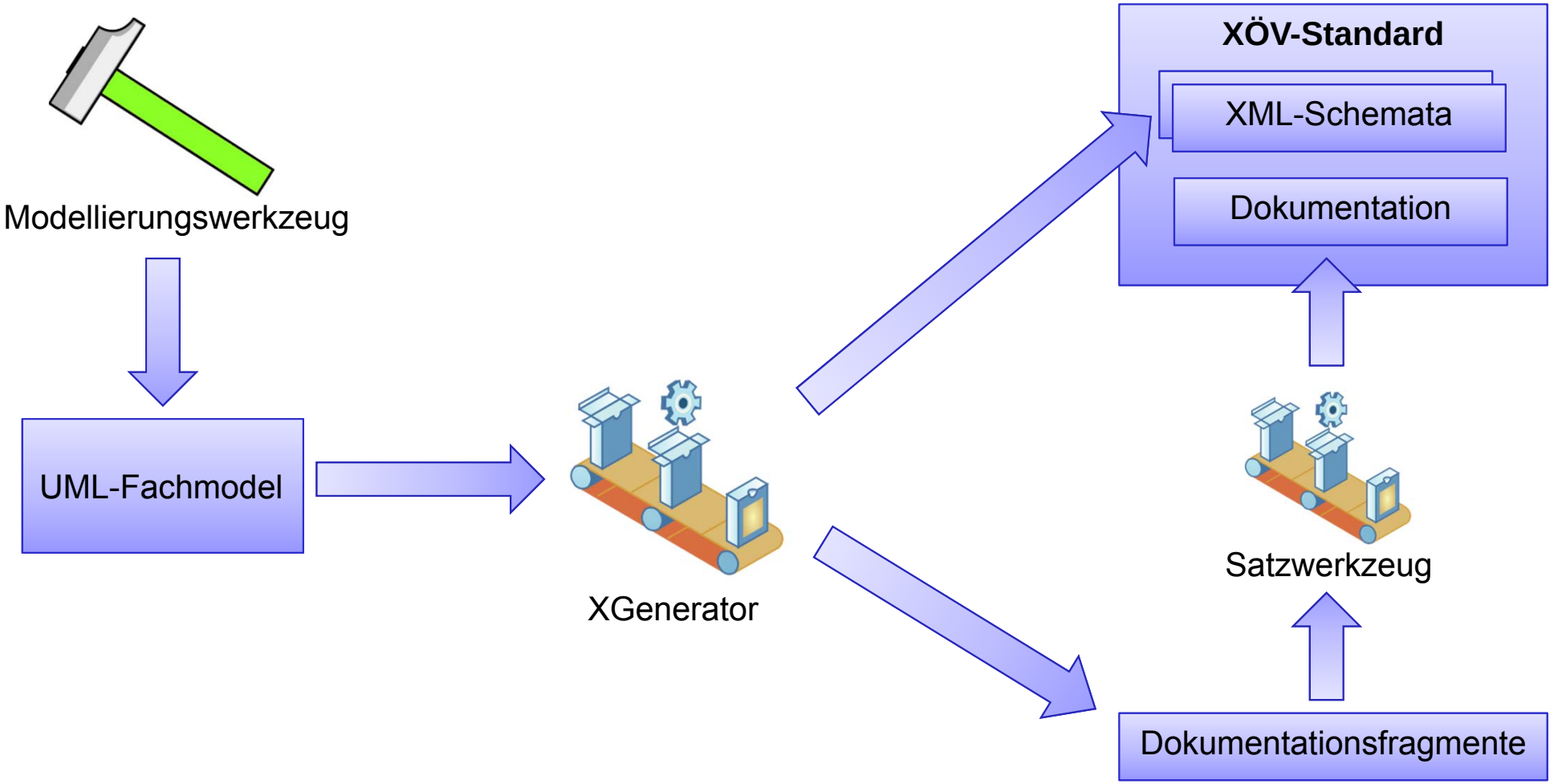
Oliver Hofrichter (hofrichter@tzi.de)
Lars Hamann (lhamann@tzi.de)

Universität Bremen
Technologie-Zentrum
Informatik und Informationstechnik

Überblick

- Motivation
  - Kontext
  - Warum Werkzeugunabhängigkeit
  - Spannungsfeld
- Problemfelder
  - Modellaustausch
  - Zusammenarbeit
- Wege zur Werkzeugunabhängigkeit
- Fazit

Kontext


Modellierungswerkzeug
UML-Fachmodel
XGenerator
XÖV-Standard
XML-Schemata
Dokumentation
Satzwerkzeug
Dokumentationsfragmente

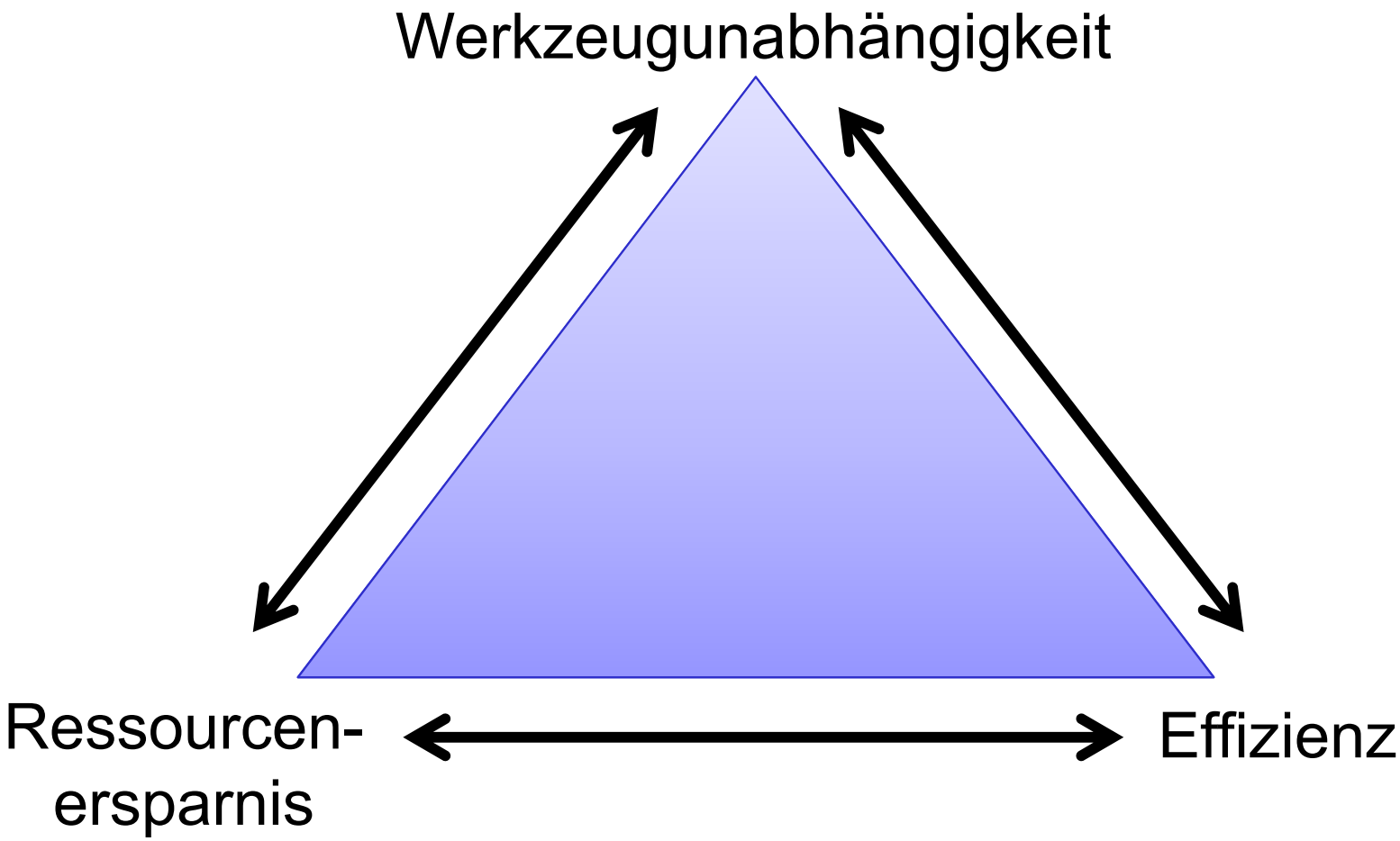
Warum Werkzeugunabhängigkeit?

- Freier Markt
  - Verwendung des geeignetsten Produkts
  - Innovationen
  - Kosten / Nutzen
- Vorhandene Werkzeuge nutzen
  - Keine neuen Lizenzkosten
  - Keine Schulungs-/Einarbeitungskosten

Ist-Zustand der Produktionsumgebung

- Stabil
  - Zusammenspiel erprobt
  - Werkzeuge sind aufeinander abgestimmt
- Effizient
  - Erfahrung vorhanden (Support)
  - Nützliche Anpassungen/Erweiterungen
- Andere Werkzeuge könnten funktionieren
  - Keine Aussage durch die KoSIT möglich (Ressourcen)

Spannungsfeld

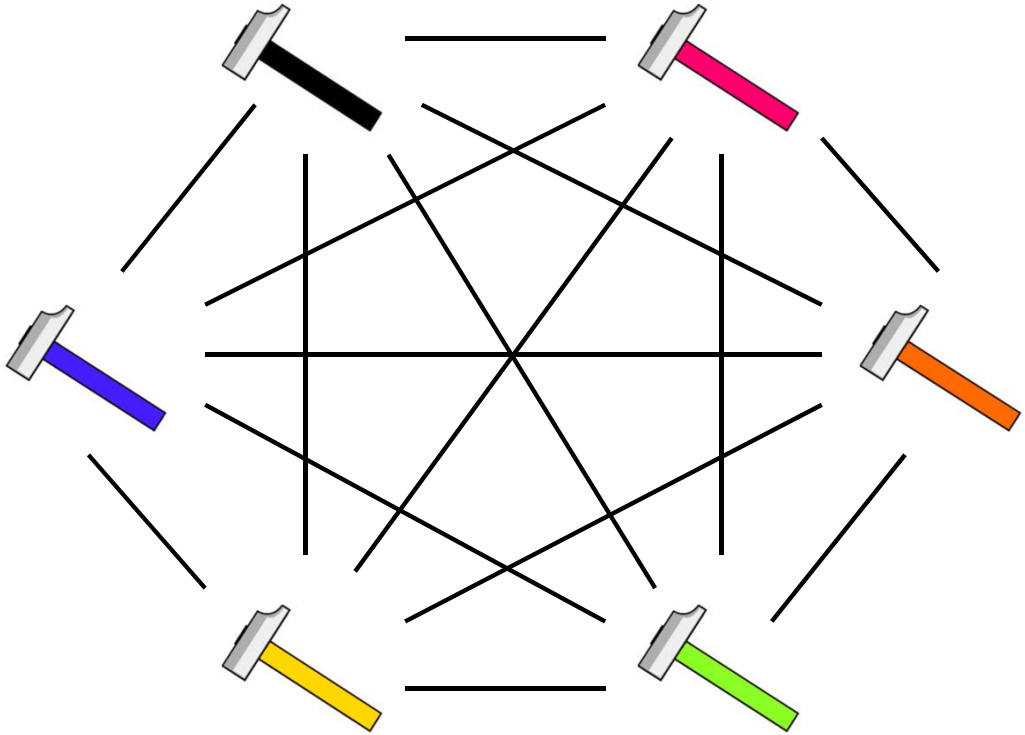


Überblick

- Motivation
  - Kontext
  - Warum Werkzeugunabhängigkeit
  - Spannungsfeld
- Problemfelder
  - Modellaustausch
  - Zusammenarbeit
- Wege zur Werkzeugunabhängigkeit
- Fazit

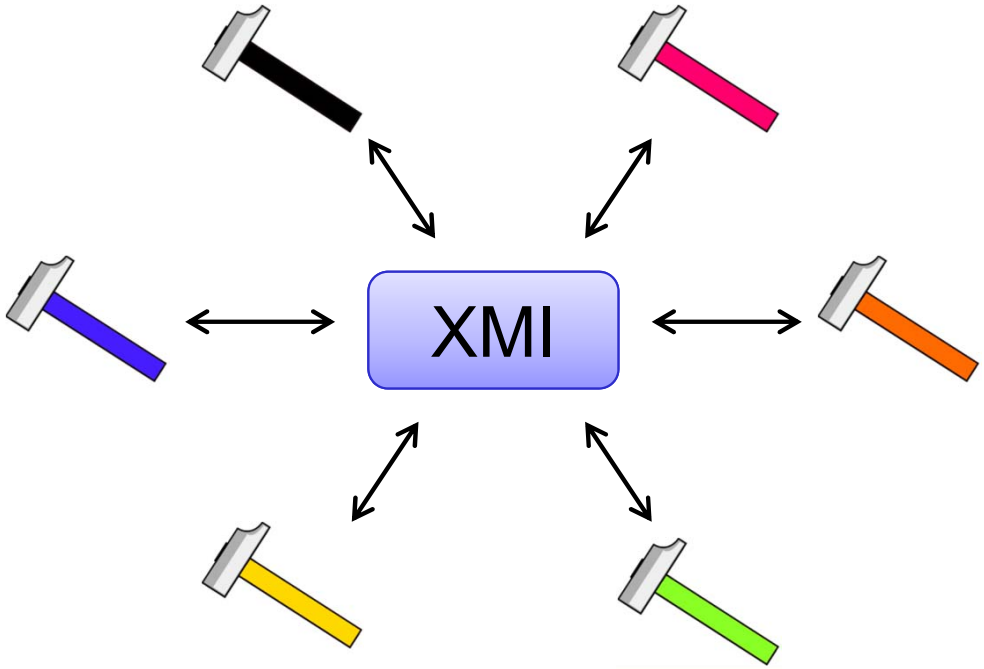
Modellaustausch

- Jedes Werkzeug verwendet „sein“ Format



Modellaustausch

- Statt „Anzahl der relevanten Werkzeuge“ mal eine Import- und Exportfunktion bereitzustellen:
  - Ein gemeinsames Format (XMI)



XMI – XML Metadata Interchange

- Standard der OMG (Object Management Group)
- Austausch von MOF-basierten Modellen
  - Nicht nur UML
- Basis: XML
- Version 1.0: Juni 2000
- Aktuelle Version: 2.4.1 (2011)

XMI – XML Metadata Interchange

- Layoutinformationen werden nicht behandelt
  - Ausschließlich Strukturinformationen / Elementdaten
  - Werkzeuge speichern die Informationen unterschiedlich
- „Lösung“ hierfür: Diagram Definition
  - OMG Spezifikation zum Austausch von Diagramminformationen
  - Version 1.0 vom Juli 2012
  - Vorgänger: Diagram Interchange Spezifikation (2006)

Probleme von XMI

- XMI mehrdeutig und unvollständig
  - Beispiel: UML-Attribute → XML-Element oder -Attribut
- XMI fehlerhaft / inkonsistent
  - Defaultwert von Multiplizitäten
- XMI unterspezifiziert / unvollständig
- Mögliche Folgen
  - Kein Import möglich
  - Modell wird anders interprätiert
  - Informationsverlust

Weitere Probleme

- Werkzeuge wollen sich von anderen abheben
  - Z. B. durch zusätzliche Funktionen / Erweiterungen
  - XMI bietet einen Container für Erweiterungen
    - Wird nicht von allen genutzt was zu Importfehlern führt
    - Informationsverlust beim Import
- Neben den technischen Problemen existiert ein generelles Problem mit XMI
  - Dazu ein kleiner Exkurs

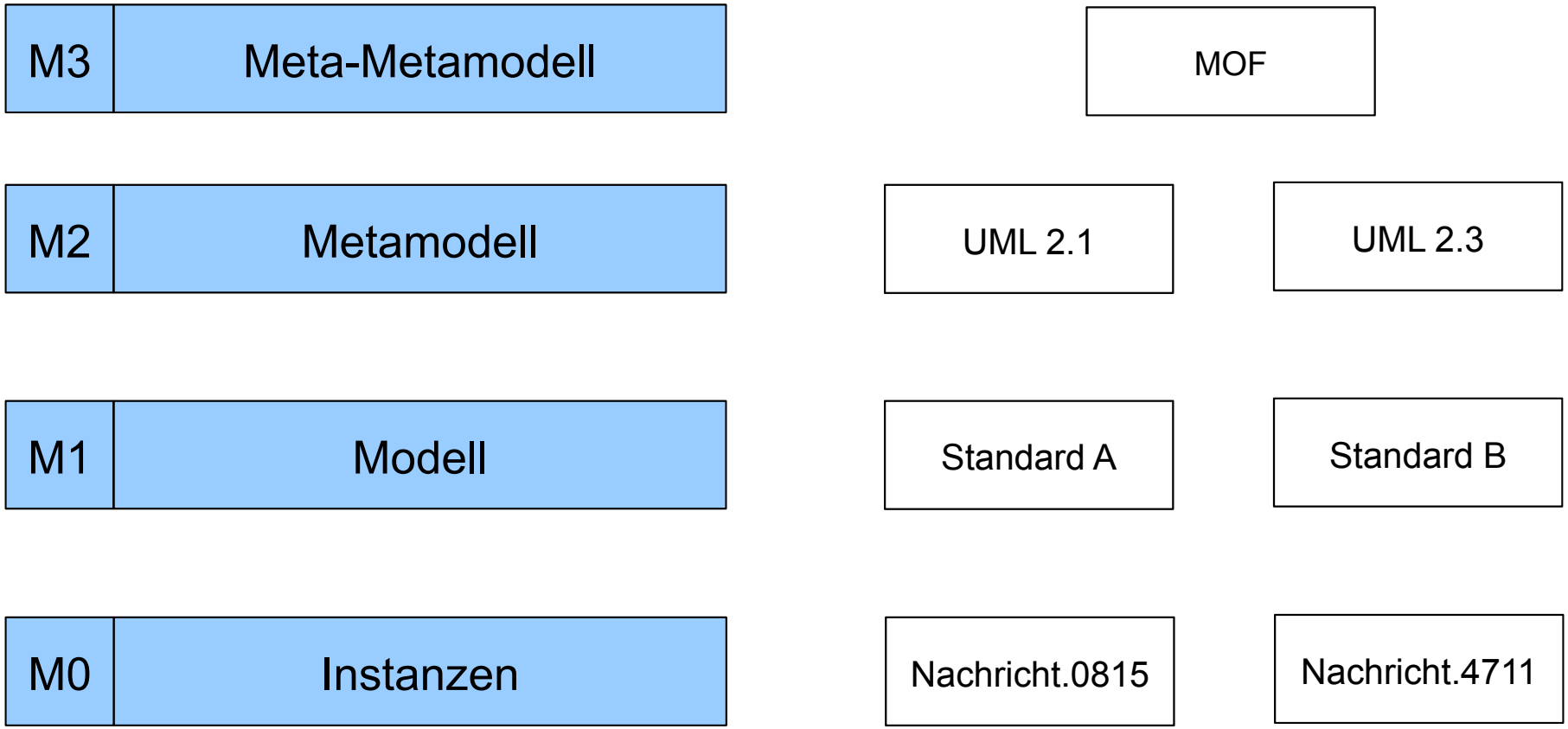
Metamodell-Architektur der OMG

M3	Meta-Metamodell	MOF: Struktur von Modellierungssprachen
M2	Metamodell	UML: Struktur von Klassendiagrammen
M1	Modell	XÖV-Standard: Struktur der Nachricht
M0	Instanzen	Betrieb: Instanz der Nachricht.0815

Metamodellversionen

- UML entwickelt sich weiter
  - Verschiedene UML Versionen
  - UML-Metamodell i.d.R. nicht abwärtskompatibel
- Ein Werkzeug benutzt i.d.R. eine UML Version
- Die Umsetzung neuer Versionen geschieht unterschiedlich schnell
- Die Verwendung unterschiedlicher Versionen führt zu Inkompatibilität trotzt XMI

Metamodell Versionen



Zusammenarbeit

- Gleichzeitig an einem Modell mit verschiedenen Werkzeugen arbeiten
- Ist für ein Produkt schon kompliziert
  - Keine Textuellen Daten
  - Diffs auf Modellebene
- Layoutinformationen bisher nicht berücksichtigt
- Mehr oder weniger unmöglich

Überblick

- Motivation
  - Kontext
  - Warum Werkzeugunabhängigkeit
  - Spannungsfeld
- Problemfelder
  - Modellaustausch
  - Zusammenarbeit
- Wege zur Werkzeugunabhängigkeit
- Fazit

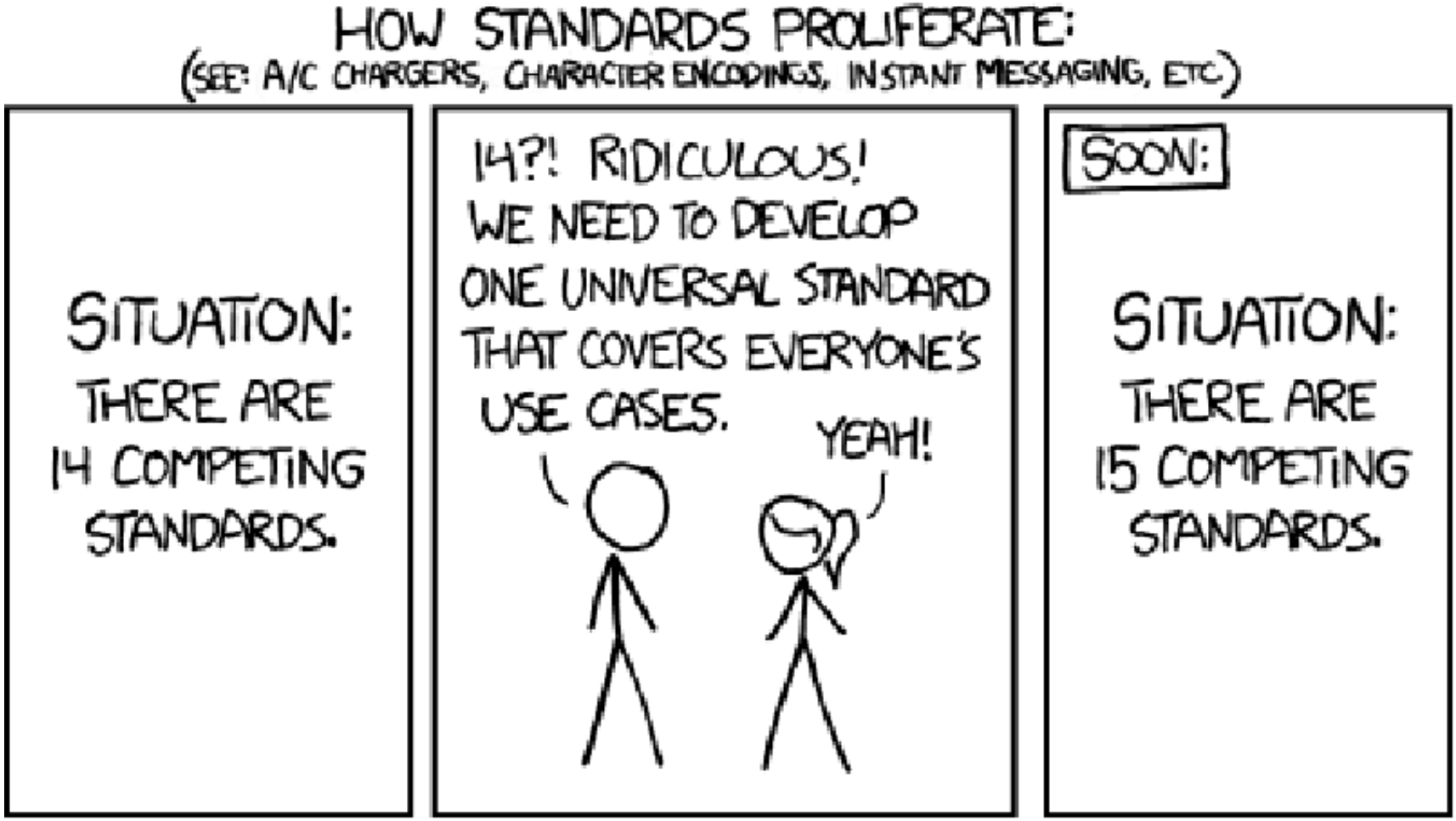
Aktuelle Entwicklungen

- Probleme wurden an die OMG herangetragen
- Gründung der Model Interchange Working Group (MIWG) im Jahr 2008
- Teilnehmer:
  - Endbenutzer
  - Werkzeughersteller (Atego, IBM, NoMagic, …)
  - Experten im Bereich UML und XMI
- Ziel: Modellaustausch testen und verbessern

MIWG: Bisherige Ergebnisse

- Im Bereich XMI Standard
  - Neuer Standard: Canonical XMI
  - Änderungsanträge an XMI
- Test Suite für XMI Modelle
- Validierungswerkzeug

Wege zur Werkzeugunabhängigkeit



Canonical XMI

- Eingeschränktes XMI
- Minimiert Variabilität der XMI-Repräsentation
- Soll so den Modellaustausch zwischen verschiedenen Werkzeugen fördern
- Beispielsweise wird die Serialisierung als XML-Element oder -Attribut geregelt
- Grundidee: Modelle sind auf textueller Ebene vergleichbar

Test Suite

- 16 Testfälle
- Ein Testfall besteht aus:
  - Diagramm(en) als Bild
  - Einer gültigen XMI-Referenz-Datei (manuell geprüft)
- Testdurchführung durch einen den Hersteller:
  - Diagramm im Werkzeug modellieren
  - Export als XMI
  - Eingabe in das Validierungswerkzeug

Übersicht der Testfälle für UML

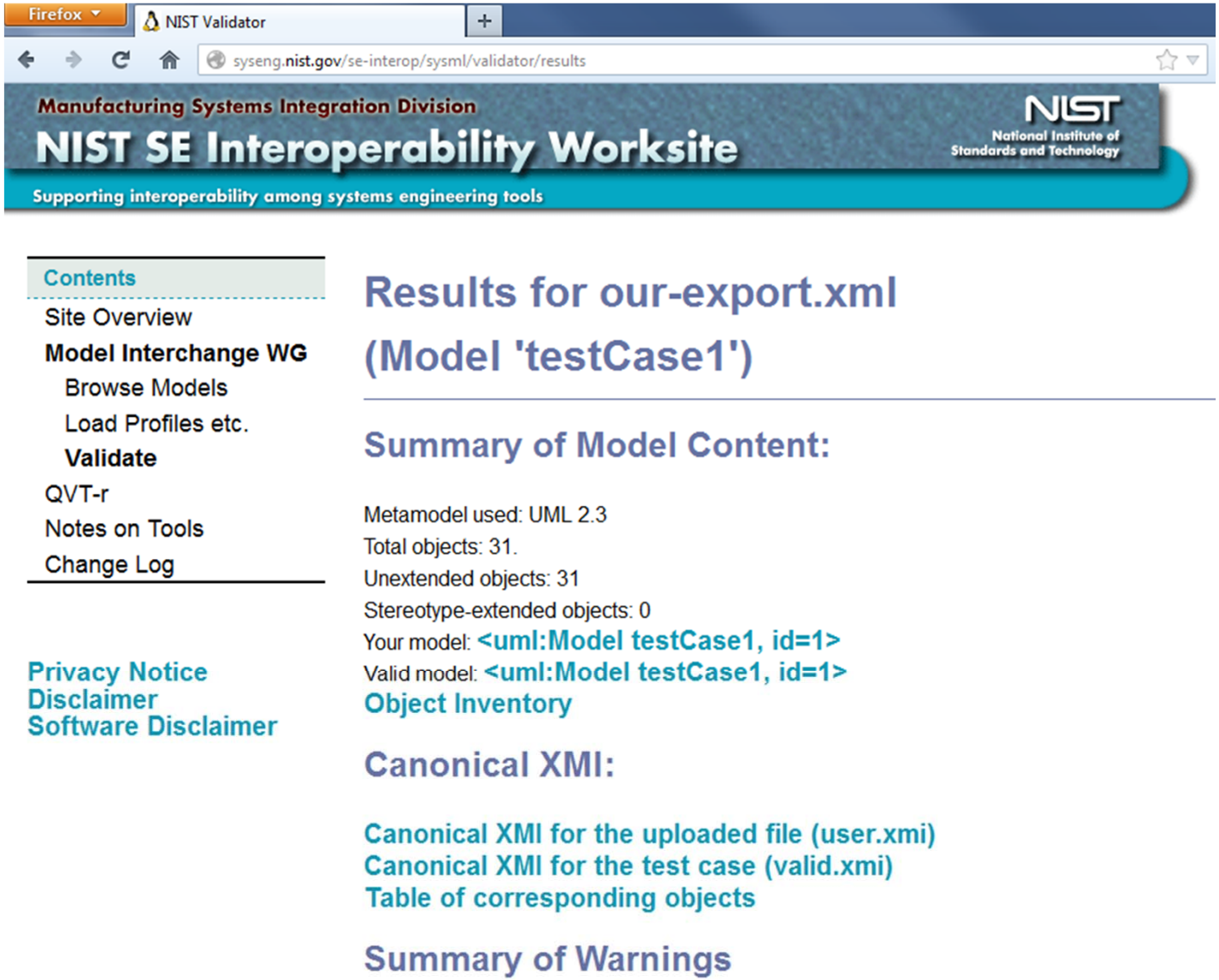
- Test Case 1 - Simple Class Model
- Test Case 2 - Advanced Class Model
- Test Case 3 - Definition and Application of Profile
- Test Case 4 - Simple Activity Model (fUML subset - executable)
- Test Case 5 - Advanced Activity Model (fUML subset - executable)
- Test Case 6 - Composite Structure
- Test Case 7 - State Machines
- Test Case 8 - Use Cases
- Test Case 9 - Interactions
- Test Case 12b - Activity Swim Lanes
- Test Case 13 - Instance Specifications
- Test Case 15 - Structured Activity Nodes

[http://www.omgwiki.org]

Validierungswerkzeug

- Importiert Modelle im XMI-Format
- Kann das gelieferte XMI allgemein testen oder
- Prüfung gegen einen der Testfälle
- Detaillierte Ausgabe der Abweichungen

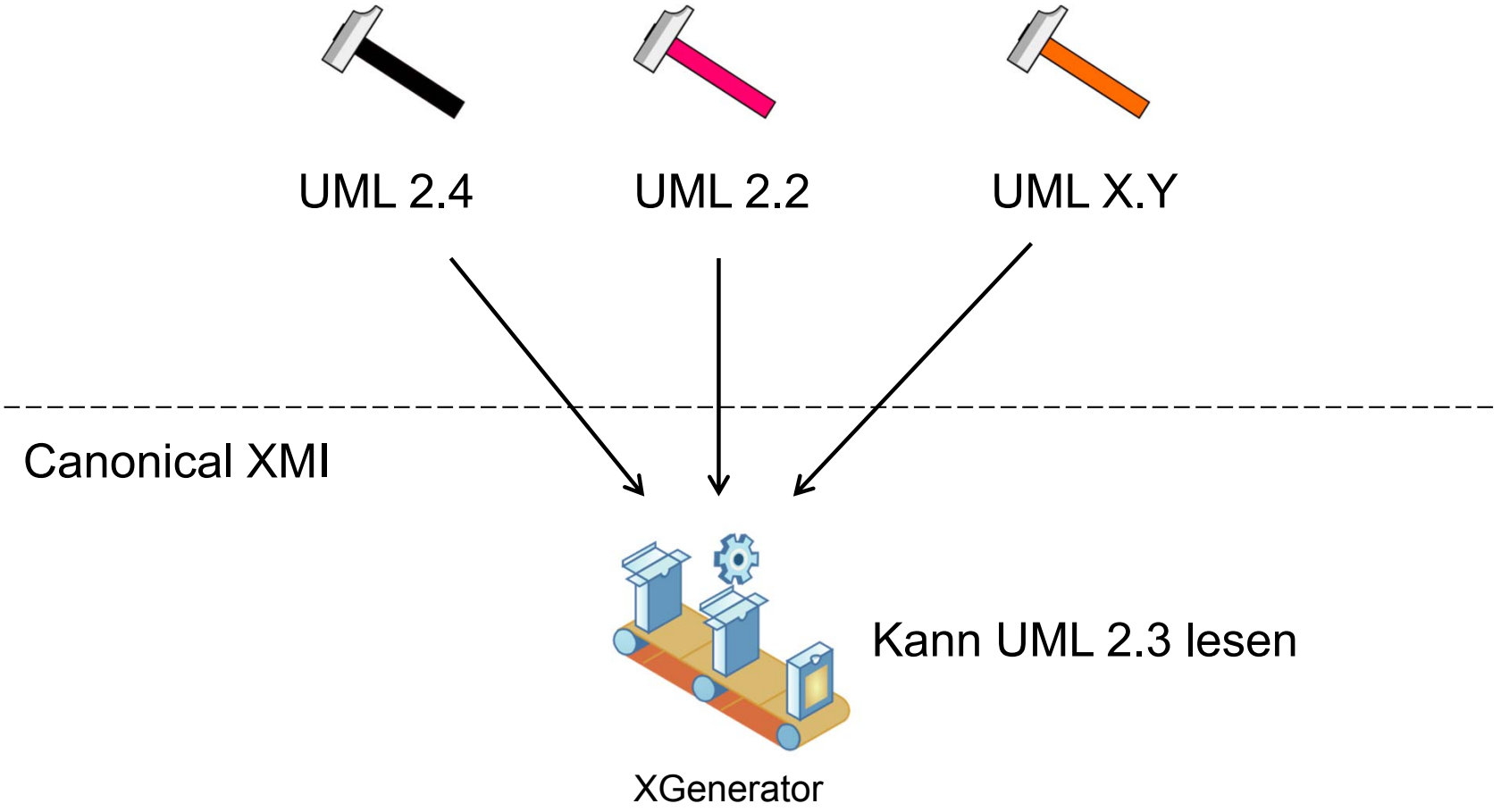
Wege zur Werkzeugunabhängigkeit



Notwendige Schritte

- KoSIT:
  - Prüfen, ob die in XÖV verwendeten UML-Konzepte durch die Testfälle abgedeckt werden
  - Bei Bedarf Testfälle einreichen
  - Verarbeitung des Referenz-XMI durch den XGenerator sicherstellen
- Projekte
  - Relevante Testfälle der Test Suite auf das verwendete Werkzeug anwenden
  - Abweichungen prüfen

Was dann nicht gelöst ist



Fazit

- Der Standard XMI ist nicht das Hauptproblem
- Problematisch ist
  - Fehler bei der Implementierung von XMI
  - Unterschiedliche Versionen des UML-Metamodells
- Layout:
  - Die meisten Werkzeuge haben eine Art Autolayout
  - Diagram Definition könnte helfen