Intelligent assistance through Formal Logic, Management of Change, Intuitive Interfaces … and Beyond

Dr. Serge Autexier

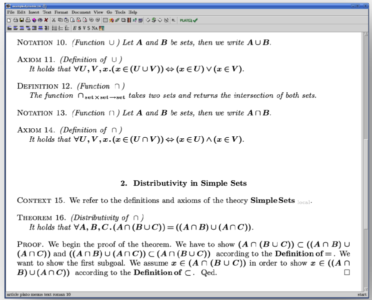
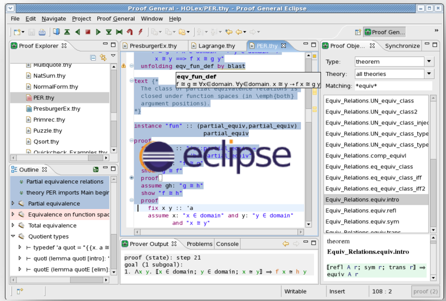
Bremen Ambient Assisted Living Lab
Research Department
Cyber-Physical Systems (CPS)

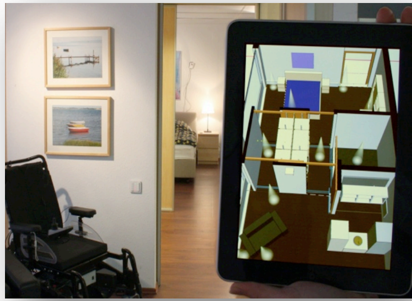
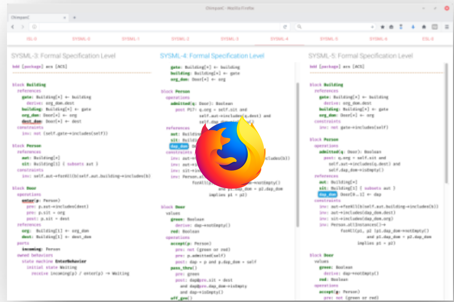
German Research Centre for Artificial Intelligence (DFKI)
Bremen site

http://www.dfki.de/cps

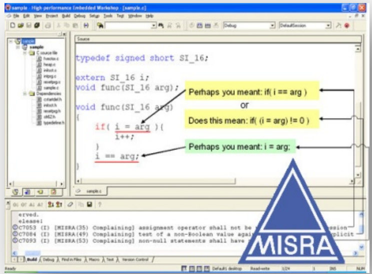
Serge.autexier@dfki.de
www.dfki.de/~serge

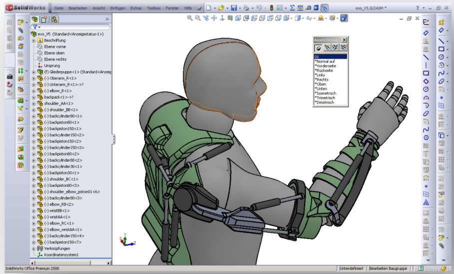
What this talk is about

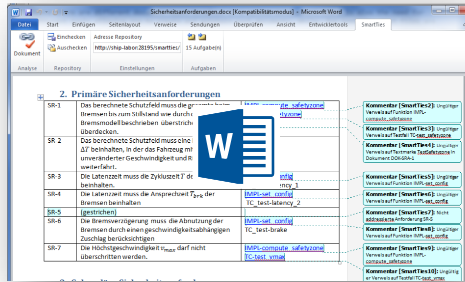



Familiar/intuitive user interfaces





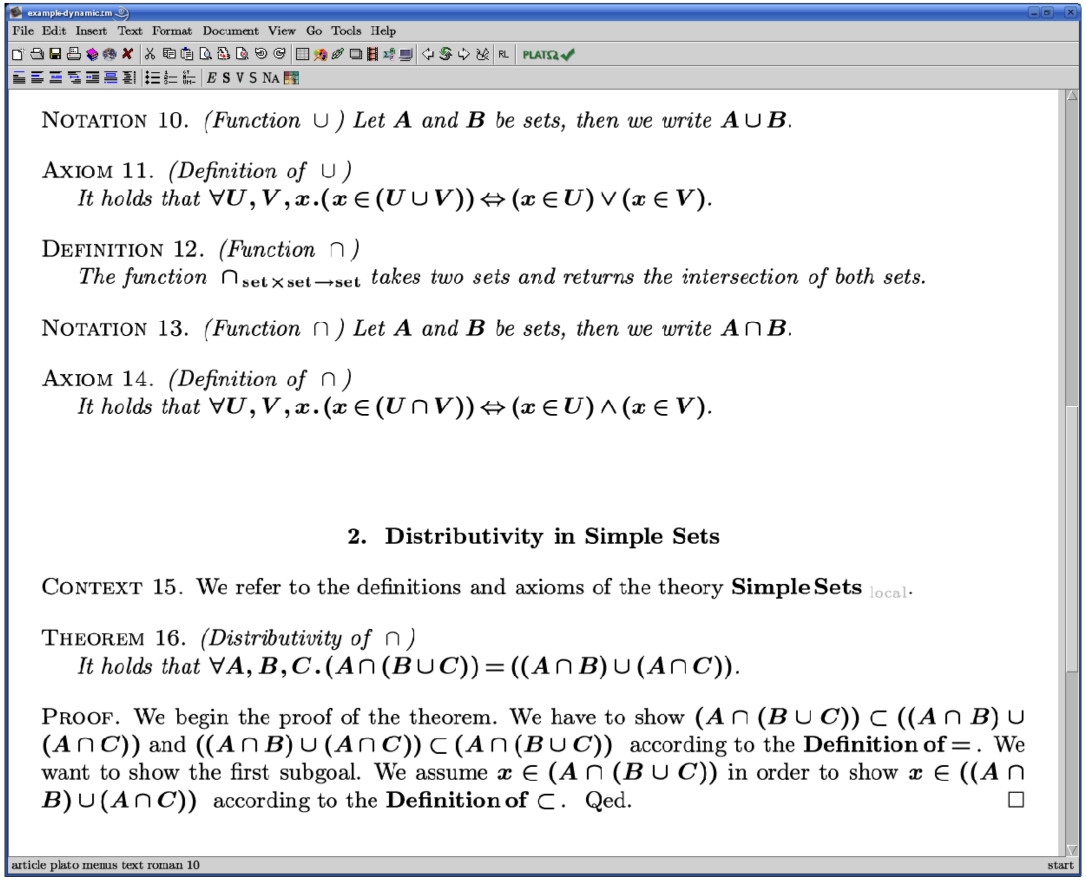


How to enable

- intelligent assistance based on rule based/formal logical reasoning directly
- through/inside familiar/intuitive user interfaces

Application in Mathematics

- Support mathematician and/or mathematics students in authoring mathematical documents
- Check definitions and notation
- For proofs
  - Verify proofs
  - Complete proofs
  - Provide more details (explanations)
  - Provide hints
- Project
  - OMEGA (SFB 378, 2005 – 2007)



NOTATION 10. *(Function $\cup$) Let A and B be sets, then we write $A \cup B$.*

AXIOM 11. *(Definition of $\cup$)*
It holds that $\forall U, V, x.(x \in (U \cup V)) \Leftrightarrow (x \in U) \vee (x \in V)$.

DEFINITION 12. *(Function $\cap$)*
The function $\cap_{\mathbf{set} \times \mathbf{set} \to \mathbf{set}}$ *takes two sets and returns the intersection of both sets.*

NOTATION 13. *(Function $\cap$) Let A and B be sets, then we write $A \cap B$.*

AXIOM 14. *(Definition of $\cap$)*
It holds that $\forall U, V, x.(x \in (U \cap V)) \Leftrightarrow (x \in U) \wedge (x \in V)$.

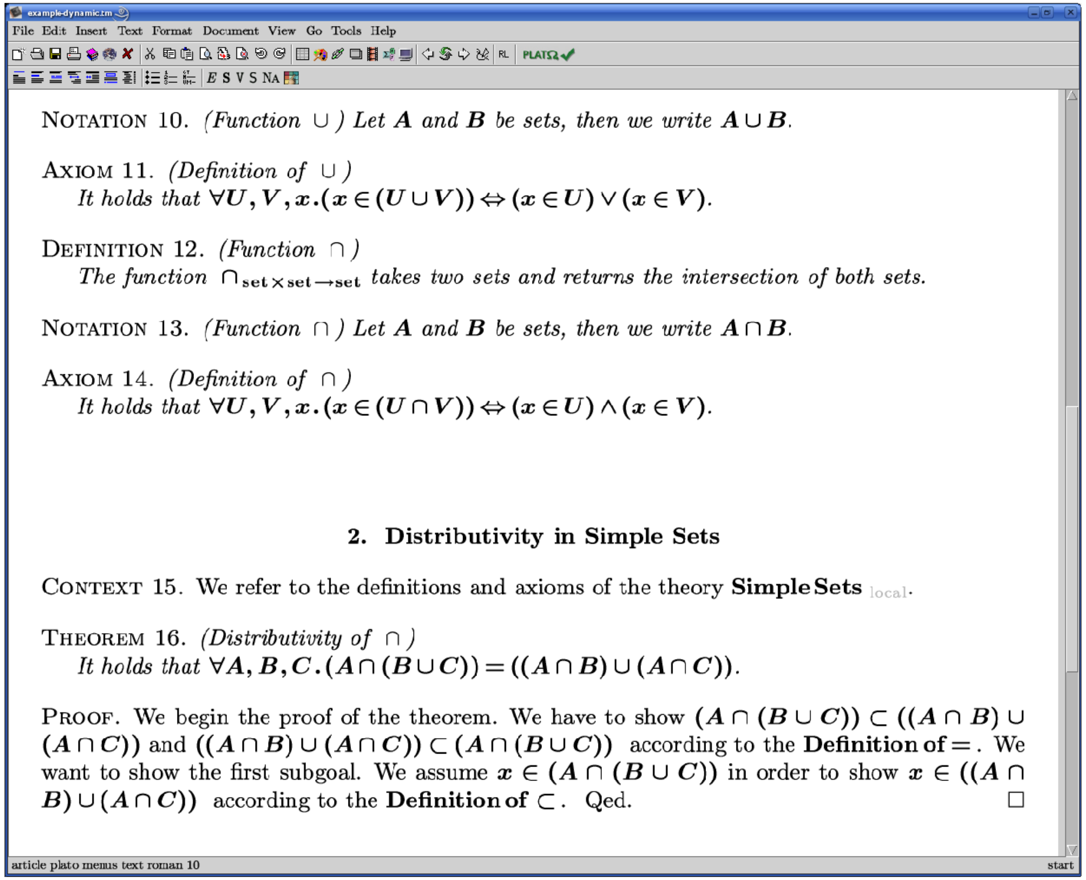
2. Distributivity in Simple Sets

CONTEXT 15. We refer to the definitions and axioms of the theory **Simple Sets** $_{\text{local}}$.

THEOREM 16. *(Distributivity of $\cap$)*
It holds that $\forall A, B, C.(A \cap (B \cup C)) = ((A \cap B) \cup (A \cap C))$.

PROOF. We begin the proof of the theorem. We have to show $(A \cap (B \cup C)) \subset ((A \cap B) \cup (A \cap C))$ and $((A \cap B) \cup (A \cap C)) \subset (A \cap (B \cup C))$ according to the **Definition of** $=$. We want to show the first subgoal. We assume $x \in (A \cap (B \cup C))$ in order to show $x \in ((A \cap B) \cup (A \cap C))$ according to the **Definition of** $\subset$. Qed. □

Transforming Editor Syntax into Proof Assistant Syntax



File Edit Insert Text Format Document View Go Tools Help

NOTATION 10. *(Function $\cup$) Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be sets, then we write $\boldsymbol{A} \cup \boldsymbol{B}$.*

AXIOM 11. *(Definition of $\cup$)*
It holds that $\forall \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{x}.(\boldsymbol{x} \in (\boldsymbol{U} \cup \boldsymbol{V})) \Leftrightarrow (\boldsymbol{x} \in \boldsymbol{U}) \vee (\boldsymbol{x} \in \boldsymbol{V})$.

DEFINITION 12. *(Function $\cap$)*
The function $\cap_{\mathbf{set} \times \mathbf{set} \rightarrow \mathbf{set}}$ takes two sets and returns the intersection of both sets.

NOTATION 13. *(Function $\cap$) Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be sets, then we write $\boldsymbol{A} \cap \boldsymbol{B}$.*

AXIOM 14. *(Definition of $\cap$)*
It holds that $\forall \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{x}.(\boldsymbol{x} \in (\boldsymbol{U} \cap \boldsymbol{V})) \Leftrightarrow (\boldsymbol{x} \in \boldsymbol{U}) \wedge (\boldsymbol{x} \in \boldsymbol{V})$.

2. Distributivity in Simple Sets

CONTEXT 15. We refer to the definitions and axioms of the theory **Simple Sets** $_{\text{local}}$.

THEOREM 16. *(Distributivity of $\cap$)*
It holds that $\forall \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}.(\boldsymbol{A} \cap (\boldsymbol{B} \cup \boldsymbol{C})) = ((\boldsymbol{A} \cap \boldsymbol{B}) \cup (\boldsymbol{A} \cap \boldsymbol{C}))$.

PROOF. We begin the proof of the theorem. We have to show $(\boldsymbol{A} \cap (\boldsymbol{B} \cup \boldsymbol{C})) \subset ((\boldsymbol{A} \cap \boldsymbol{B}) \cup (\boldsymbol{A} \cap \boldsymbol{C}))$ and $((\boldsymbol{A} \cap \boldsymbol{B}) \cup (\boldsymbol{A} \cap \boldsymbol{C})) \subset (\boldsymbol{A} \cap (\boldsymbol{B} \cup \boldsymbol{C}))$ according to the **Definition of** $=$. We want to show the first subgoal. We assume $\boldsymbol{x} \in (\boldsymbol{A} \cap (\boldsymbol{B} \cup \boldsymbol{C}))$ in order to show $\boldsymbol{x} \in ((\boldsymbol{A} \cap \boldsymbol{B}) \cup (\boldsymbol{A} \cap \boldsymbol{C}))$ according to the **Definition of** $\subset$. Qed. $\square$

article plato menus text roman 10 start

Combined Editor and Formal Representation

Axiom “Subset”:

! U,V . U ⊂ V ⇔ ! x . x in U => x in V

“We define ∀U, V . U ⊂ V ⇔ ∀

Axiom “Set E

! U, V

Axiom “Inter

! U, V, x

”It holds ∀U
V)”

Theorem “Distributivity of ∩”:

! A, B, C . A ∩ (B ∪ C) ⇔ (A ∩ B) ∪ (A ∩ C)

“It holds that ∀ A, B, C . A ∩ (B ∪ C) = (A ∩ B) ∪ (A ∩ C)”

Proof of “Distributivity of ∩”:

“We begin the proof of the theorem”

“Set

A ∩ B) ∪

”

to show x ∈
tion of ⊂”

Combined representation is essential

to not overwrite the users formulations

when including changes

from the proof assistant back into the editor

Application in Mathematics

NOTATION 10. *(Function $\cup$) Let A and B be sets, then we write $A \cup B$.*

AXIOM 11. *(Definition of $\cup$)*
It holds that $\forall U, V, x.(x \in (U \cup V)) \Leftrightarrow (x \in U) \vee (x \in V)$.

DEFINITION 12. *(Function $\cap$)*
The function $\cap_{\mathrm{set} \times \mathrm{set} \to \mathrm{set}}$ takes two sets and returns the intersection of both sets.

NOTATION 13. *(Function $\cap$) Let A and B be sets, then we write $A \cap B$.*

AXIOM 14. *(Definition of $\cap$)*
It holds that $\forall U, V, x.(x \in (U \cap V)) \Leftrightarrow (x \in U) \wedge (x \in V)$.

2. Distributivity in Simple Sets

THEOREM 16. *(Distributivity of $\cap$)*
It holds that $\forall A, B, C.(A \cap (B \cup C)) = ((A \cap B) \cup (A \cap C))$.

PROOF. We begin the proof of the theorem. We have to show $(A \cap (B \cup C)) \subset ((A \cap B) \cup (A \cap C))$ and $((A \cap B) \cup (A \cap C)) \subset (A \cap (B \cup C))$ according to the **Definition of** $=$. We want to show the first subgoal. We assume $x \in (A \cap (B \cup C))$ in order to show $x \in ((A \cap B) \cup (A \cap C))$ according to the **Definition of** $\subset$. Qed. □

article plato menus text roman 10 start

Axiom “Subset”:

! U,V . U ⊂ V ⇔ ! x . x in U => x in V

Axiom “Set Equality”:

! U, V . U = V ⇔ (U ⊂ V) &(V ⊂ U)

Axiom “Intersection Def.” :

! U, V, x . (x in U & x in V) ⇔ x in (U ∩ V)

Theorem “Distributivity of ∩”:

! A, B, C . A ∩ (B ∪ C) ⇔ (A ∩ B) ∪ (A ∩ C)

Proof of “Distributivity of ∩”:

Subgoals

A ∩ (B ∪ C) ⊂ (A ∩ B) ∪ (A ∩ C) by “Set Equality”:

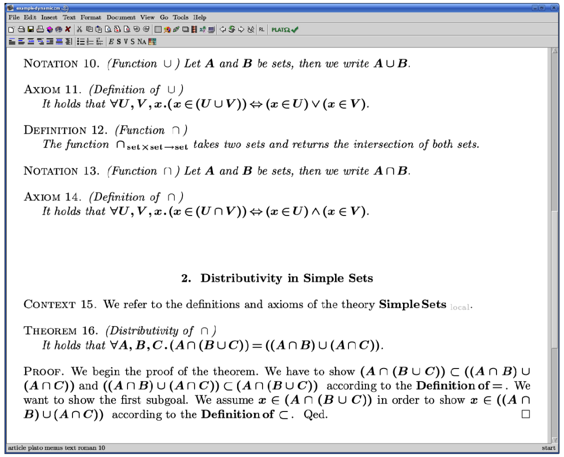
Assume x in A ∩ (B ∪ C) from “Subset”.

Handling inside Theorem Prover

```
Axiom "Subset":
      ! U,V . U ⊂ V ⇔ ! x . x in U => x in V

Axiom "Set Equality":
      ! U, V . U = V ⇔ (U ⊂ V) &(V ⊂ U)

Axiom "Intersection Def." :
  ! U, V, x .  (x in U & x in V) ⇔ x in (U ∩ V)

Theorem "Distributivity of ∩":
   ! A, B, C . A ∩ (B ∪ C) ⇔ (A ∩ B) ∪ (A ∩ C)

Proof of "Distributivity of ∩":
Subgoals
A ∩ (B ∪ C) ⊂ (A ∩ B) ∪ (A ∩ C) by "Set Equality":
 Assume x in A ∩ (B ∪ C) from "Subset".
```

```
Axiom "Subset":
      ! U,V . U ⊂ V ⇔ ! x . x in U => x in V

Axiom "Set Equality":
      ! U, V . U = V ⇔ (U ⊂ V) &(V ⊂ U)

Axiom "Intersection Def." :
  ! U, V, x .  (x in U & x in V) ⇔ x in (U ∩ V)

Theorem "Distributivity of ∩":
   ! A, B, C . A ∩ (B ∪ C) ⇔ (A ∩ B) ∪ (A ∩ C)

Proof of "Distributivity of ∩":
Subgoals
A ∩ (B ∪ C) ⊂ (A ∩ B) ∪ (A ∩ C) by "Set Equality":
 Assume x in A ∩ (B ∪ C) from "Subset"
 Fact x in A & x in (B U C) by "Intersection Def".
 Fact x in A & (X in B | x in C) by "Union Def"
```

Application in Mathematics


NOTATION 10. (Function $\cup$) Let A and B be sets, then we write $A \cup B$.

AXIOM 11. (Definition of $\cup$)
It holds that $\forall U, V, x.(x \in (U \cup V)) \Leftrightarrow (x \in U) \vee (x \in V)$.

DEFINITION 12. (Function $\cap$)
The function $\cap_{set \times set \rightarrow set}$ takes two sets and returns the intersection of both sets.

NOTATION 13. (Function $\cap$) Let A and B be sets, then we write $A \cap B$.

AXIOM 14. (Definition of $\cap$)
It holds that $\forall U, V, x.(x \in (U \cap V)) \Leftrightarrow (x \in U) \wedge (x \in V)$.

2. Distributivity in Simple Sets

CONTEXT 15. We refer to the definitions and axioms of the theory SimpleSets

THEOREM 16. (Distributivity of $\cap$)
It holds that $\forall A, B, C.(A \cap (B \cup C)) = ((A \cap B) \cup (A \cap C))$.

PROOF. We begin the proof of the theorem. We have to show $(A \cap (B \cup C)) \subset ((A \cap B) \cup (A \cap C))$ and $((A \cap B) \cup (A \cap C)) \subset (A \cap (B \cup C))$ according to the Definition of $=$. We want to show the first subgoal. We assume $x \in (A \cap (B \cup C))$ in order to show $x \in ((A \cap B) \cup (A \cap C))$ according to the Definition of $\subset$. Qed. □

```
Axiom "Subset":
    ! U,V . U ⊂ V ⇔ !! x . x in U ⇒ x in V

Axiom "Set Equality":
    ! U, V . U = V ⇔ ((U ⊂ V) & (V ⊂ U))

Axiom "Intersection Def.":
  ! U, V, x .  (x in U & x in V) ⇔ x in (U ∩ V)

Section "Distributivity in Simple Sets"
   ! A, B, C . A ∩ (B ∪ C) ⇔ (A ∩ B) ∪ (A ∩ C)
Theorem "Distributivity of ∩":
Proof of "Distributivity of ∩":
Subgoals
A ∩ (B ∪ C) ⊂ (A ∩ B) ∪ (A ∩ C) by "Set Equality":
"We begin the proof of the theorem"
Subgoals
 Fact x in A & x in (B U C) by "Intersection Def".
A ∩ (B ∪ C) ⊂ (A ∩ B) ∪ (A ∩ C) by "Set Equality":
 Fact x in A & (X in B | x in C) by "Union Def"
 Assume x in A ∩ (B ∪ C) from "Subset".
 Fact x in A & x in (B U C) by "Intersection Def".
 Fact x in A & (X in B | x in C) by "Union Def"
```

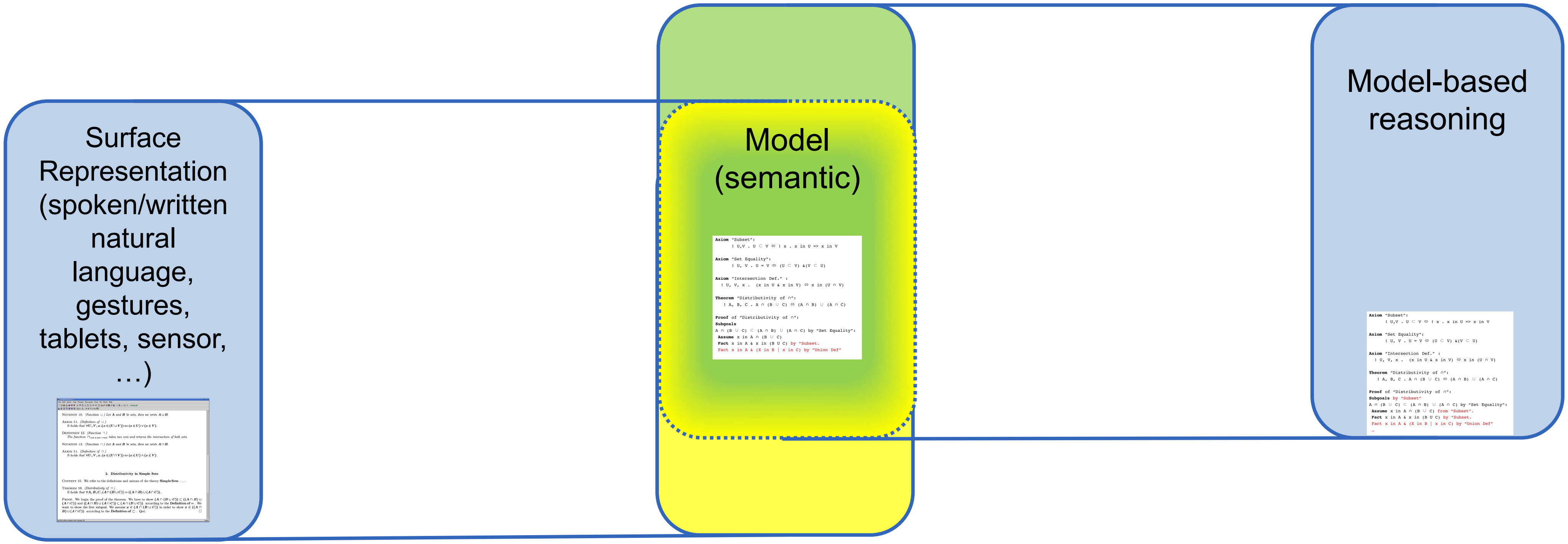
```
Axiom "Subset":
    ! U,V . U ⊂ V ⇔ !! x . x in U ⇒ x in V

Axiom "Set Equality":
    ! U, V . U = V ⇔ ((U ⊂ V) & (V ⊂ U))

Axiom "Intersection Def.":
  ! U, V, x .  (x in U & x in V) ⇔ x in (U ∩ V)

Theorem "Distributivity of ∩":
  ! A, B, C . A ∩ (B ∪ C) ⇔ (A ∩ B) ∪ (A ∩ C)

Proof of "Distributivity of ∩":
Subgoals
A ∩ (B ∪ C) ⊂ (A ∩ B) ∪ (A ∩ C) by "Set Equality":
 Assume x in A ∩ (B ∪ C) from "Subset".
 Fact x in A & x in (B U C) by "Intersection Def".
 Fact x in A & (X in B | x in C) by "Union Def"
```

Push relevant changes

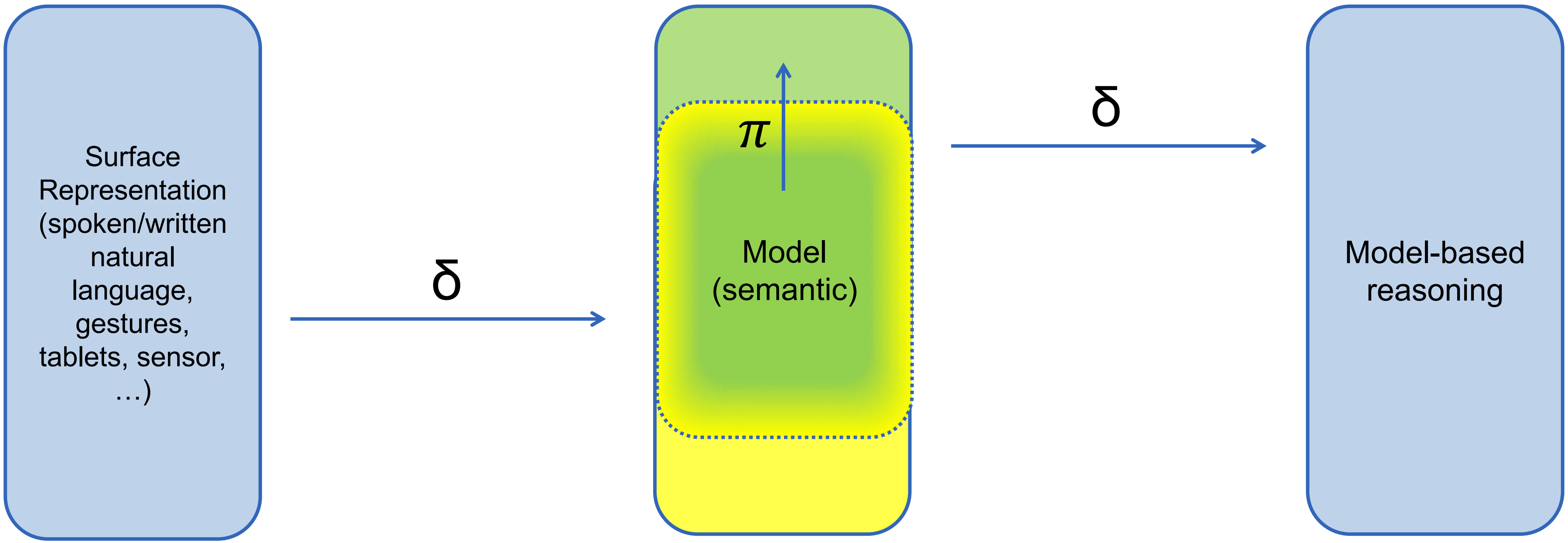
Propagate Changes

Push relevant changes

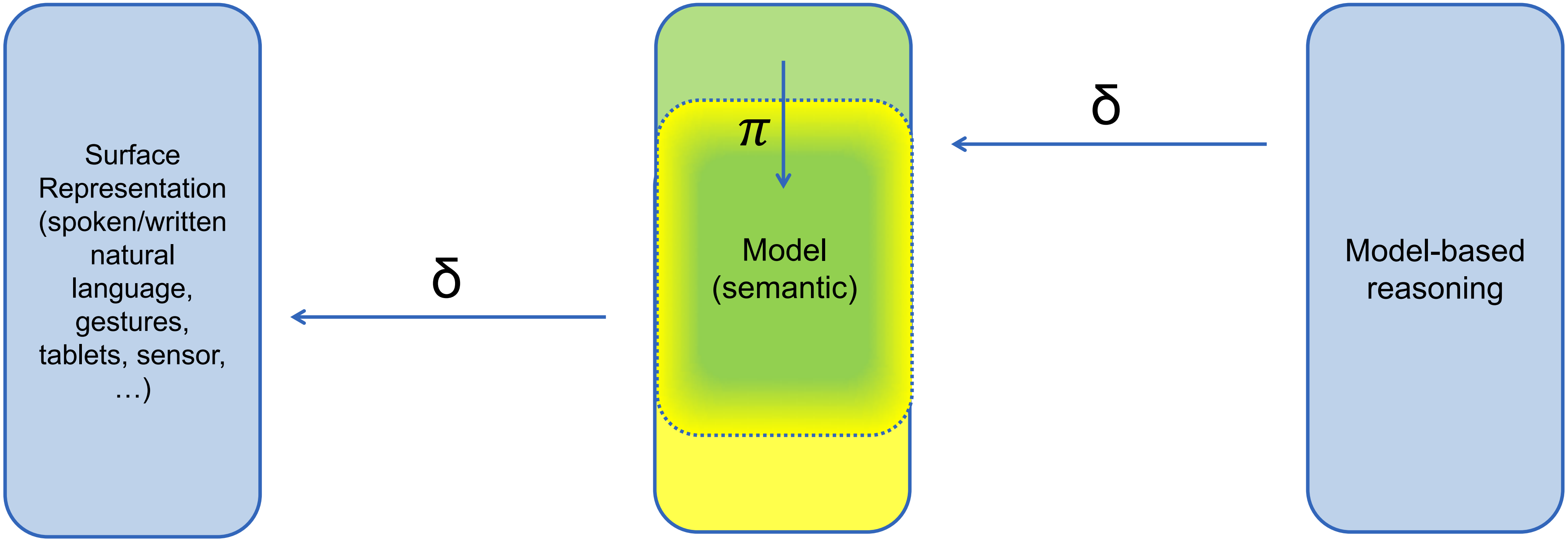
General Principle



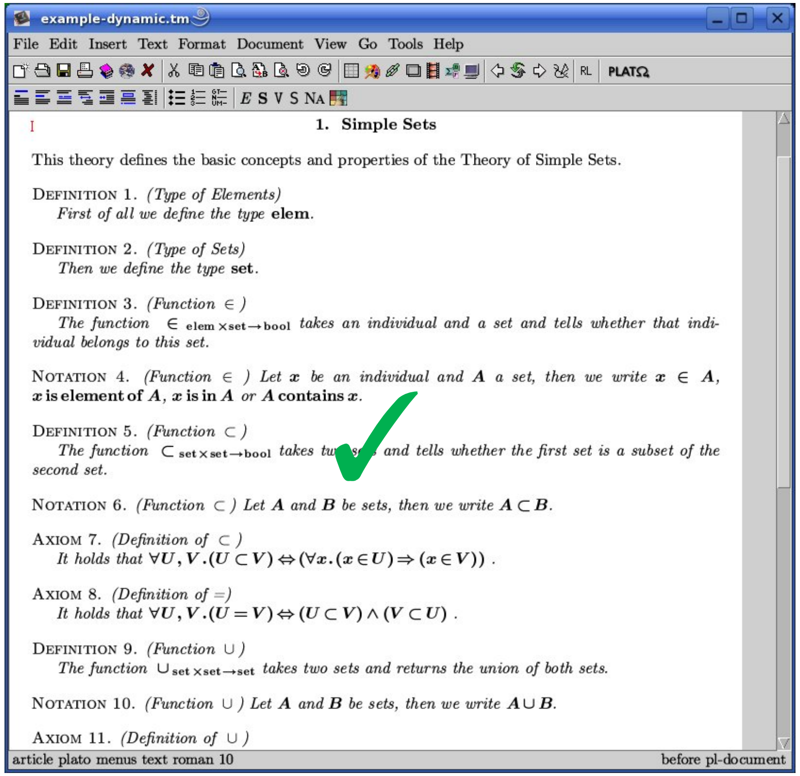
General Principle: From UI to Reasoner



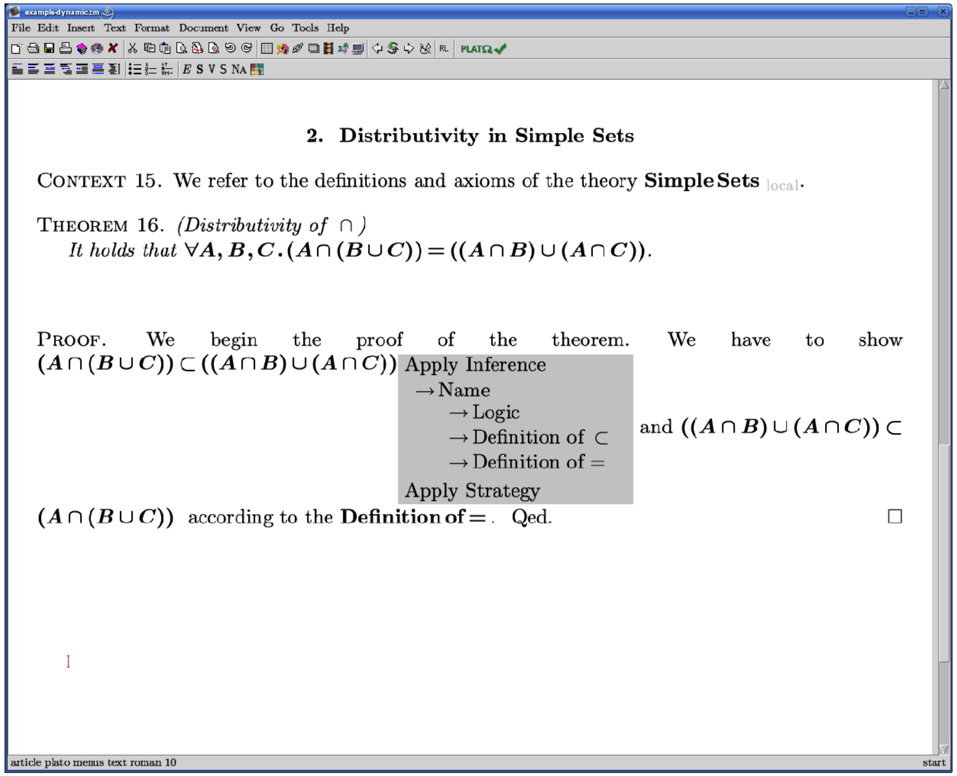
General Principle: From Reasoner to UI



Services offered by Proof Assistant



Checking soundness of textbooks / articles



Interactive proof assistance inside the editor

In Situ Explanations from Proof Assistant

Introduction to Algebra
Thomas H.

1 Logic

2 Classes and Sets

In Gödel-Bernays form of axiomatic set theory, which we shall follow, the primitive (undefined) notions are **class, membership**, and **equality**. Intuitively, we consider a class to be a collection A of objects (elements) such that given any object x if it is possible to determine whether or not x is a member (or element) of A. We write $x \in A$ for "x is an element of A" and $x \notin A$ for "x is not an element of A".

[. . .]

The **axiom of extensionality** asserts that two classes with the same elements are equal (formally, $[x \in A \Leftrightarrow x \in B] \Rightarrow A = B$).

A class A is defined to be a **set** if and only if there exists a class B such that $A \in B$. Thus a set is a particular kind of class. A class that is not a set is called a **proper class**. Intuitively the distinction between sets and proper classes is not too clear. Roughly speaking a set is a "small" class and a proper class is exceptionnally "large". The **axiom of class formation** asserts that for any statement $P(y)$ in the first-order predicate calculus involving a variable y, there exists a class A such that $x \in A$ if and only if x is a set and the statement $P(x)$ is true. We denote this class A by $\{x \mid P(x)\}$.

[. . .]

A class A is a **subclass** of a class B (written $A \subset B$) provided:

$$\text{for all } x \in A, x \in A \Rightarrow x \in B.$$

By the axioms of extensionality and the properties of equality *Details*

$$A = B \Leftrightarrow A \subset B \text{ and } B \subset A$$

Details We first prove $A = B \Rightarrow A \subset B$ and $B \subset A$: Assume (h) $A = B$, then we have to prove (1) $A \subset B$ and (2) $B \subset A$: For (1), assuming $x \in A$, we conclude $x \in B$ from (h) and properties of equality. For (2), assuming $x \in B$, we conclude $x \in A$ from (h) and properties of equality. Conversely, we prove $A \subset B$ and $B \subset A \Rightarrow A = B$: By Definition of $\subset$ we know from $A \subset B$ and $B \subset A$ that $x \in A \Rightarrow x \in B$ and $x \in B \Rightarrow x \in A$ for all x. Hence, $x \in A \Leftrightarrow x \in B$ for all x and by extensionality follows $A = B$. □

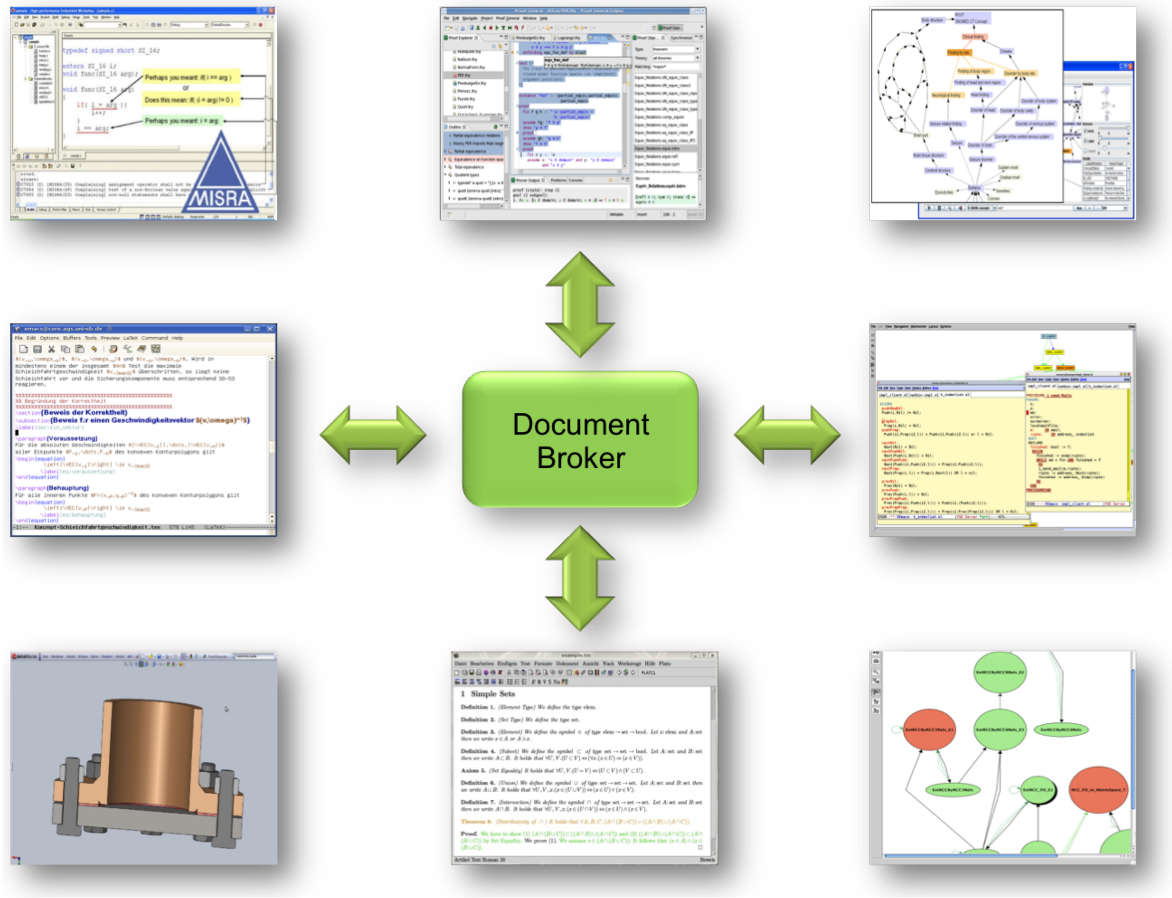
3 Relations

4 Functions

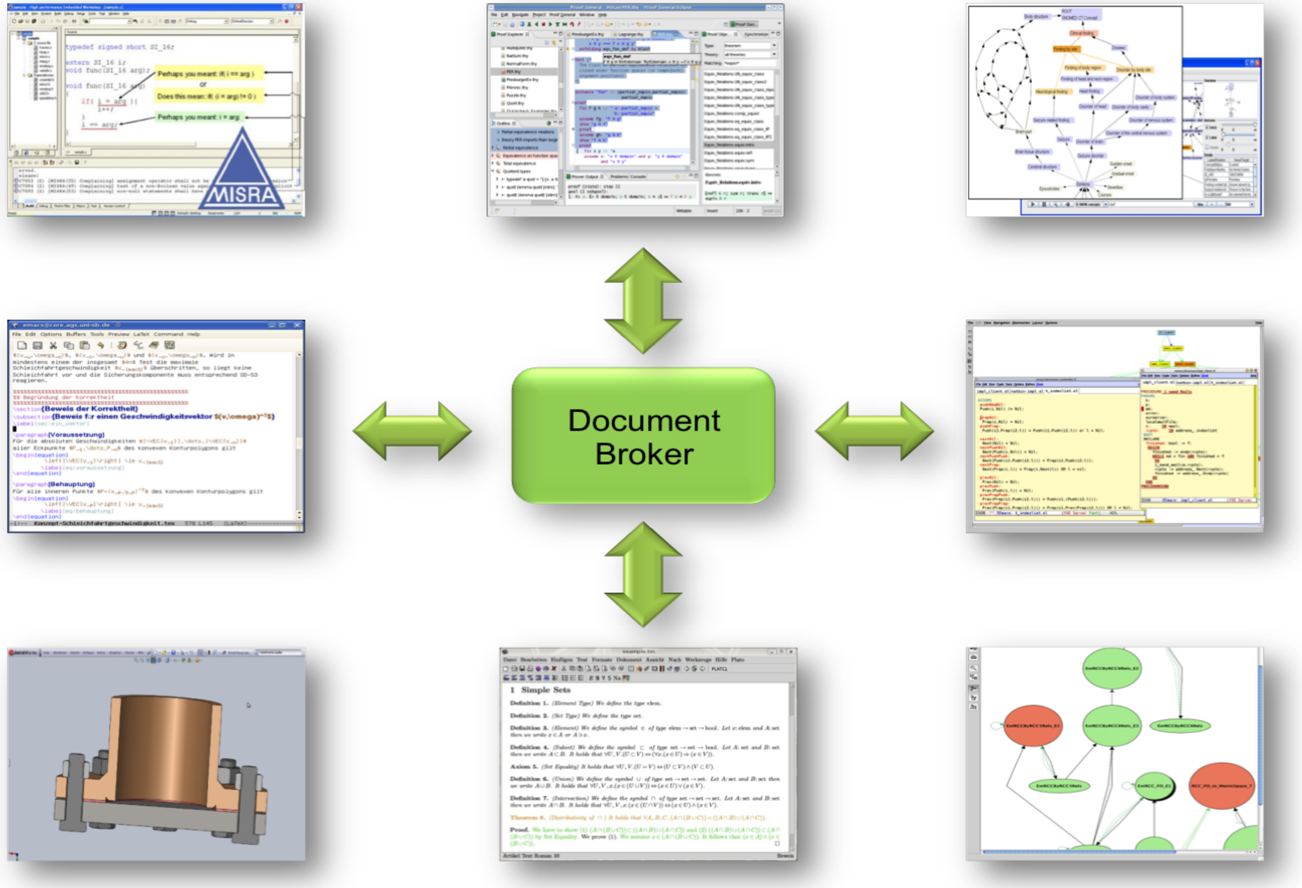
FormalSafe

Project FormalSafe (BMBF 2008 – 2010)

- Formal Methods Tools
  - Heterogeneous specification and system
  - Proof support through VSE
- Document centered development
  - Integration of formal and informal development documents
  - Dependencies and traceability
- Management of change
  - Evolutionary (agile) development
  - Formal integrity

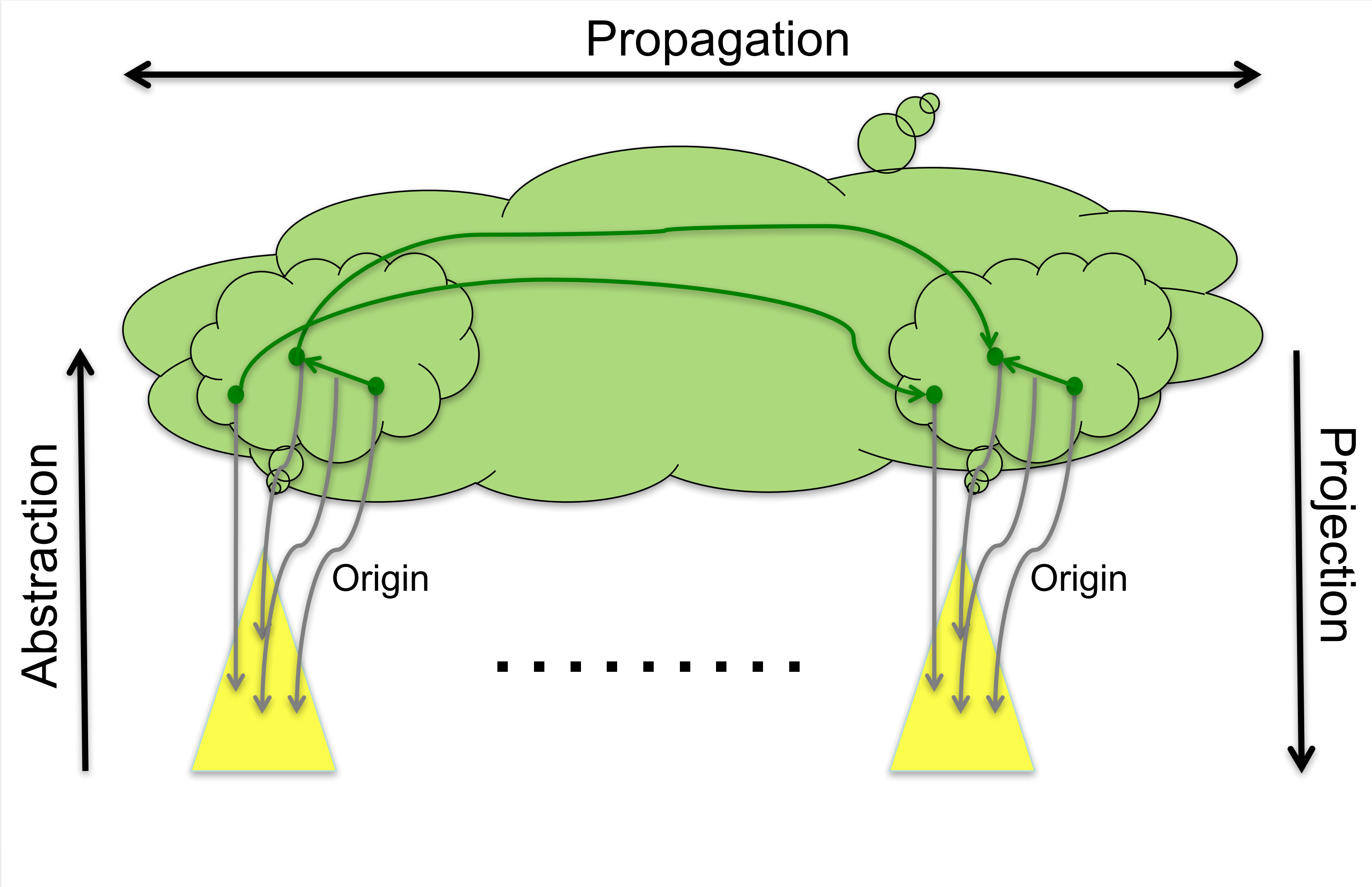


FormalSafe Broker aka DocTIP



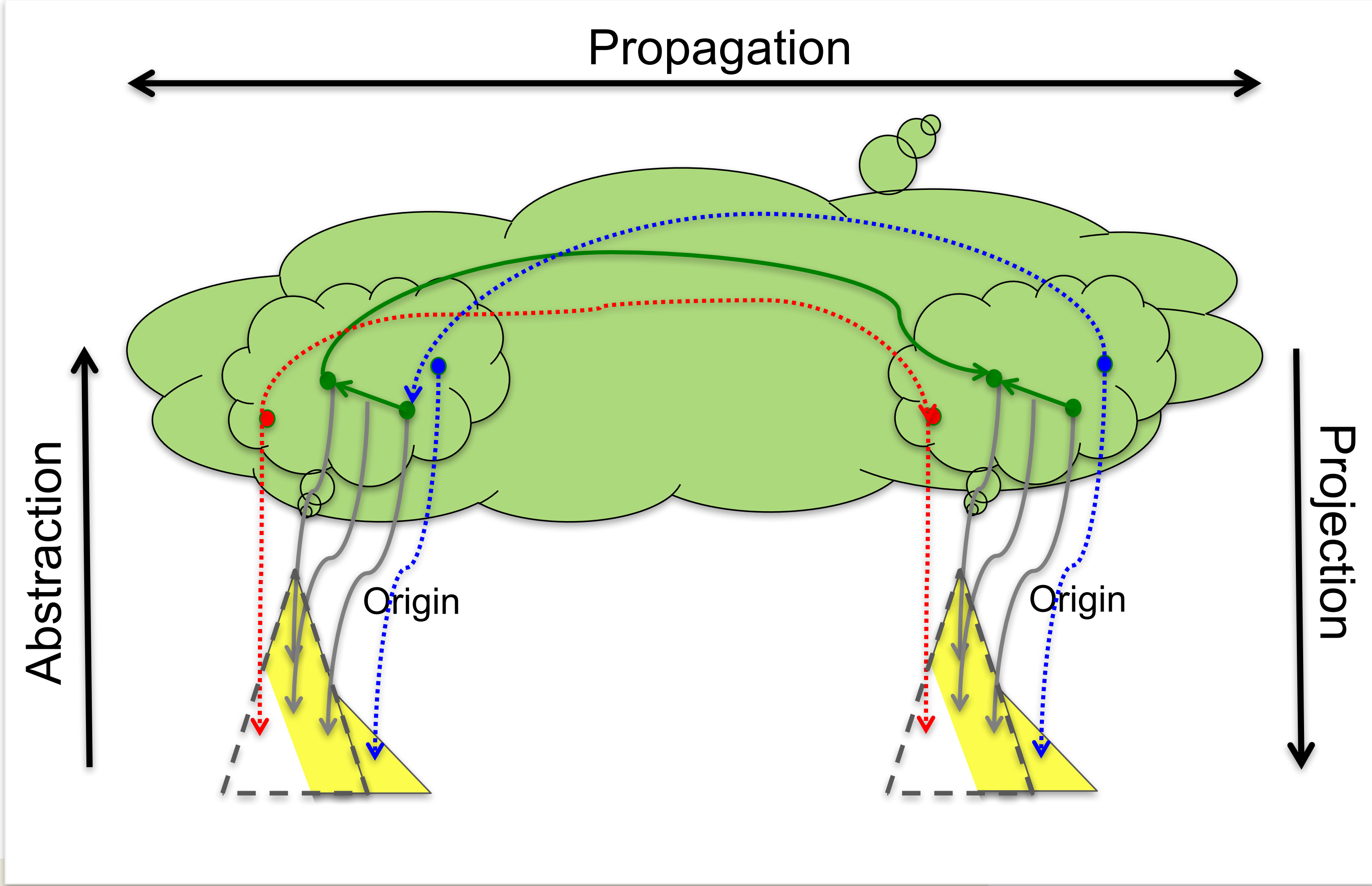
Related multiple documents

- Different types of documents
- Different types of reasoner syntaxes
- Automatically establish and maintain **combined representations**



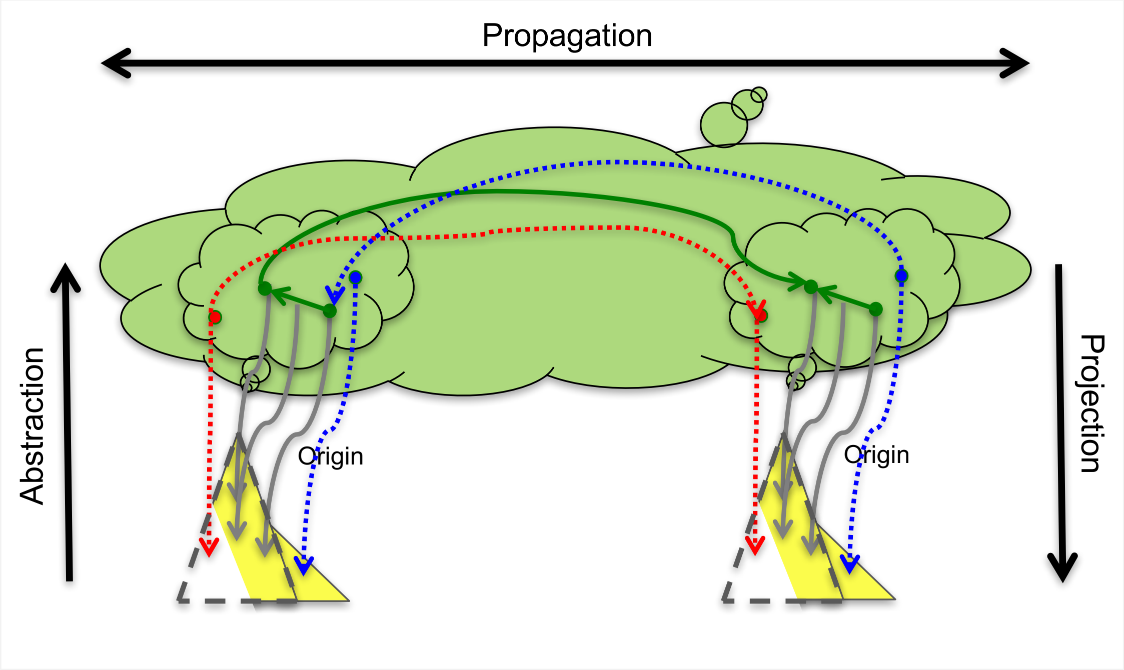
Related multiple documents

- Different types of documents
- Different types of reasoner syntaxes
- Automatically establish and maintain **combined representations**
- Propagate changes from one document or reasoner to **all other** documents and reasoner



Parameterized Change Management

- Semantic Difference Analysis
  - Parameterized over document type **specific similarity specifications**
- Change Propagation
  - Representation of syntactic and semantic document parts as **typed graphs**
  - Parameterized over Document Specific Propagation rules (as **graph rewriting rules**)



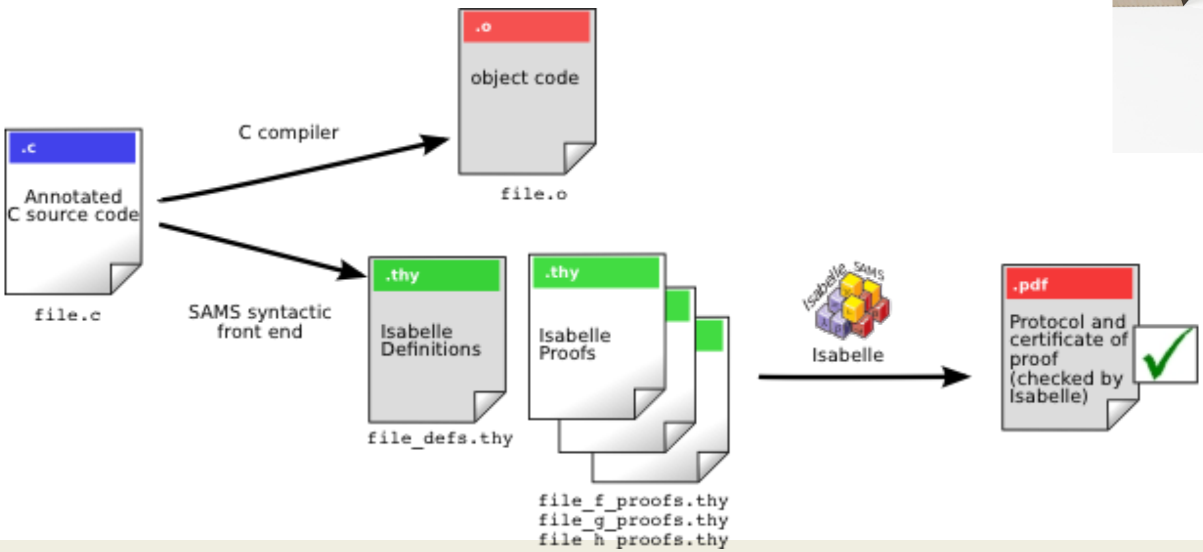
Right methodology, bad scalability

FormalSafe Applications

Application: Formal Verification of C Programs

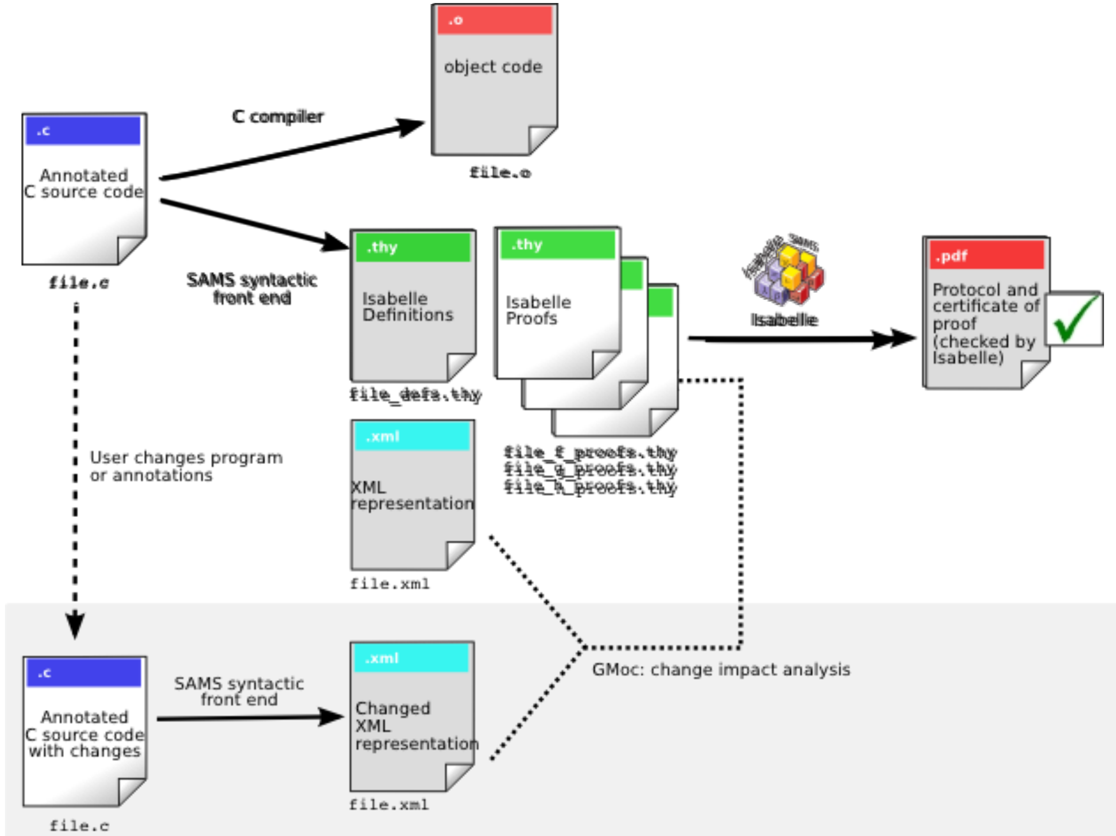
```
/*@ @requires 0 <= w
             && w < sams_config.brkdist.measurements[0].v
             && brkconfig_OK(sams_config)
    @modifies \nothing
    @ensures 0 < \result
             && \result < sams_config.brkdist.length
             && sams_config.brkdist.measurements[\result-1].v > w
             && w >= sams_config.brkdist.measurements[\result].v
    @*/
Int32 bin_search_idx_v( Float32 w);
```

- SAMS: Formal Verification of MISRA C programs
  - annotated by preconditions, postconditions and modification information
- Used to verify algorithm computing safety zone
- Modular verification of each C-Function
- Workflow



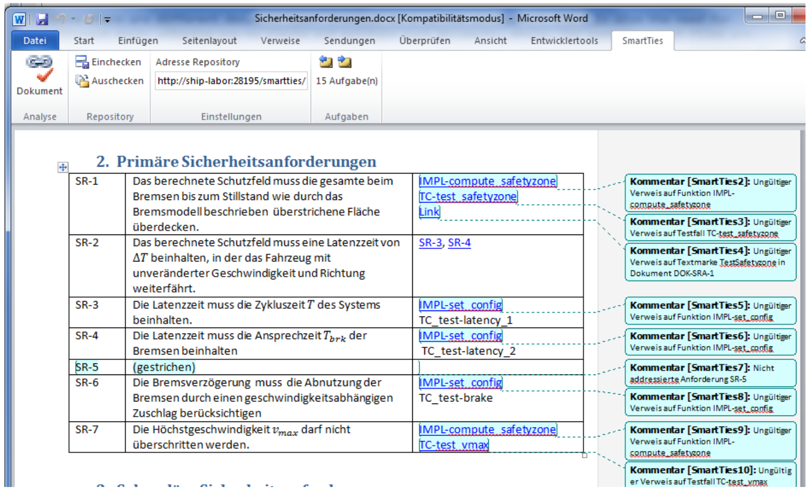
Application: Formal Verification of C Programs

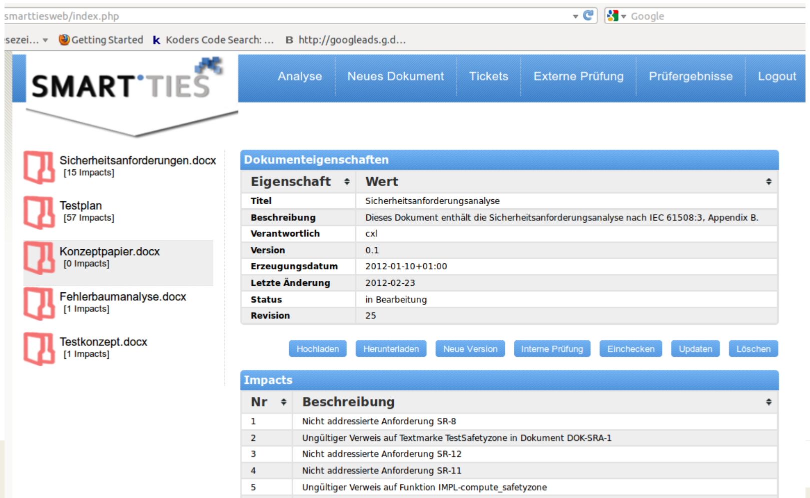
- Goal
  - Avoid having to reprove everything upon each change
  - Determine proofs affected by changes in source code and/or annotations

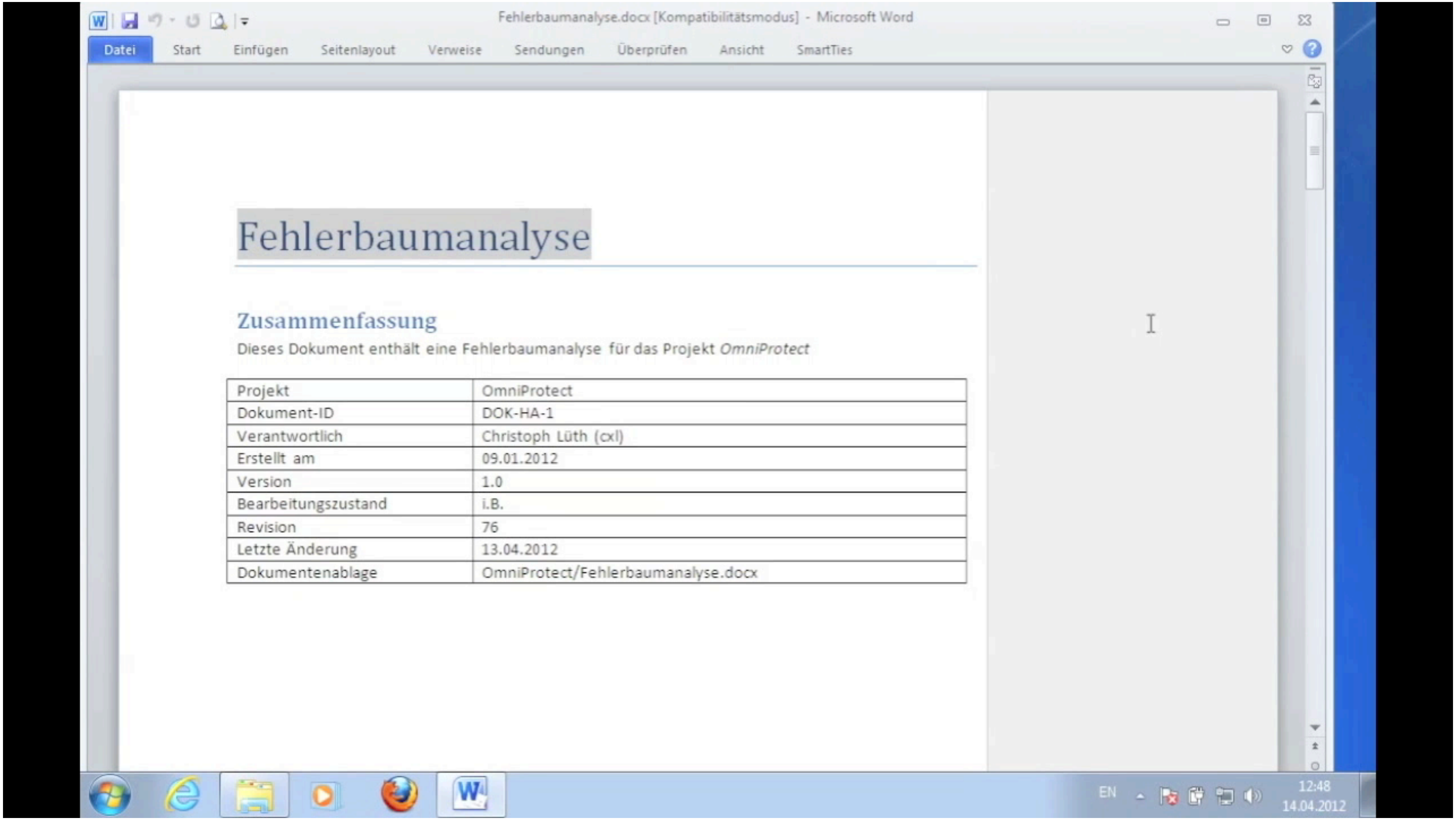


Application: SmartTies

- Development of Safety-Critical Systems (e.g. IEC 61508) demands a variety of documents
  - Concept Paper
  - Software Failure Modes and Effects Analysis
  - Safety Requirement Specification
  - Test Plans (tables)
  - Test Suites
  - Implementation
- SmartTies tool aims to maintain these documents in a consistent way







Fehlerbaumanalyse

Zusammenfassung

Dieses Dokument enthält eine Fehlerbaumanalyse für das Projekt *OmniProtect*

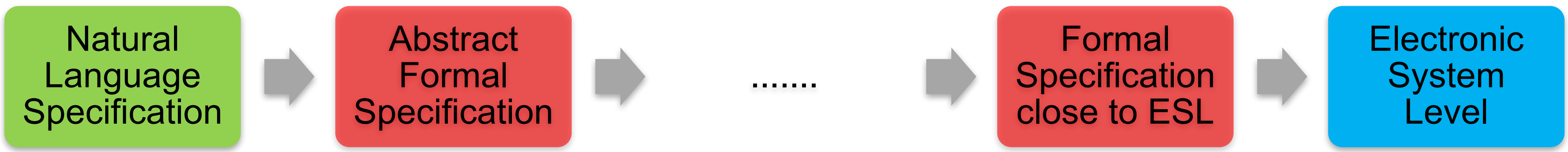
Projekt	OmniProtect
Dokument-ID	DOK-HA-1
Verantwortlich	Christoph Lüth (cxl)
Erstellt am	09.01.2012
Version	1.0
Bearbeitungszustand	i.B.
Revision	76
Letzte Änderung	13.04.2012
Dokumentenablage	OmniProtect/Fehlerbaumanalyse.docx

Application: Specific (BMBF 2013 – 2016)

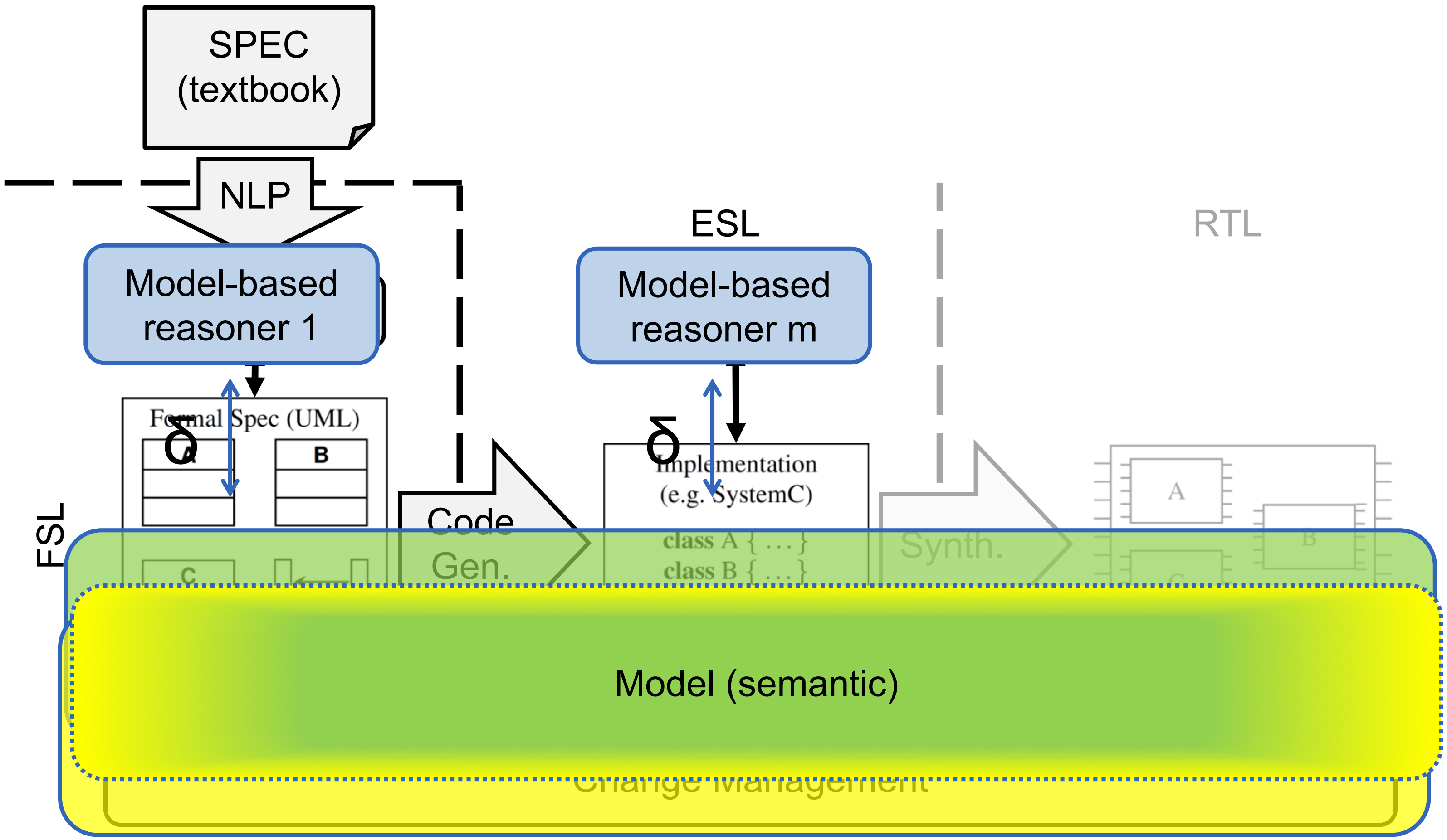
- Specific:

  Quality-Driven Design Flow using Formal Specification and Functional Change Management

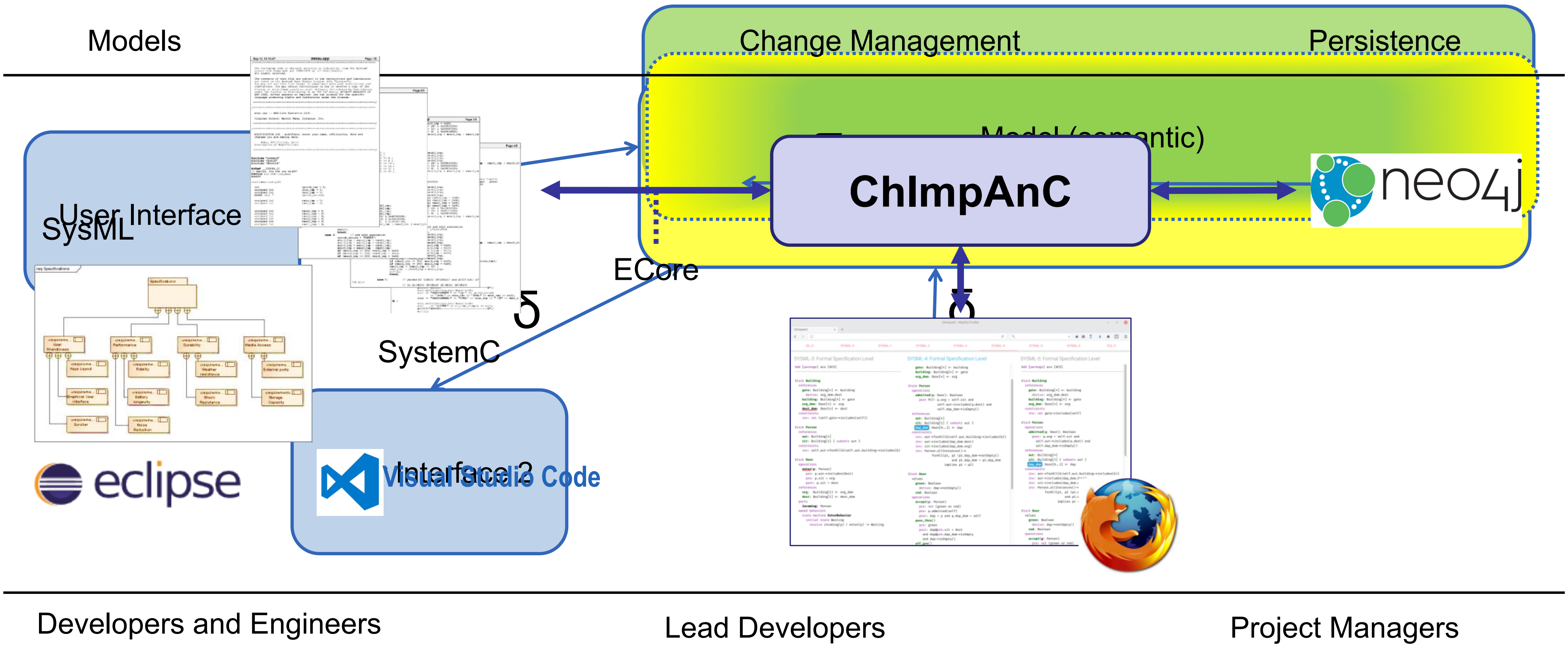
- Support and maintain system development from natural language specifications down to System Level



SPECifIC Design Flow

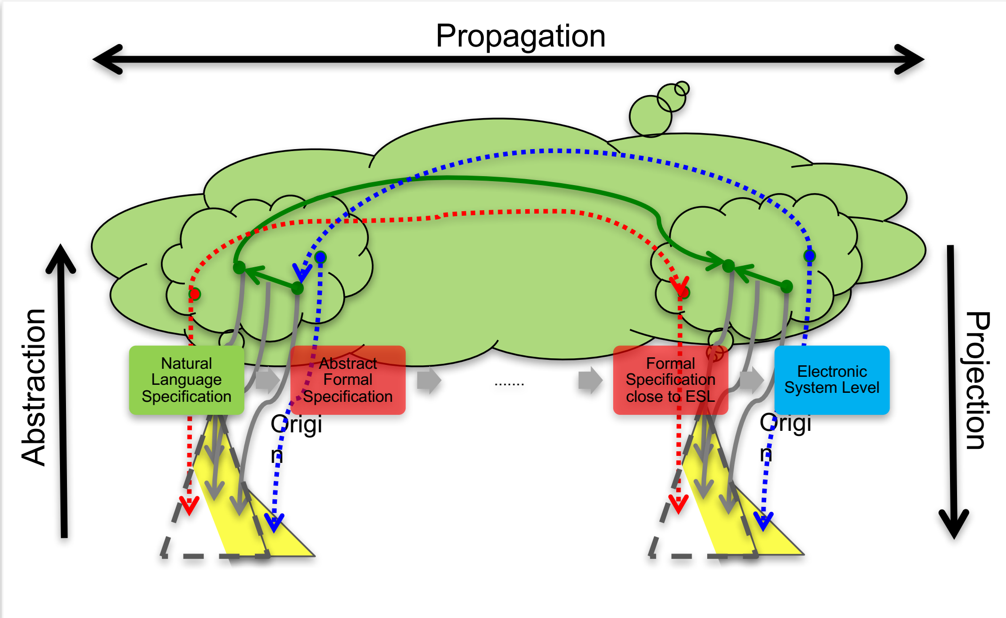


CM Tool Architecture



Tool Support for Change Management

- **Explicit semantics** approach:
  - represent syntax and semantics
  - Uniform representation by **typed graphs**
  - Graphs stored in **neo4j** graph database
- Change impact analysis:
  - Semantic difference analysis
  - Analysis and propagation of changes
  - Implemented **natively** in neo4j
- Enhancement of work in FormalSafe
  - but different implementation



Still right methodology, now also scalable, robust

SHIP

Project SHIP (BMBF 2011 – 2013)

SHIP: Semantic Integration of Heterogeneous Processes

Research question

- How to orchestrate individual, highly specialized systems
  - Individual systems, with individual process and data models
  - Individual systems operating on different abstraction levels
  - Combination of process-oriented view with non-trivial data models

Application: Assistance Processes @ Home

- Automate assistance for specific activities of daily living
  - Cooking, reading, dressing up, …
  - Reduce energy consumption, Comfort
  - Medical assistance, Safety
- Flexible, adaptive, interruptible
- Many persons, many goals
- Many assistance processes simultaneously
- **How to achieve robustness and safety?**

Development of an
ontology-based process description language (SHIP)
to ***orchestrate*** *and* ***monitor*** *environment and devices*

SHIP (BMBF 2011 – 2013)

Assistance Processes

Abox updates

OWL Ontologies Modelling BAALL and Status

Propagation to handle ramification problem via causal relationships

Actions as Abox updates

SHIP Process Language

Dynamic Description Logic

Monitoring via LTL over OWL Queries

User Interfaces

δ

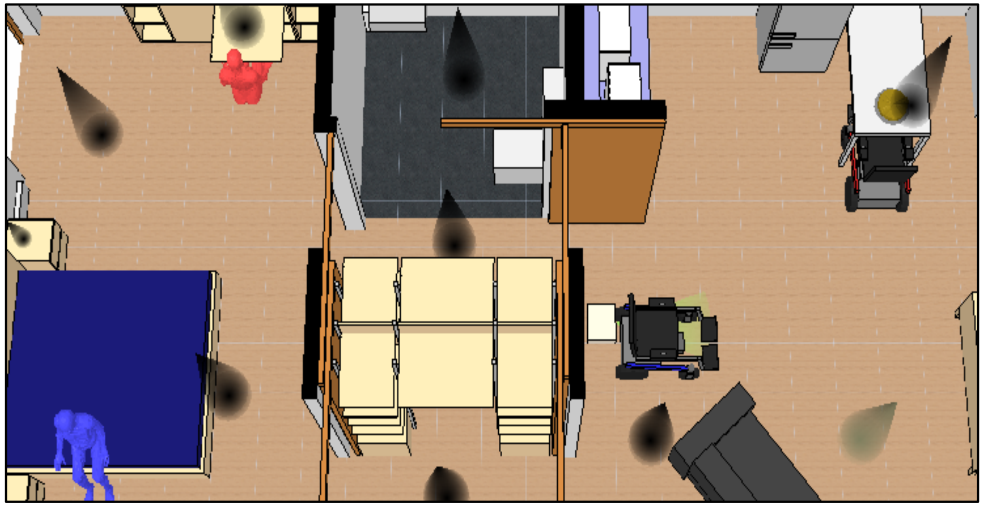
π

Model (semantic)

δ

Model-based reasoner

Application Example



- Multiple persons, multiple automatic wheelchairs
- Organisational Layer as Assistance Process
  - Handle transportation requests from individuals until completion
  - open blocking doors, illuminate dark portions of the path
  - Avoid wheelchairs hindering each other

Application: Coordinate Rolland and AILA

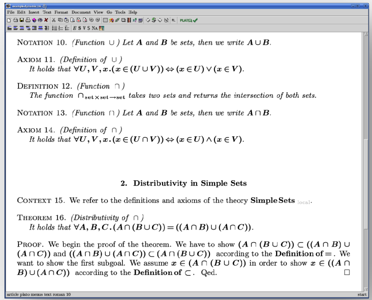
Common scenario to demonstrate work from CAPIO and SHIP

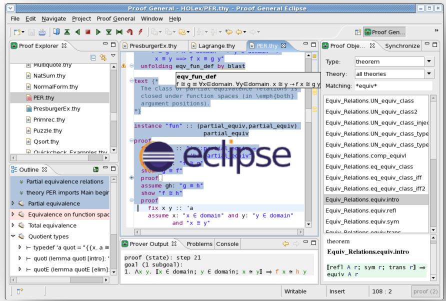
Person wearing an exoskeleton wants a scarf. Rolland and AILA are used to bring the scarf to the person.
The person assists AILA in grabbing the scarf using the exoskeleton to remotely control AILA.

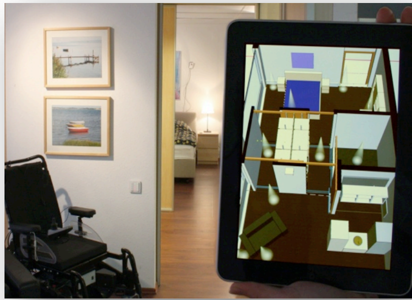
SHIP Tool is used to

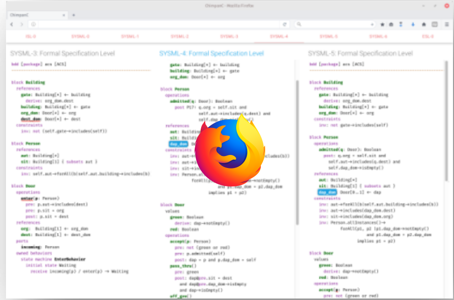
- receive requests
- know where the scarf is
- indicate the right shelf to AILA by blinking light
- send Rolland to the pickup position and then to the person
- control light and door during rides (as usual)

What this talk was about so far

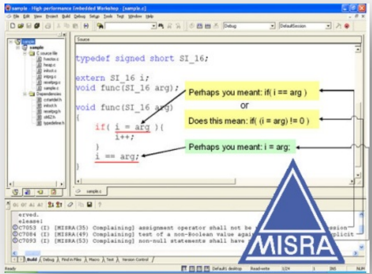


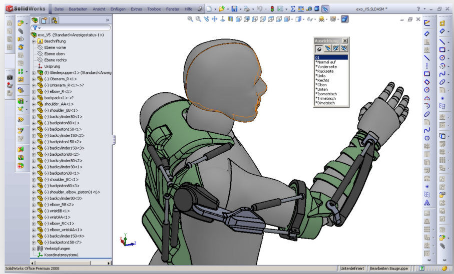


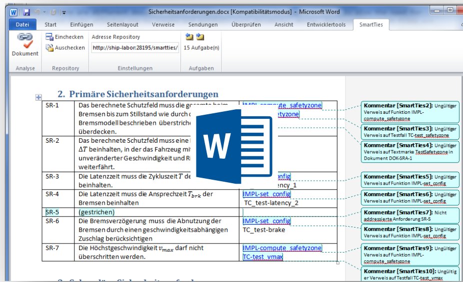




Familiar/intuitive user interfaces







How to enable

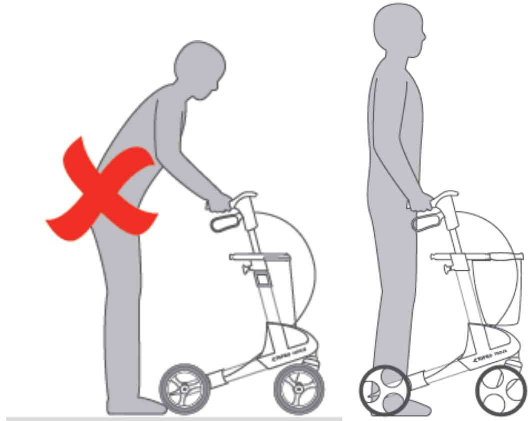
- intelligent assistance based on rule based/formal logical reasoning directly
- through/inside familiar/intuitive user interfaces

Current BAALL Projects

Project ModESt (1.2017– 12.2019)

- Walker-Module for posture recognition and fall prevention
- Problem:
  - the correct use must be properly learned and practiced continuously
- Goal:
  - is to prevent latent poor postures and risk of falls.
- Solution:
  - distance sensors based with software to recognize poor postures
  - low-threshold feedback for posture corrections
  - in an electronic box integrated into the frame of the walker
- Funded by BMBF in program “Human-Machine-Interaction” in the scheme ”Initiatives for SMEs”

Christian Mandel



Federal Ministry of Education and Research

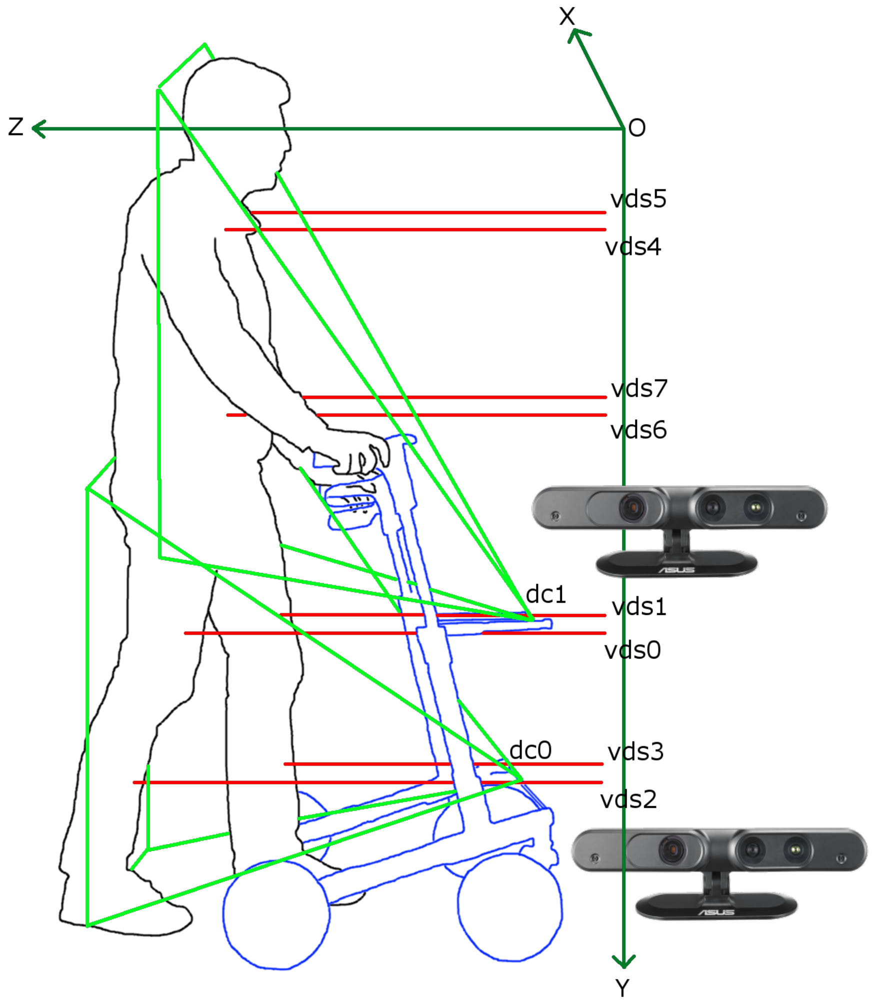
Project Partners

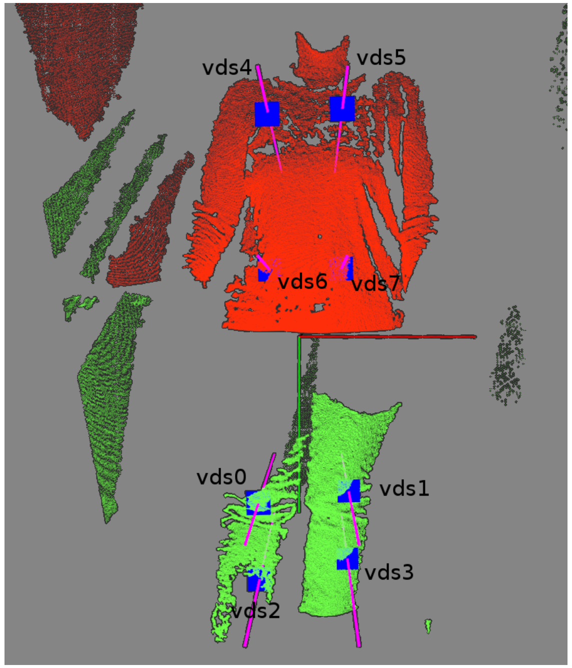
BUDELMANN Elektronik

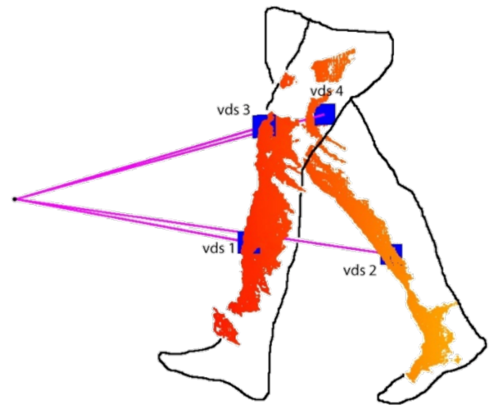
GESUNDHEIT NORD KLINIKVERBUND BREMEN

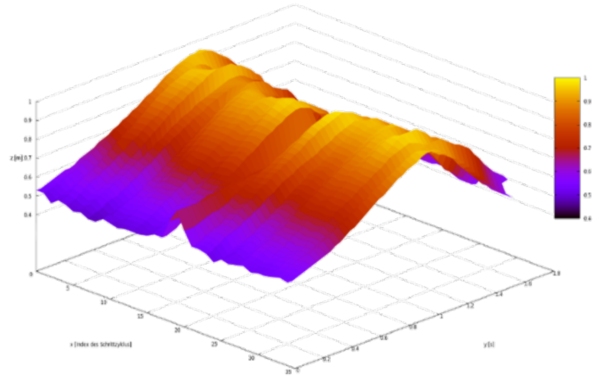
TOPRO

DFKI

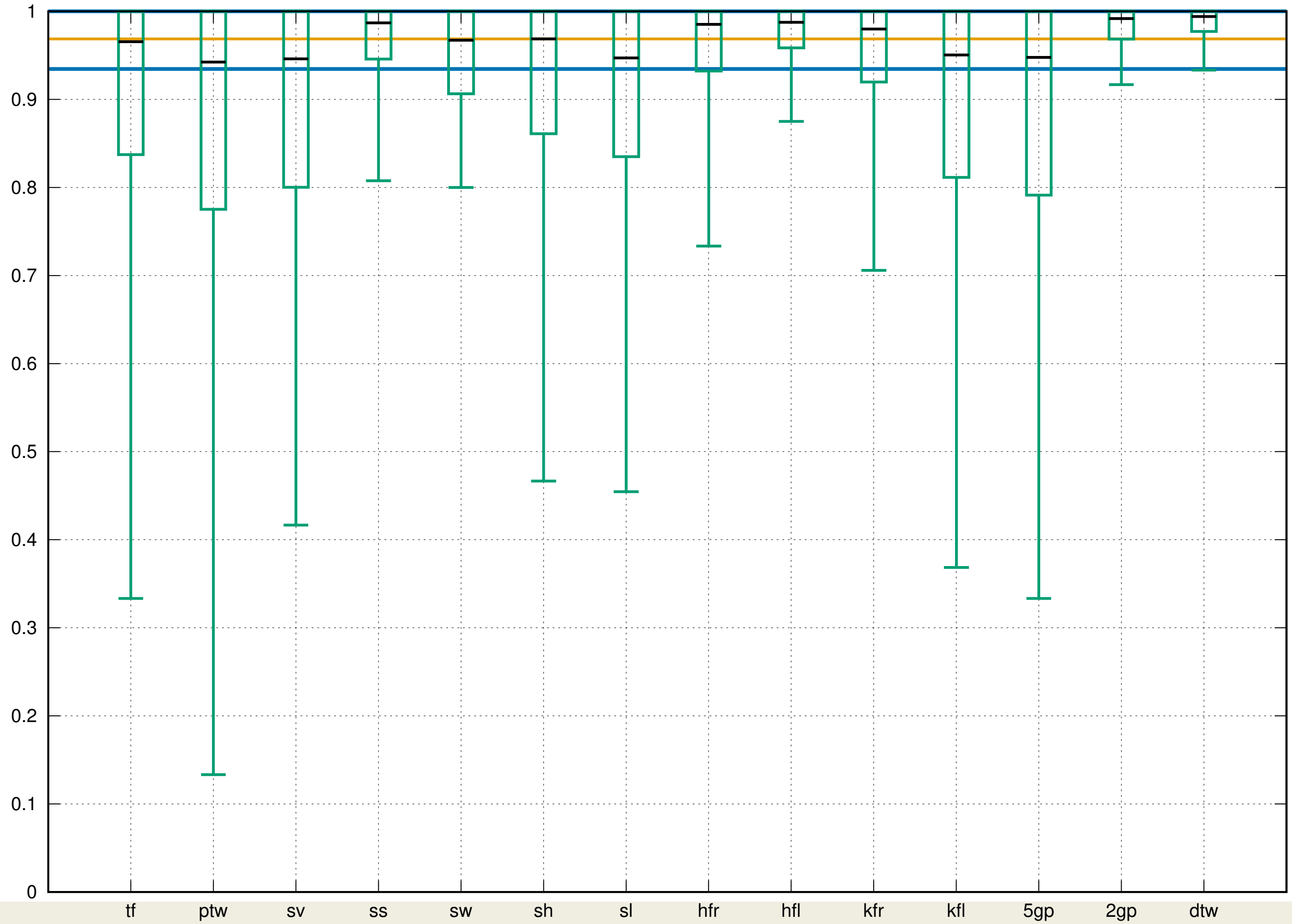

X
Z
O
vds5
vds4
vds7
vds6
dc1
vds1
vds0
dc0
vds3
vds2
Y


vds4
vds5
vds6
vds7
vds0
vds1
vds3
vds2


vds 3
vds 4
vds 1
vds 2



Learned Classifier for typical geriatric gait properties



- 2 gait pattern (2gp)
- 5 gait pattern (5gp)
- position to walker (ptw)
- distance to walker (dtw)
- hip flection left (hfl)
- hip flection right (hfr)
- knee flection left (kfl)
- knee flection right (kfr)
- torso flection (tf)
- stride symmetry (ss)
- stride width (sw)
- stride variability (sv)
- stride length (sl)
- stride height (sh)

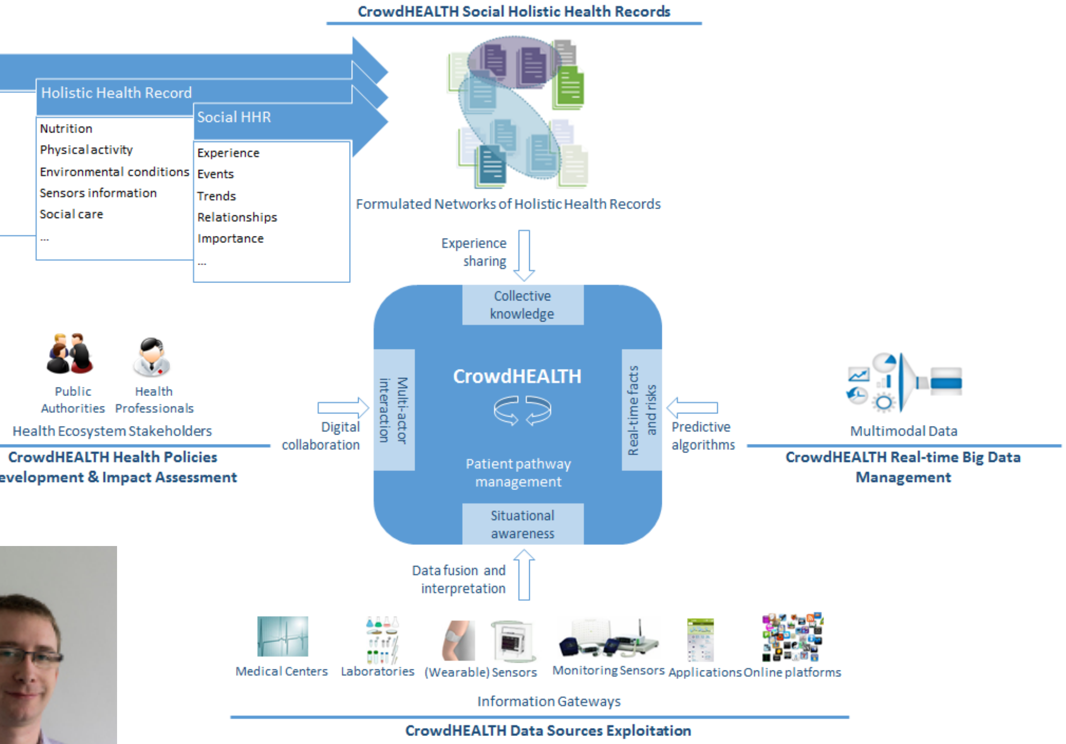
Project CrowdHEALTH (4.2017 - 3.2020)
Collective wisdom driving public health policies

Decision support for public health policy makers based on integrated heterogeneous health data

Funded by EU in the *Call H2020-SC1-2016-CNECT „Personalised Medicine“ through Grant 727560*

19 Partners

Jan Janssen



Project SMILE 4.2017 – 3.2020

… For an increasing proportion of women in Computer Science professions

Federal Ministry of Education and Research

„Smart Environments as a context of motivating learning opportunities for girls for a growing proportion of computer scientists by involving teachers and parents“

- *For female scholars starting age of 12*
- *Awake and preserve interest in Computer Science*
- *Teach Computer Science concepts and methods*

Funded by funded by the Federal Ministry for Education and Research within the support program for "Strategies for Achieving Equality of Opportunity for Women in Education and Research ("Success with MINT - New Opportunities for Women")" (FKZ 01FP1613)

Partners Universität Bremen, OFFIS, Hochschule für Angewandte Wissenschaften Hamburg – Hamburg University of Applied Sciences, DFKI, Carl von Ossietzky Universität Oldenburg

Anke Königschulte

Mazyar Seraj

smile IT SMART · FUTURE · ME

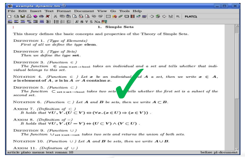
Summarizing …

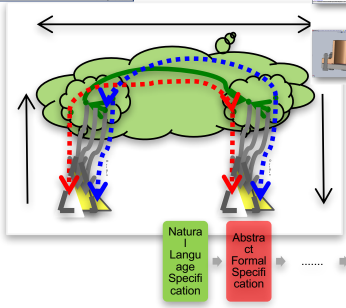
Formal Logic, Management of Change, Intuitive Interfaces in …

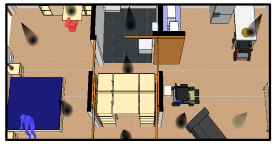
- Project OMEGA (SFB 2005-2007)
- Project FormalSafe (BMBF 2008-2010)
- Project SHIP (BMBF 2011-2013)
- Project Specific (BMBF 2013-2016)

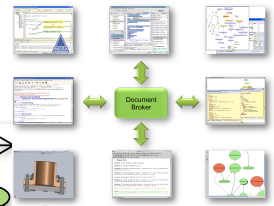
… and beyond

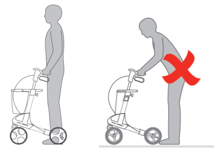
  - Project ModEST (BMBF 2017-2019)
  - Project CrowdHEALTH (EU 2017-2020)
  - Project SMILE (BMBF 2017-2020)











CrowdHEALTH

smile
SMART · FUTURE · ME

Relevant publications on the main topic

Serge Autexier. *Similarity-Based Diff, Three-Way-Diff and Merge*, International Journal of Software and Informatics (IJSI) 9(2), August, 2015

Serge Autexier and Dieter Hutter. *SHIP - A Logic-Based Language and Tool to Program Smart Environments*, In Moreno Falaschi (Ed) Post-Proceedings 25th International Conference on Logic-Based Program Synthesis and Transformation (LOPSTR 2015), LNCS, Springer, October, 2015

Serge Autexier, Dieter Hutter, and Christoph Stahl. *An Implementation, Execution and Simulation Platform for Processes in Heterogeneous Smart Environments*, In Juan Carlos Augusto and Reiner Wichert (Ed) Fourth International Joint Conference on Ambient Intelligence, LNCS, Dublin, Ireland, Springer , December, 2013

Serge Autexier, Dieter Hutter, Christian Mandel, and Christoph Stahl. *SHIP-Tool Live: Orchestrating the Activities in the Bremen Ambient Assisted Living Lab (Demo)*, In Juan Carlos Augusto and Reiner Wichert (Ed) Fourth International Joint Conference on Ambient Intelligence, LNCS, Dublin, Ireland, Springer , December, 2013

Serge Autexier, Dominik Dietrich, Dieter Hutter, Christoph Lüth, and Christian Maeder. *SmartTies - Management of Safety-Critical Developments*, In Tiziana Margaria and Bernhard Steffen (Ed) Proceedings 5th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation (ISoLa'12), LNCS, Amirandes, Heraclion, Crete, Springer, October, 2012

Serge Autexier and Dominik Dietrich and Marvin Schiller. *Towards an Intelligent Tutor for Mathematical Proofs*, In Pedro Quaresma and Ralph-Johan Back (Ed) Proceedings First Workshop on CTP Components for Educational Software (THedu'11), Vol. 79, EPTCS, p. 1-28, February, 2012

Serge Autexier, Catalin David, Dominik Dietrich, Michael Kohlhase, and Vyacheslav Zholudev . *Workflows for the Management of Change in Science, Technologies, Engineering and Mathematics* , In James H. Davenport, William Farmer, Florian Rabe, and Joseph Urban (Ed) Proceedings of Calculemus/MKM 2011, Nr. 6824 of LNAI, p. 164-179, Springer-Verlag Berlin Heidelberg, July, 2011

Serge Autexier and Normen Müller. *Semantics-Based Change Impact Analysis for Heterogeneous Collections of Documents*, In Michael Gormish and Rolf Ingold (Ed) Proceedings of 10th ACM Symposium on Document Engineering (DocEng2010), Manchester, UK, September, 2010

Serge Autexier and Christoph Lüth. *Adding Change Impact Analysis to the Formal Verification of C Programs*, In Dominique Méry and Stephan Merz (Ed) Proceedings 8th International Conference on integrated Formal Methods (IFM2010), LNCS, Nancy, France, Springer, October, 2010

Serge Autexier, Christoph Benzmueller, Dominik Dietrich, and Marc Wagner. *Organisation, Transformation, and Propagation of Mathematical Knowledge in OMEGA*, Journal Mathematics in Computer Science, 2008

Christoph Benzmüller, Dominik Dietrich, Marvin Schiller, Serge Autexier. *Deep Inference for Automated Proof Tutoring?*, In Joachim Hertzberg, Michael Beetz, Roman Englert (Ed) KI 2007: Advances in Artificial Intelligence, LNAI, Springer, September, 2007

Serge Autexier, Armin Fiedler, Thomas Neumann, and Marc Wagner. *Supporting User-Defined Notations when Integrating Scientific Text-Editors with Proof Assistance* , In Manuel Kauers, Manfred Kerber, Robert Miner, and Wolfgang Windsteiger (Ed) Towards Mechanized Mathematical Assistants, LNAI, Springer, June, 2007

Serge Autexier and Claudio Sacerdoti-Coen. *A Formal Correspondence between OMDoc with Alternative Proofs and the LambdabarMuMutilde-Calculus*, In Jon Borwein and Bill Farmer (Ed) Proceedings of MKM'06, Vol. 4108, LNAI, p. 67-81, Springer, August, 2006

Serge Autexier, Christoph Benzmüller, Armin Fiedler, Henri Lesourd. *Integrating Proof Assistants as Reasoning and Verification Tools into a Scientific WYSIWIG Editor*, In David Aspinall and Christoph Lüth (Ed) Proceedings of UITP'05, ENTCS, January, 2006

Jörg Siekmann, Christoph Benzmüller, and Serge Autexier. *Computer Supported Mathematics with OMEGA*, Journal of Applied Logic, special issue on Mathematics Assistance Systems 4(4), December, 2006

Serge Autexier, Christoph Benzmüller, Dominik Dietrich, Andreas Meier, and Claus-Peter Wirth. *A Generic Modular Data Structure for Proof Attempts Alternating on Ideas and Granularity*, In Kohlhase, Michael (Ed) Proceedings of MKM'05, Vol. 3863, LNAI, IUB Bremen, Germany, Springer, January, 2006

Serge Autexier and Armin Fiedler. *Textbook Proofs Meet Formal Logic - The Problem of Underspecification and Granularity*, In Kohlhase, Michael (Ed) Proceedings of MKM'05, Vol. 3863, LNAI, IUB Bremen, Germany, Springer, January, 2006

Thank you.