Change Management for Heterogeneous Development Graphs

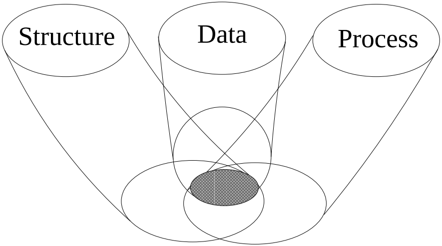
Till Mossakowski
joint work with Serge Autexier, Dominik Dietrich and Dieter Hutter

German Research Center for Artificial Intelligence (DFKI GmbH)
Bremen, Germany

January 9, 2011

Outline

Heterogeneous Specifications: Motivation



Desirable for complex systems:

- multiple viewpoints using different formalisms
- change of formalism during development
- multiple, special purpose provers

Hence, heterogeneous specifications are needed.

The Heterogeneous Tool Set (Hets)

Isabelle:

Paradigm shift from ad-hoc to generic treatment
of proof rules and unification.

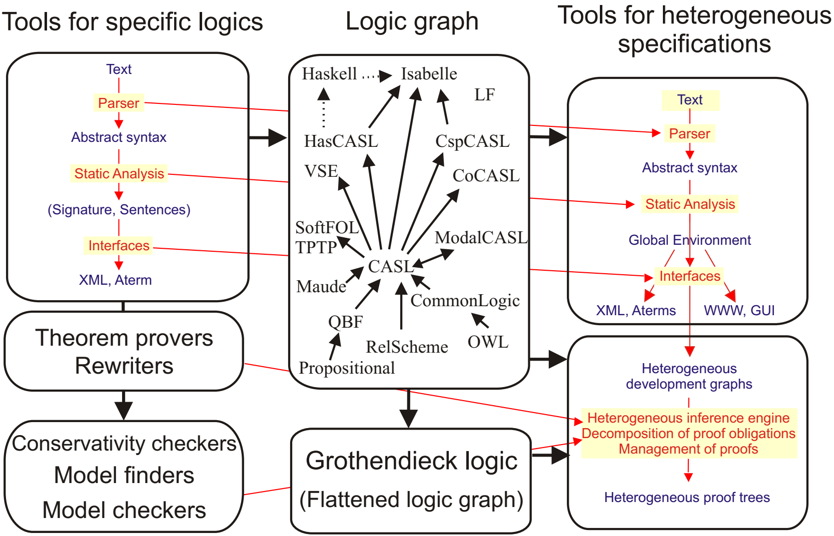
HETS:

Paradigm shift from ad-hoc to generic treatment
of structuring-in-the-large and heterogeneous integration.

Further strengths of HETS:

- flexible selection of tool-supported sublangauges suitable for subproblems
- systematic connection of new formalisms to tools via translations
- logic translations are first-class citizens
- easy plug-in of new formalisms and translations

Architecture of the heterogeneous tool set Hets



Definition

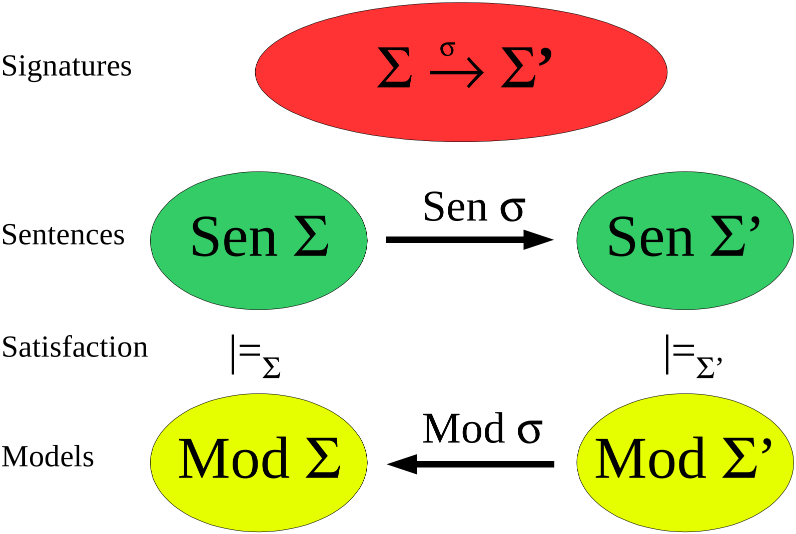
An *institution* $\mathcal{I}$ consists of:

- a category $\mathbf{Sign}_{\mathcal{I}}$ of *signatures*;
- a functor $\mathbf{Sen}_{\mathcal{I}}\colon \mathbf{Sign}_{\mathcal{I}} \to \mathbf{Set}$,giving a set $\mathbf{Sen}(\Sigma)$ of Σ-*sentences* for each signature $\Sigma \in |\mathbf{Sign}_{\mathcal{I}}|$, and a function $\mathbf{Sen}(\sigma)\colon \mathbf{Sen}(\Sigma) \to \mathbf{Sen}(\Sigma')$ that yields σ-*translation* of Σ-sentences to Σ'-sentences for each signature morphism $\sigma\colon \Sigma \to \Sigma'$;
- a functor $\mathbf{Mod}_{\mathcal{I}}\colon \mathbf{Sign}_{\mathcal{I}}^{op} \to \mathbf{Set}$,giving a set $\mathbf{Mod}(\Sigma)$ of Σ-*models* for each signature $\Sigma \in |\mathbf{Sign}_{\mathcal{I}}|$, and a functor $\mathbf{Mod}(\sigma)\colon \mathbf{Mod}(\Sigma') \to \mathbf{Mod}(\Sigma)$, denoted by $\_|_{\sigma}$, that yields σ-*reducts* of Σ'-models for each signature morphism $\sigma\colon \Sigma \to \Sigma'$; and
- for each $\Sigma \in |\mathbf{Sign}_{\mathcal{I}}|$, a *satisfaction relation* $\models_{\mathcal{I},\Sigma} \ \subseteq \mathbf{Mod}_{\mathcal{I}}(\Sigma) \times \mathbf{Sen}_{\mathcal{I}}(\Sigma)$

such that for any signature morphism $\sigma\colon \Sigma \to \Sigma'$, Σ-sentence $\varphi \in \mathbf{Sen}_{\mathcal{I}}(\Sigma)$ and Σ'-model $M' \in \mathbf{Mod}_{\mathcal{I}}(\Sigma')$:

$$M' \models_{\mathcal{I},\Sigma'} \sigma(\varphi) \iff M'|_{\sigma} \models_{\mathcal{I},\Sigma} \varphi \qquad [\mathit{Satisfaction\ condition}]$$

Institutions



Logics currently supported by Hets

general-purpose logics
: Propositional, QBF, SoftFOL, CASL (FOL), HasCASL (HOL)

logical frameworks
: Isabelle, LF, DFOL

ontologies and constraint languages
: OWL, CommonLogic, RelScheme, ConstraintCASL

reactive systems
: CspCASL, CoCASL, ModalCASL, Maude

programming languages
: Haskell, VSE

logics of specific tools
: Reduce, DMU (CATIA)

Definition

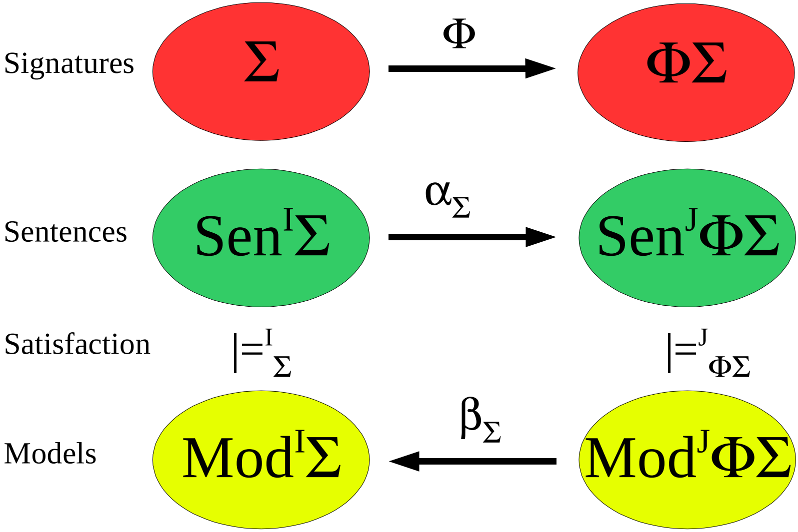
An *institution comorphism* $\rho \colon \mathcal{I} \to \mathcal{I}'$ consists of:

- a functor $\rho^{Sign} \colon \mathbf{Sign} \to \mathbf{Sign}'$;
- a natural transformation $\rho^{Sen} \colon \mathbf{Sen} \to \mathbf{Sen}' \circ \rho^{Sign}$, that is, a family of functions $\rho^{Sen}_\Sigma \colon \mathbf{Sen}(\Sigma) \to \mathbf{Sen}'(\rho^{Sign}(\Sigma))$, natural in $\Sigma \in |\mathbf{Sign}|$; and
- a natural transformation $\rho^{Mod} \colon \mathbf{Mod}' \circ (\rho^{Sign})^{op} \to \mathbf{Mod}$, that is, a family of functions $\rho^{Mod}_\Sigma \colon \mathbf{Mod}'(\rho^{Sign}(\Sigma)) \to \mathbf{Mod}(\Sigma)$, natural in $\Sigma \in |\mathbf{Sign}|$,

such that for any $\Sigma \in |\mathbf{Sign}|$, the translations $\rho^{Sen}_\Sigma \colon \mathbf{Sen}(\Sigma) \to \mathbf{Sen}'(\rho^{Sign}(\Sigma))$ of sentences and $\rho^{Mod}_\Sigma \colon \mathbf{Mod}'(\rho^{Sign}(\Sigma)) \to \mathbf{Mod}(\Sigma)$ of models preserve the satisfaction relation, i.e., for any $\varphi \in \mathbf{Sen}(\Sigma)$ and $M' \in \mathbf{Mod}'(\rho^{Sign}(\Sigma))$:

$$M' \models'_{\rho^{Sign}(\Sigma)} \rho^{Sen}_\Sigma(\varphi) \iff \rho^{Mod}_\Sigma(M') \models_\Sigma \varphi \qquad [\mathit{Satisfaction\ condition}]$$

Institution comorphisms



Definition

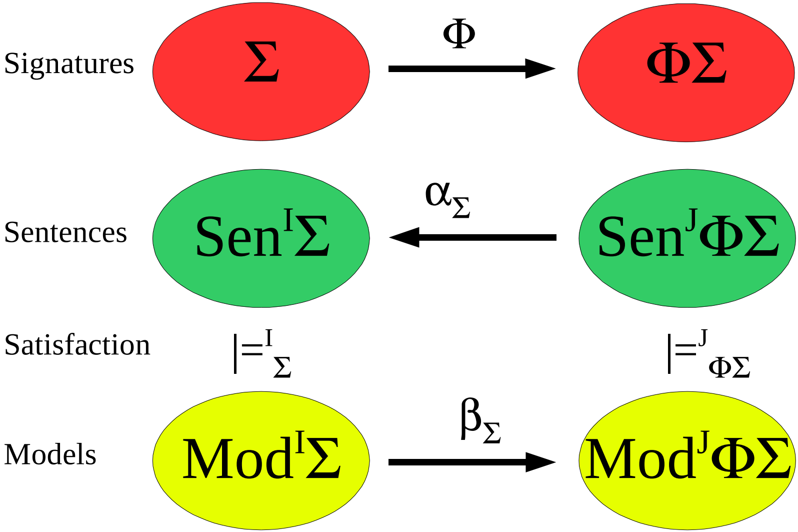
Let $\mathcal{I}$ and $\mathcal{I}'$ be institutions. An *institution morphism* $\mu\colon \mathcal{I} \to \mathcal{I}'$ consists of:

- a functor $\mu^{Sign}\colon \mathbf{Sign} \to \mathbf{Sign}'$;
- a natural transformation $\mu^{Sen}\colon \mathbf{Sen}' \circ \mu^{Sign} \to \mathbf{Sen}$, that is, a family of functions $\mu^{Sen}_{\Sigma}\colon \mathbf{Sen}'(\mu^{Sign}(\Sigma)) \to \mathbf{Sen}(\Sigma)$, natural in $\Sigma \in |\mathbf{Sign}|$; and
- a natural transformation $\mu^{Mod}\colon \mathbf{Mod} \to \mathbf{Mod}' \circ (\mu^{Sign})^{op}$, that is, a family of functions $\mu^{Mod}_{\Sigma}\colon \mathbf{Mod}(\Sigma) \to \mathbf{Mod}'(\mu^{Sign}(\Sigma))$, natural in $\Sigma \in |\mathbf{Sign}|$,

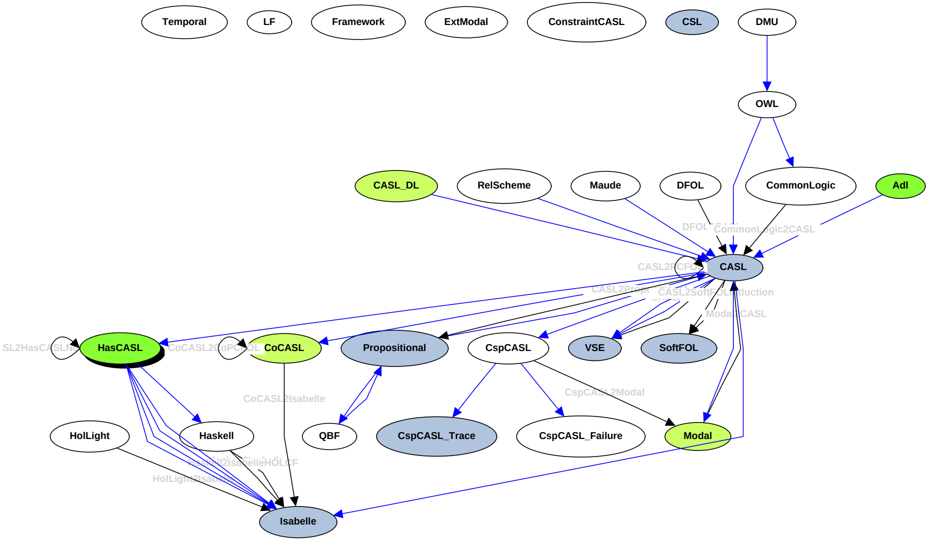
such that for any signature $\Sigma \in |\mathbf{Sign}|$, the translations $\mu^{Sen}_{\Sigma}\colon \mathbf{Sen}'(\rho^{Sign}(\Sigma)) \to \mathbf{Sen}(\Sigma)$ of sentences and $\mu^{Mod}_{\Sigma}\colon \mathbf{Mod}(\Sigma) \to \mathbf{Mod}'(\rho^{Sign}(\Sigma))$ of models preserve the satisfaction relation, i.e., for any $\varphi' \in \mathbf{Sen}'(\mu^{Sign}(\Sigma))$ and $M \in \mathbf{Mod}(\Sigma)$:

$$M \models_{\Sigma} \mu^{Sen}_{\Sigma}(\varphi') \iff \mu^{Mod}_{\Sigma}(M) \models'_{\mu^{Sign}(\Sigma)} \varphi' \qquad [\mathit{Satisfaction\ condition}]$$

Institution morphisms



The Current Hets Logic Graph



Syntax of Structured Specifications

SP ::=	BASIC-SPEC	basic specification
	| SP then SP	extension
	| SP and SP	union
	| SP with SYMBOL-MAP	renaming
	| SP hide SYMBOLS	hiding
	| SPEC-NAME [PARAM*]	reference to named spec

LIBRARY-ITEM ::=

spec SPEC-NAME [PARAM*] = SP end   name a spec

| view VIEW-NAME : SP to SP = SYMBOL-MAP end

refinement between specifications

[Mossakowski/Haxthausen/Sannella/Tarlecki 2008]
[Baumeister/Cerioli/Haxthausen/Mossakowski/Mosses/Sannella/Tarlecki 2004]

Syntax of Structured Specifications

SP ::= BASIC-SPEC
| SP then SP
| SP and SP
| SP with SYMBOL-MAP
| SP hide SYMBOLS
| SPEC-NAME [PARAM*]

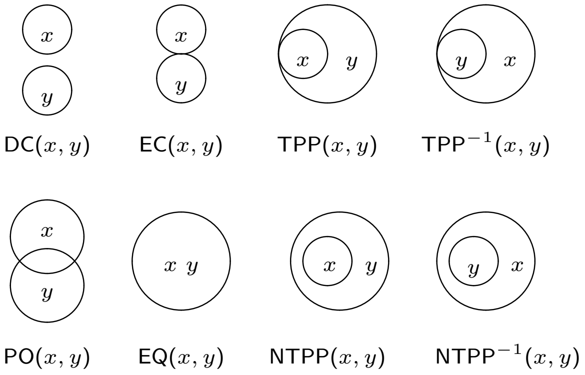
LIBRARY-ITEM ::=
spec SPEC-NAME [PARAM*] = SP end
| view VIEW-NAME : SP to SP = SYMBOL-MAP end

Syntax of Heterogeneous Specifications

```
SP ::=  BASIC-SPEC                 | logic LOGIC-NAME : {SP}
      | SP then SP
      | SP and SP
      | SP with SYMBOL-MAP         | SP with logic COMORPHISM
      | SP hide SYMBOLS            | SP hide logic MORPHISM
      | SPEC-NAME [PARAM*]

LIBRARY-ITEM ::=
    spec SPEC-NAME [PARAM*] = SP end
  | view VIEW-NAME : SP to SP = SYMBOL-MAP end
  | view VIEW-NAME : SP to SP = SYMBOL-MAP, COMORPHISM
  | logic LOGIC-NAME
```

[Mossakowski 2005]

Example: the Region Connection Calculus



RCC8 forms a relation algebra and is used for qualitative constraint reasoning about spatial configurations.

A Heterogeneous Refinement

Question: is the composition table correct w.r.t. the interpretation of RCC regions as closed unit balls in an arbitrary metric space?

Example (Closed Balls as Regions)

view $\mathrm{RCC\_FO\_{IN}\_CLOSEDBALL}$:
 $\mathrm{RCC\_FO}$ **to**
 $\{\mathrm{CLOSEDBALL}$
 then %def
  pred $\_\_C\_\_ : \mathit{ClosedBall} \times \mathit{ClosedBall}$
  $\forall\, a, b : \mathit{ClosedBall}$
  $\bullet\ a\ C\ b \Leftrightarrow \exists\, x : \mathit{Space} \bullet \mathit{covers}\,(a, x) \wedge \mathit{covers}\,(b, x)$
 $\} = \mathit{Region} \mapsto \mathit{ClosedBall}$, **logic** $\mathrm{CASL} \rightarrow \mathrm{HASCASL}$

Heterogeneous Development Graphs

Heterogeneous structured specifications are mapped into heterogeneous development graphs:

- nodes correspond to individual specification modules
  - each node is equipped with a signature and a set of axioms
  - semantics: class of Σ-models satisfying axioms
- definition links correspond to imports of modules
  - each link is equipped with a (Grothendieck) signature morphism
  - semantics: models (when reduced) also have so satisfy imported constraints
- theorem links express proof obligations
  - semantics: translated source theory is provable in target theory

Theorem. There is a proof calculus for heterogeneous development graphs that is sound, and (relative to an oracle for conservative extensions) also complete [Mossakowski MFCS 2002]

Change Management: Motivation

- evolutionary formal development approach
- not only implementations change, but also specifications!
- change management can reduce the need for time-consuming proof replay
- tracking of effects of changes

Institutions with Pre-Signatures

[AutexierHutterMossakowski2010]

- change management is based on manipulation of individual symbols
- problem: institutions provide just a category of signatures
- solution: make signatures behave more set-like

Related work

- inclusive institutions (used in the OBJ/CaféOBJ community)
- institutions with qualified symbols (uxed in the CASL semantics)

Both do not directly support the assembly of signatures from local symbols

Institutions with Pre-Signatures

Definition

An *institution with pre-signatures* is an institution equipped with an embedding $|\_| : \mathbf{Sign} \to \mathbf{Set}$, the *symbol functor*, and a map $sym\colon \bigcup_{\Sigma \in |\mathbf{Sign}|} \mathbf{Sen}(\Sigma) \to |\mathbf{Set}|$, such that

$$\varphi \in \mathbf{Sen}(\Sigma) \text{ iff } sym(\varphi) \subseteq |\Sigma|$$

for all $\varphi \in \bigcup_{\Sigma \in |\mathbf{Sign}|} \mathbf{Sen}(\Sigma)$. The map *sym* gives the set of symbols used in a sentence, and sentences are uniform in the sense that a well-formed sentence is well-formed over a certain signature iff its symbols belong to that signature. Moreover, we require that any inclusion $\iota\colon |\Sigma_1| \hookrightarrow |\Sigma_2|$ is a signature morphism (i.e., is in the image of $|\_|$).

Definition

A *pre-signature* Σ is a set, and a *pre-signature morphism* $\bar{\sigma}$ consists of a right-unique set of pairs $graph(\bar{\sigma})$ and a set $dom(\bar{\sigma})$, subject to the requirement that

$$dom(\bar{\sigma}) \subseteq def(\bar{\sigma}),$$

where

$$def(\bar{\sigma}) = \{x | \exists y.\ (x, y) \in graph(\bar{\sigma})\}.$$

We also define

$$codef(\bar{\sigma}) = \{y | \exists x.\ (x, y) \in graph(\bar{\sigma})\}.$$

We write $\bar{\sigma}(x) = y$ iff $(x, y) \in graph(\bar{\sigma})$, and $\bar{\sigma}(x) = \perp$ iff $x \notin def(\bar{\sigma})$.

Definition

Given a pre-signature morphism $\bar{\sigma}$ and a pre-signature Σ, define the induced function as

$$fun_\Sigma(\bar{\sigma}) = graph(\bar{\sigma})|_{|\Sigma|} \cup Id_{|\Sigma| \setminus def(\bar{\sigma}|_{|\Sigma|})},$$

where $graph(\bar{\sigma})$ is construed as a function and $\bar{\sigma}|_X$ denotes the restriction of $\bar{\sigma}$ to X.

Definition

A pre-signature morphism $\bar{\sigma}$ is *well-formed* wrt. a source signature Σ_1 and a target signature Σ_2, if $dom(\bar{\sigma}) \subseteq |\Sigma_1|$ and there exists a signature morphism $\sigma\colon \Sigma_1 \to \Sigma_2$ with $|\sigma| = fun_{|\Sigma_1|}(\bar{\sigma})$. In this case, σ is unique and is called *the signature morphism from Σ_1 to Σ_2 induced by $\bar{\sigma}$* and we will not distinguish between the σ and $\bar{\sigma}$ if Σ_1 and Σ_2 are clear from the context.

Definition

Composition of pre-signature morphisms is defined by

$$\bar{\sigma}_2 \circ \bar{\sigma}_1(x) := \begin{cases} \bar{\sigma}_2(\bar{\sigma}_1(x)) & \text{if } x \in \mathit{def}(\bar{\sigma}_1) \text{ and } \bar{\sigma}_1(x) \in \mathit{def}(\bar{\sigma}_2) \\ \bar{\sigma}_1(x) & \text{if } x \in \mathit{def}(\bar{\sigma}_1) \text{ and } \bar{\sigma}_1(x) \notin \mathit{def}(\bar{\sigma}_2) \\ \bar{\sigma}_2(x) & \text{if } x \notin \mathit{def}(\bar{\sigma}_1) \text{ and } x \in \mathit{def}(\bar{\sigma}_2) \\ \perp & \text{otherwise} \end{cases}$$

$$\mathit{dom}(\bar{\sigma}_2 \circ \bar{\sigma}_1) := \mathit{dom}(\bar{\sigma}_1)$$

The definition of composition ensures the following properties:

Theorem

Composition of pre-signature morphisms is associative.

Theorem

If $codef(fun_{\Sigma_1}(\bar{\sigma}_1)) \subseteq |\Sigma_2|$, *then*

$$fun_{\Sigma_2}(\bar{\sigma}_2) \circ fun_{\Sigma_1}(\bar{\sigma}_1) = fun_{\Sigma_1}(\bar{\sigma}_2 \circ \bar{\sigma}_1)$$

Definition

A pre-signature $\bar{\Sigma}$ is *well-formed*, if there exists a signature Σ with $|\Sigma| = \bar{\Sigma}$. Since $|\_|$ is an embedding, if Σ exists, it is uniquely determined by $\bar{\Sigma}$. Hence, in the sequel, we often will not distinguish between (a well-formed) $\bar{\Sigma}$ and Σ.

Definition

Every signature morphism $\sigma : \Sigma \to \Sigma'$ induces a pre-signature morphism $\bar{\sigma}$ defined by

$$dom(\bar{\sigma}) := \{x \in \Sigma \mid \sigma(x) \neq x\} \text{ and } \bar{\sigma}(x) := \begin{cases} |\sigma|(x) & \text{if } x \in dom(\bar{\sigma}) \\ \bot & \text{otherwise} \end{cases}$$

Definition

An institution comorphism $\rho : \mathcal{I} \to \mathcal{I}'$ is *modular* if there is

- $\rho^{PreSign}$ mapping pre-signatures to pre-signatures and pre-signature morphisms to pre-signature morphisms[a] and
- $\rho^{PreSen} : \bigcup_{\Sigma \in |\mathbf{Sign}|} \to \bigcup_{\Sigma \in |\mathbf{Sign}'|}$

satisfying the following conditions:

- $|\rho^{Sign}(\Sigma)| = \rho^{PreSign}(|\Sigma|)$,
- $|\rho^{Sign}(\sigma)| = \rho^{PreSign}(|\sigma|)$,
- $\rho^{PreSign}(fun_\Sigma(\bar{\sigma})) = fun_{\rho(\Sigma)}(\rho^{PreSign}(\bar{\sigma}))$,
- for each signature morphism $\sigma : \Sigma_1 \to \Sigma_2$,

$$|\rho^{Sign}(\Sigma_2)| = |\rho^{Sign}(\sigma)|(|\rho^{Sign}(\Sigma_1)|) \cup \rho^{PreSign}(|\Sigma_2| \setminus |\sigma|(|\Sigma_1|))$$

- $\rho^{Sen}_\Sigma(\varphi) = \rho^{PreSen}(\varphi)$, if $\varphi \in \mathbf{Sen}(\Sigma)$.

[a] Note that this is not the same as a functor **Set** $\to$ **Set**, since pre-signature morphisms are not equipped with codomains, and domains also differ from their standard meaning.

Theorem

If $\rho : \mathcal{I} \to \mathcal{I}'$ is modular, then for any signatures Σ_1, Σ_2 in $\mathcal{I}$ such that $\Sigma_1 \cup \Sigma_2$ is well-formed,

$$|\rho^{Sign}(\Sigma_1 \cup \Sigma_2)| = |\rho^{Sign}(\Sigma_1)| \cup |\rho^{Sign}(\Sigma_2)|$$

Definition

Institution comorphisms can induce institution morphisms via natural transformations: let $\rho\colon \mathcal{I} \to \mathcal{I}'$ be an institution comorphism, let $\mu^{Sign}\colon \mathbf{Sign}' \to \mathbf{Sign}$ be a functor and $\varepsilon\colon \rho^{Sign} \circ \mu^{Sign} \to id_{\mathbf{Sign}'}$ a natural transformation. Then ρ ε-induces the institution morphism $\mu = \langle \mu^{Sign}, \mu^{Sen}, \mu^{Mod} \rangle\colon\ \mathcal{I}' \to \mathcal{I}$, where for $\Sigma' \in |\mathbf{Sign}'|$, $\mu^{Sen}_{\Sigma'} = \mathbf{Sen}'(\varepsilon_{\Sigma'}) \circ \rho^{Sen}_{\mu^{Sign}(\Sigma')}$ and $\mu^{Mod}_{\Sigma} = \rho^{Mod}_{\mu^{Sign}(\Sigma')} \circ \mathbf{Mod}'(\varepsilon_{\Sigma'})$. Given such an institution morphism μ, we denote the corresponding institution comorphism ρ by $CoM(\mu)$.

Definition

A *modular heterogeneous logical environment* $\mathcal{HLE}$ is a collection of institutions with pre-signatures and of modular institution morphisms and comorphisms between them, that is, a pair of diagrams $\langle \mathcal{HLE}^\mu : \mathcal{G}^\mu \to \mathcal{INS}, \mathcal{HLE}^\rho : \mathcal{G}^\rho \to co\mathcal{INS} \rangle$ in the category $\mathcal{INS}$ of institutions and their morphisms and $co\mathcal{INS}$ of institutions and their comorphisms, respectively, such that the two underlying graphs have no common edges and diagrams coincide on common nodes, i.e., for all nodes $n \in |\mathcal{G}^\mu| \cap |\mathcal{G}^\rho|$, $\mathcal{HLE}^\mu(n) = \mathcal{HLE}^\rho(n)$.
For simplicity, we assume that each institution morphisms in $\mathcal{HLE}$ is induced by some institution comorphism in $\mathcal{HLE}$ via some natural transformation which is a pointwise inclusion. Most practical examples obey this additional assumption.

Definition

Consider institutions $\mathcal{I}$ and $\mathcal{I}'$ and signatures $\Sigma \in |\mathbf{Sign}|$ and $\Sigma' \in |\mathbf{Sign}'|$. A *heterogeneous signature morphism* is a pair $\langle \mu, \sigma \rangle \colon \Sigma \to \Sigma'$ that consists of an institution morphism $\mu \colon \mathcal{I}' \to \mathcal{I}$ and a signature morphism $\sigma \colon \Sigma \to \mu^{Sign}(\Sigma')$ in $\mathbf{Sign}$. It induces the *heterogeneous reduct* $\_|_{\langle \mu, \sigma \rangle} \colon \mathbf{Mod}'(\Sigma') \to \mathbf{Mod}(\Sigma)$ defined as the composition $\mathbf{Mod}(\sigma) \circ \mu^{Mod}_{\Sigma'}$, i.e., $M'|_{\langle \mu, \sigma \rangle} = \mu^{Mod}_{\Sigma'}(M')|_{\sigma}$, for all $M' \in \mathbf{Mod}'(\Sigma')$.

Definition

A *heterogeneous pre-signature morphism* is a pair $\langle \mu, \Delta \rangle$ that consists of an institution morphism $\mu \colon \mathcal{I}' \to \mathcal{I}$ and a pre-signature Δ. It is *well-formed* wrt. a source signature Σ and target signature Σ' if there is some heterogeneous signature morphism $\langle \mu, \sigma \rangle \colon \Sigma \to \Sigma'$ such that $|\sigma|$ is an inclusion and $\rho^{Sign}(\Sigma') \setminus \Sigma = \Delta$. In this case, $\langle \mu, \sigma \rangle$ is called the heterogeneous signature morphism from Σ to Σ' induced by $\langle \mu, \Delta \rangle$.

Definition

A *heterogeneous signature comorphism* is a pair $\langle \rho, \sigma' \rangle \colon \Sigma \to \Sigma'$ that consists of an institution comorphism $\rho{:}\mathcal{I} \to \mathcal{I}'$ and a signature morphism $\sigma'{:}\rho^{Sign}(\Sigma) \to \Sigma'$ in $\mathbf{Sign}'$. It induces the *heterogeneous reduct* $\_|_{\langle \rho, \sigma' \rangle} \colon \mathbf{Mod}'(\Sigma') \to \mathbf{Mod}(\Sigma)$ defined as the composition $\rho^{Mod}_{\Sigma} \circ \mathbf{Mod}'(\sigma')$, i.e., $M'|_{\langle \rho, \sigma' \rangle} = \rho^{Mod}_{\Sigma}(M'|_{\sigma'})$, for all $M' \in \mathbf{Mod}'(\Sigma')$.

Definition

A *heterogeneous pre-signature comorphism* is a pair $\langle \rho, \bar{\sigma} \rangle$ that consists of an institution comorphism $\rho \colon \mathcal{I} \to \mathcal{I}'$ and a pre-signature morphism $\bar{\sigma}$. It is *well-formed* if there is some heterogeneous signature comorphism $\langle \rho, \sigma \rangle \colon \Sigma \to \Sigma'$ such that σ is the signature morphism (in $\mathcal{I}'$) from $\rho^{Sign}(\Sigma)$ to Σ' induced by $\bar{\sigma}$. In this case, $\langle \rho, \sigma \rangle$ is called the heterogeneous signature comorphism from Σ to Σ' induced by $\langle \rho, \bar{\sigma} \rangle$.

Definition

Given institutions $\mathcal{I}, \mathcal{I}'$ and an $\mathcal{I}$-signature Σ. Two heterogeneous pre-signature comorphisms $\langle \rho_1, \bar{\sigma}_1 \rangle$ and $\langle \rho_2, \bar{\sigma}_2 \rangle$ with $\rho_1, \rho_2 : \mathcal{I} \to \mathcal{I}'$ are *equivalent on* Σ, written $\langle \rho_1, \bar{\sigma}_1 \rangle \equiv_\Sigma \langle \rho_2, \bar{\sigma}_2 \rangle$, if $\rho_1 = \rho_2$ and $fun_{\rho_1^{PreSign}(\Sigma)}(\bar{\sigma}_1) = fun_{\rho_1^{PreSign}(\Sigma)}(\bar{\sigma}_2)$.

Theorem

Two heterogeneous pre-signature comorphisms $\langle \rho_1, \bar{\sigma}_1 \rangle$ *and* $\langle \rho_2, \bar{\sigma}_2 \rangle$ *are equivalent on* Σ *if and only if* $\rho_1 = \rho_2$ *and* $\bar{\sigma}_1(x) = \bar{\sigma}_2(x)$ *for any* $x \in \rho_1^{PreSign}(\Sigma)$.

Definition

Let $\langle \rho, \sigma \rangle : \Sigma \to \Sigma'$ be a heterogeneous signature comorphism. It induces the heterogeneous pre-signature comorphism $\langle \rho, \bar{\sigma} \rangle$ where $\bar{\sigma}$ is the pre-signature morphisms induced by the signature morphism $\sigma : \rho^{PreSign}(\Sigma) \to \Sigma'$.

Theorem

The heterogeneous pre-signature comorphism induced by a heterogeneous signature co-morphism $\langle \rho, \sigma \rangle : \Sigma \to \Sigma'$ is well-formed and induces the same signature co-morphism $\langle \rho, \sigma \rangle$ between Σ and Σ'.

Definition

Given two heterogeneous signature comorphisms $\langle \rho_1, \sigma_1 \rangle \colon \Sigma_1 \to \Sigma_2$ and $\langle \rho_2, \sigma_2 \rangle \colon \Sigma_2 \to \Sigma_3$, their *composition* is defined as

$$\langle \rho_2, \sigma_2 \rangle \circ \langle \rho_1, \sigma_1 \rangle := \langle \rho_2 \circ \rho_1, \sigma_2 \circ \rho_2^{Sign}(\sigma_1) \rangle \colon \Sigma_1 \to \Sigma_3.$$

The problem of composing heterogeneous signature morphisms with heterogeneous signature comorphisms is solved by ε-inducibility:

Definition

Given a heterogeneous signature morphism $\langle \mu, \sigma \rangle : \Sigma \to \Sigma'$ such that μ is ε-induced by the institution comorphism ρ, the *ε-translation of* $\langle \mu, \sigma \rangle$ is the heterogeneous signature comorphism $\langle \rho, \varepsilon_{\Sigma'} \circ \rho^{\mathit{Sign}}(\sigma) \rangle : \Sigma \to \Sigma'$.

Theorem (Compatibility of Compositions)

Given heterogeneous pre-signature comorphisms $\langle \rho_1, \bar{\sigma}_1 \rangle$ *and* $\langle \rho_2, \bar{\sigma}_2 \rangle$*, such that there are heterogeneous signature comorphisms* $\langle \rho_1, \sigma_1 \rangle : \Sigma_1 \to \Sigma_2$ *and* $\langle \rho_2, \sigma_2 \rangle : \Sigma_2 \to \Sigma_3$ *induced by* $\langle \rho_1, \bar{\sigma}_1 \rangle$ *and* $\langle \rho_2, \bar{\sigma}_2 \rangle$*, respectively, then*

$$\langle \rho_2, \sigma_2 \rangle \circ \langle \rho_1, \sigma_1 \rangle : \Sigma_1 \to \Sigma_3 \text{ is induced by } \langle \rho_2, \bar{\sigma}_2 \rangle \circ \langle \rho_1, \bar{\sigma}_1 \rangle$$

Definition

Given a heterogeneous pre-signature morphism $\langle \mu, \Delta \rangle$ such that μ is ε-induced by the institution comorphism ρ, the *ε-translation of* $\langle \mu, \Delta \rangle$ is the heterogeneous pre-signature comorphism $\langle \rho, \emptyset \rangle$. The latter will induce a heterogeneous signature comorphism with a signature morphism component being an inclusion. Note that this is general enough because both ε and hiding wrt. Δ give inclusion signature morphisms.

Heterogeneous Development Graphs

Definition

Heterogeneous development graph $\mathcal{S} = \langle \mathcal{N}, \mathcal{L} \rangle$ over $\mathcal{HLE}$:
$\mathcal{N}$ is a set of nodes of form $(\mathcal{I}^N, \Sigma^N, \Gamma^N)$ such that $\mathcal{I}^N$ is an institution from $\mathcal{HLE}$, Σ^N is a $\mathcal{I}^N$-pre-signature called the **local signature** of N, and Γ^N a set of $\mathcal{I}$-sentences called the **local axioms** of N.
$\mathcal{L}$ is a set of directed links from a node M to a node N:

- **global** (denoted $M \xRightarrow{\langle \rho, \bar{\sigma} \rangle} N$), with a heterogeneous pre-signature comorphism $\langle \rho, \bar{\sigma} \rangle$ such that $\rho : \mathcal{I}^M \to \mathcal{I}^N$, or
- **local** (denoted $M \xrightarrow{\langle \rho, \bar{\sigma} \rangle} N$), with a heterogeneous pre-signature comorphism $\langle \rho, \bar{\sigma} \rangle$ such that $\rho : \mathcal{I}^M \to \mathcal{I}^N$, or
- **hiding** (denoted $M \xRightarrow[hide]{\langle \mu, \Delta \rangle} N$), with a heterogeneous pre-signature morphism $\langle \mu, \Delta \rangle$ where Δ is a $\mathcal{I}^M$-pre-signature of symbols to hide, or
- **free** (denoted $M \xRightarrow[free]{\Sigma_F} N$), annotated with a pre-signature of symbols over which N is freely generated.

Definition

The *global pre-signature* $Sig_{\mathcal{S}}(N)$ *of some node* N *wrt.* $\mathcal{S}$ is defined inductively over the definition links:

$$
\begin{aligned}
Sig_{\mathcal{S}}(N) = \Sigma^N \quad & \cup \bigcup_{M \xRightarrow{\langle\rho,\bar{\sigma}\rangle} N \in S} \bar{\sigma}(\rho^{PreSign}(Sig_{\mathcal{S}}(M))) \\
& \cup \bigcup_{M \xrightarrow{\langle\rho,\bar{\sigma}\rangle} N \in S} \bar{\sigma}(\rho^{PreSign}(\Sigma^M \cup sym(\Gamma^M))) \\
& \cup \bigcup_{M \xRightarrow[hide]{\langle\mu,\Delta\rangle} N \in S} \mu^{PreSign}(Sig_{\mathcal{S}}(M)) \setminus \Delta \\
& \cup \bigcup_{M \xRightarrow[free]{\Sigma_F} N \in S} Sig_{\mathcal{S}}(M)
\end{aligned}
$$

Definition

A node N has a well-formed signature iff $Sig_{\mathcal{S}}(N)$ is a valid $\mathcal{I}^N$-signature.
A development graph has a well-formed signature iff all its nodes have well-formed signatures.
Let M be a node with well-founded signature: we call the signature $Sig^{loc}_{\mathcal{S}}(M) := \langle \Sigma^M \cup sym(\Gamma^M) \rangle_{Sig_{\mathcal{S}}(M)}$ the *local signature* of M.

Definition

Given two nodes M and N with well-formed signatures, then

- $M \xRightarrow{\langle\rho,\bar{\sigma}\rangle} N$ induces a heterogeneous signature comorphism $\langle\rho,\sigma\rangle$ from $Sig_{\mathcal{S}}(M) \to Sig_{\mathcal{S}}(N)$;
- $M \xrightarrow{\langle\rho,\bar{\sigma}\rangle} N$ induces a heterogeneous signature comorphism $\langle\rho,\sigma\rangle$ from $Sig^{loc}_{\mathcal{S}}(M) \to Sig_{\mathcal{S}}(N)$;
- $M \xRightarrow[hide]{\langle\mu,\Delta\rangle} N$ induces a heterogeneous signature morphism $\langle\mu,\iota\rangle$ where $\iota : Sig_{\mathcal{S}}(N) \to \mu^{PreSign}(Sig_{\mathcal{S}}(M))$ is the identity inclusion;
- $M \xRightarrow[free]{\Sigma_F} N$ induces the trivial heterogeneous signature morphism $\langle Id,\iota\rangle$.

Definition

The set of **global axioms** of some N with well-formed signature is also defined inductively over the definition link structure:

$$
\begin{aligned}
Ax_{\mathcal{S}}(N) = \Gamma^N \quad & \cup \bigcup_{M \overset{\langle\rho,\bar{\sigma}\rangle}{\Longrightarrow} N \in S} \sigma(\rho^{PreSen}(Ax_{\mathcal{S}}(M))) \\
& \cup \bigcup_{M \xrightarrow{\langle\rho,\bar{\sigma}\rangle} N \in S} \sigma(\rho^{PreSen}(\Gamma^M)) \\
& \cup \bigcup_{M \underset{hide}{\overset{\langle\mu,\Delta\rangle}{\Longrightarrow}} N \in S} \{\varphi \in \mu^{PreSen}(Ax_{\mathcal{S}}(M)) | sym(\varphi) \cap \Delta = \emptyset\} \\
& \cup \bigcup_{M \underset{free}{\overset{\Sigma_F}{\Longrightarrow}} N \in S} Ax_{\mathcal{S}}(M)
\end{aligned}
$$

Definition

Let $\mathcal{S}$ be a development graph. The notion of *global reachability* is defined inductively: a node N is globally reachable from a node M via a heterogeneous pre-signature comorphism $\langle \rho, \bar{\sigma} \rangle$, $M \xRightarrow{\langle \rho, \bar{\sigma} \rangle} N$ for short, iff

- either $M = N$ and $\rho = id$, $\bar{\sigma} = id$, or
- $M \xRightarrow{\langle \rho', \bar{\sigma}' \rangle} K \in \mathcal{S}$, and $K \xRightarrow{\langle \rho'', \bar{\sigma}'' \rangle} N$, with $\langle \rho, \bar{\sigma} \rangle = \langle \rho'', \bar{\sigma}'' \rangle \circ \langle \rho', \bar{\sigma}' \rangle$.

A node N is **locally reachable** from a node M via a heterogeneous pre-signature comorphism $\langle \rho, \bar{\sigma} \rangle$, $M \xrightarrow{\langle \rho, \bar{\sigma} \rangle} N$ for short, iff $M \xRightarrow{\langle \rho, \bar{\sigma} \rangle} N$ or there is a node K with $M \xrightarrow{\langle \rho', \bar{\sigma}' \rangle} K \in \mathcal{S}$ and $K \xRightarrow{\langle \rho'', \bar{\sigma}'' \rangle} N$, such that $\langle \rho, \bar{\sigma} \rangle = \langle \rho'', \bar{\sigma}'' \rangle \circ \langle \rho', \bar{\sigma}' \rangle$.

Definition

Let $\mathcal{S} = \langle \mathcal{N}, \mathcal{L} \rangle$ be a development graph. A node $N \in \mathcal{N}$ is *flattenable* iff for all nodes $M \in \mathcal{N}$ with incoming hiding or free definition links, it holds that N is not globally reachable from M.

The models of flattenable nodes do not depend on existing hiding or free links. For flattenable nodes N, $Ax_{\mathcal{S}}(N)$ captures N completely. However, this is not the case for nodes that are not flattenable.

Definition

For $N \in \mathcal{N}$ with well-formed signature, $\mathbf{Mod}^{\mathcal{S}}(N)$ consists of those $Sig_{\mathcal{S}}(N)$-models n for which

- n satisfies the local axioms Γ^N,
- for each $K \overset{\langle\rho,\bar{\sigma}\rangle}{\Longrightarrow} N \in \mathcal{S}$, $n|_{\langle\rho,\sigma\rangle}$ is a K-model,
- for each $K \xrightarrow{\langle\rho,\bar{\sigma}\rangle} N \in \mathcal{S}$, $n|_{\langle\rho,\sigma\rangle}$ is a $Sig^{loc}_{\mathcal{S}}(K)$-model which satisfies the local axioms Γ^K, and
- for each $K \overset{\langle\mu,\Delta\rangle}{\underset{hide}{\Longrightarrow}} N \in \mathcal{S}$ with $\iota : \mu^{PreSign}(Sig_{\mathcal{S}}(K)) \setminus \Delta \to Sig_{\mathcal{S}}(N)$ the corresponding inclusion mapping, $n|_{\langle id,\iota\rangle}$ has a $\langle\mu,\theta\rangle$-expansion k that is a K-model where $\langle\mu,\theta\rangle$ is the heterogeneous signature morphism from $\mu^{PreSign}(Sig_{\mathcal{S}}(K)) \setminus \Delta$ to $Sig_{\mathcal{S}}(K)$ induced by $\langle\mu,\Delta\rangle$;
- for each $K \overset{\langle Id,\Sigma_F\rangle}{\underset{free}{\Longrightarrow}} N \in \mathcal{S}$, n is a K-model which is free (in the class of K-models) over its own ι-reduct, where $\iota\colon \langle\Sigma_F\rangle_{Sig_{\mathcal{S}}(K)} \to Sig_{\mathcal{S}}(K)$ is the inclusion.

Theorem

Let $\mathcal{S}$ be a heterogeneous development graph. Then:

- *if* $M \overset{\langle\rho,\bar\sigma\rangle}{\Longrightarrow} N$ *and* $n \in \mathbf{Mod}^{\mathcal{S}}(N)$, *then* $n|_{\langle\rho,\sigma\rangle} \in \mathbf{Mod}^{\mathcal{S}}(M)$ *where* $\langle\rho,\sigma\rangle : Sig_{\mathcal{S}}(M) \to Sig_{\mathcal{S}}(N)$ *is induced by* $\langle\rho,\bar\sigma\rangle$.
- *if* $M \xrightarrow{\langle\rho,\bar\sigma\rangle} N$ *and* $n \in \mathbf{Mod}^{\mathcal{S}}(N)$, *then* $n|_{\langle\rho,\sigma\rangle} \models \Gamma^M$ *where* $\langle\rho,\sigma\rangle : Sig^{loc}_{\mathcal{S}}(M) \to Sig_{\mathcal{S}}(N)$ *is induced by* $\langle\rho,\bar\sigma\rangle$.

Theorem

- $\mathbf{Mod}^{\mathcal{S}}(N) \subseteq \mathbf{Mod}_{Sig_{\mathcal{S}}(N)}(Ax_{\mathcal{S}}(N))$.
- *If N is flattenable, then* $\mathbf{Mod}^{\mathcal{S}}(N) = \mathbf{Mod}_{Sig_{\mathcal{S}}(N)}(Ax_{\mathcal{S}}(N))$.

Definition

A *theorem link* is

- local $N \stackrel{\langle\rho,\bar{\sigma}\rangle}{=\!=\!\Rightarrow} M$,
- global $N \stackrel{\langle\rho,\bar{\sigma}\rangle}{-\,-\!\!\succ} M$,
- a local implication $N \Rightarrow \Gamma$, $\Gamma \subseteq Sen(Sig_{\mathcal{S}}(N))$,
- hiding $N \underset{hide\ \langle\mu,\Delta\rangle}{\stackrel{\langle\rho,\sigma\rangle}{=\!=\!\Rightarrow}} M$
  (where for $\Sigma_H := \mu^{PreSign}(Sig_{\mathcal{S}}(N)) \setminus \Delta$, $\langle\mu,\Delta\rangle : Sig_{\mathcal{S}}(N) \to \Sigma_H$ and $\langle\rho,\sigma\rangle : \Sigma_H \to Sig_{\mathcal{S}}(M)$), or
- free $N \underset{free\ \langle Id,\Sigma_F\rangle}{\stackrel{\langle\rho,\bar{\sigma}\rangle}{=\!=\!\Rightarrow}} M$

Definition

Let $\mathcal{S}$ be a development graph and N, M nodes in $\mathcal{S}$.

- $\mathcal{S}$ **satisfies** a global theorem link $N \overset{\langle\rho,\bar{\sigma}\rangle}{=\!=\!\Rightarrow} M$ (denoted $\mathcal{S} \models N \overset{\langle\rho,\bar{\sigma}\rangle}{=\!=\!\Rightarrow} M$) iff for all $m \in \mathbf{Mod}^{\mathcal{S}}(M)$, $m|_{\langle\rho,\sigma\rangle} \in \mathbf{Mod}^{\mathcal{S}}(N)$ where $\langle\rho,\sigma\rangle$ is the heterogeneous signature comorphism from $Sig_{\mathcal{S}}(N)$ to $Sig_{\mathcal{S}}(M)$ induced by $\langle\rho,\bar{\sigma}\rangle$.
- $\mathcal{S}$ **satisfies** a local theorem link $N \overset{\langle\rho,\bar{\sigma}\rangle}{- - \rightarrow} M$ (denoted $\mathcal{S} \models N \overset{\langle\rho,\bar{\sigma}\rangle}{- - \rightarrow} M$) iff for all $m \in \mathbf{Mod}^{\mathcal{S}}(M)$, $m|_{\langle\rho,\sigma\rangle} \in \mathbf{Mod}_{Sig^{loc}_{\mathcal{S}}(N)}(\Gamma^N)$
- $\mathcal{S}$ **satisfies** a local implication $N \Rightarrow \Gamma$, written $\mathcal{S} \models N \Rightarrow \Gamma$, if for all $n \in \mathbf{Mod}_{\mathcal{S}}(N)$, $n \models \Gamma$.

Definition

- $\mathcal{S}$ **satisfies** a hiding theorem link $N \overset{\langle\rho,\bar{\sigma}\rangle}{\underset{hide\ \langle\mu,\Delta\rangle}{\Longrightarrow}} M$ (denoted $\mathcal{S} \models N \overset{\langle\rho,\bar{\sigma}\rangle}{\underset{hide\ \langle\mu,\Delta\rangle}{\Longrightarrow}} M$) iff for all $m \in \mathbf{Mod}^{\mathcal{S}}(M)$, $m|_{\langle\rho,\sigma\rangle\circ\langle id,\iota\rangle}$ has a $\langle\mu,\theta\rangle$-expansion to some N-model where $\langle\mu,\theta\rangle$ is the heterogeneous signature morphism from $\mu^{PreSign}(Sig_{\mathcal{S}}(N)) \setminus \Delta \to Sig_{\mathcal{S}}(N)$ induced by $\langle\mu,\Delta\rangle$, $\langle id,\iota\rangle : \mu^{PreSign}(Sig_{\mathcal{S}}(N))\Delta \to Sig_{\mathcal{S}}(M)$ is the identity inclusion, and $\langle\rho,\sigma\rangle$ is the heterogeneous signature comorphism from $\mu^{PreSign}(Sig_{\mathcal{S}}(N)) \setminus \Delta \to Sig_{\mathcal{S}}(M)$ induced by $\langle\rho,\bar{\sigma}\rangle$
- $\mathcal{S}$ **satisfies** a free theorem link $N \overset{\langle\rho,\bar{\sigma}\rangle}{\underset{free\ \langle Id,\Sigma_F\rangle}{\Longrightarrow}} M$ if for all $m \in \mathbf{Mod}^{\mathcal{S}}(M)$ it holds that $m|_{\langle\rho,\sigma\rangle}$ is an N-model which is free (in the class of N-models) over its own ι-reduct, where $\iota\colon \langle\Sigma_F\rangle_{Sig_{\mathcal{S}}(N)} \to Sig_{\mathcal{S}}(N)$ is the inclusion.

Global-Decomposition Rule

$$\frac{\begin{array}{l} N \overset{\langle\rho,\bar{\sigma}\rangle}{- - \succ} K \\ P \overset{\langle\rho,\bar{\sigma}\rangle\circ\langle\rho',\bar{\tau}\rangle}{- - \succ} K \text{ for each } P \xrightarrow{\langle\rho',\bar{\tau}\rangle} N \\ P \overset{\langle\rho,\bar{\sigma}\rangle\circ\langle\rho',\bar{\tau}\rangle}{===\Rightarrow} K \text{ for each } P \xRightarrow{\langle\rho',\bar{\tau}\rangle} N \\ P \underset{hide\ \langle\mu,\Delta\rangle}{\overset{\langle\rho,\bar{\sigma}\rangle}{===\Rightarrow}} K \text{ for each } P \underset{hide}{\xRightarrow{\langle\mu,\Delta\rangle}} N \\ P \underset{free\ \Sigma_F}{\overset{\langle\rho,\bar{\sigma}\rangle}{===\Rightarrow}} K \text{ for each } P \underset{free}{\xRightarrow{\Sigma_F}} N \end{array}}{N \overset{\langle\rho,\bar{\sigma}\rangle}{===\Rightarrow} K}$$

Borrowing Rule

$$
\dfrac{
\begin{array}{ccc}
K_{\langle\rho,\bar{\theta}\rangle} & & N^{cons}_{\langle\rho',\bar{\theta}'\rangle} \\
\Downarrow & & \Downarrow \\
K'_{\langle\rho_{\bar{\sigma}'},\bar{\sigma}'\rangle} & \Longrightarrow & N'
\end{array}
}{
\begin{array}{ccc}
K_{\langle\rho,\bar{\theta}\rangle\langle\rho_{\bar{\sigma}},\bar{\sigma}\rangle} & \Longrightarrow & N^{cons}_{\langle\rho',\bar{\theta}'\rangle} \\
\Downarrow & & \Downarrow \\
K' & & N'
\end{array}
}
$$

if $\langle\rho_{\bar{\sigma}'},\bar{\sigma}'\rangle \circ \langle\rho,\bar{\theta}\rangle \equiv \langle\rho',\bar{\theta}'\rangle \circ \langle\rho_{\bar{\sigma}},\bar{\sigma}\rangle$
wrt. $Sig_{\mathcal{S}}(K)$

Change Impact Analysis

- transfer proof work done in one particular development graph to another graph
- graphs differ one only in some locally constricted areas
- $\Rightarrow$ it is possible to relate most of the nodes and (definition) links of the two graphs
- $\Rightarrow$ mapping proof work encoded in theorem links, their decompositions, and proofs of theorems to the new development graph.
- Smart replay: anticipate the result of applying a rule in a changed setting by adaptation of the result of application in the original setting.
- *domain*: subgraph with all elements that contribute to the semantics of the involved entities
- *pre-domain* parts that actually (syntactically) matter for the rule application.

Global-Decomposition Rule

$$N \stackrel{\langle \rho, \bar{\sigma} \rangle}{\Longrightarrow} K$$

- *pre-domain*: $\langle \rho, \bar{\sigma} \rangle$, the node N and all direct incoming definition links (local, global, hiding, free) along with their heterogeneous pre-signature morphisms and comorphisms and their source nodes.
- *domain*: theorem link and the subgraphs imported into N and K including all signature elements and axioms.
- *Impact analysis*: if some definition link from the pre-domain is deleted, the corresponding (local/global) theorem link needs to be deleted as well. If some definition link has been added to the pre-domain, a new (local/global) theorem link needs to be added. If $\langle \rho, \bar{\sigma} \rangle$ has changed, the heterogeneous pre-signature comorphisms and heterogeneous pre-signature morphisms of the introduced theorem links are affected and must be recomputed. Analogously for incoming definition links and hiding definition links.

Borrowing Rule

$$
\begin{array}{ccc}
K_{\langle\rho,\bar{\theta}\rangle\,\langle\rho_{\bar{\sigma}},\bar{\sigma}\rangle} & \Longrightarrow & N^{cons}_{\langle\rho',\bar{\theta}'\rangle} \\
\Downarrow & & \Downarrow \\
K' & & N'
\end{array}
$$

- *domain:* imported subgraphs with all signature elements and axioms of K, N, K' and N', as well as all three theorem links with heterogeneous pre-signature comorphisms
- *pre-domain*: three theorem links and the global signature of K.
- *Impact Analysis:* if one of the involved heterogeneous pre-signature comorphisms is affected, the heterogeneous pre-signature comorphism of the new theorem link between K' and N' needs to be recomputed and the side condition rechecked. If the global signature of K has changed, then the side-condition needs to be rechecked.

Change impact analysis and pre-signature morphisms

- pre-signature morphisms change less frequently in general than full signature morphisms
- change of effect on axioms can be checked locally and without computing the full heterogeneous signature comorphism

Realization

- change impact analysis has been realized in HETS, but with signatures and signature morphisms rather than pre-signatures and pre-signature morphisms
- pre-signatures and pre-signature morphisms implemented it in the *GMoc*-tool for generic change impact analysis
- combination with change impact analysis of other documents: source code, requirements documents, general documentation
- information about affected theorem links it also provides information about those development graph nodes and links for which the signature and respectively the signature morphisms need to be recomputed by HETS
- impact analysis is formalized as a set of graph rewriting rules.

GMoC Change Management Tool

[AutexierMüller2010], [AutexierLüth2010]

- More principled approach with *one* tool parameterized over change impact analysis rules for different types of document
- Embrace existing types while being open to add interactions
- Allow for cross-document impact analysis rules to deal with heterogeneous collections of documents
- Comprises analysis of documents (consistency of document (meta-)properties)
- Use some of the intentional semantics of the documents

Modular Impact Analysis Rule Specification

Document Rules and Interaction Rules

- Avoid monolithical set of rules (difficult to extend)
- Parameterize analysis tool over modular sets of rules:
  - *document type specific* analysis rule systems
    - Use these to analyse single documents (specification input text)
  - *interaction rule systems* between documents of specific types
    - Interact between *semantic* graphs of the documents

Approach

- Given a set of documents of specific type
- Determine document type specific rule systems to use
- Determine interaction rule systems

How to Organize Interplay of Rule Systems?

Methodological Subdivision of Analysis

Annotation Model

For each document type $\mathbb{S}$, have three rule systems/phases

(i) an *abstraction* phase which synchronizes the semantic graph with the (new) document tree ($\alpha_{\mathbb{S}}$)

(ii) a *propagation* phase which propagates the information inside the semantic graph only ($\pi_{\mathbb{S}}$), and

(iii) a *projection* phase which dumps the information from the semantic graph into the document tree and its impact graph ($\iota_{\mathbb{S}}$)

Interaction Model

For each interaction model $\mathbb{I}$ only have propagation rule system $\pi_{\mathbb{I}}$.

Combined Analysis for Document Collections

Combined Models

Given Annotation Models and Interaction Models

- *Combined Abstraction:* $\alpha := \alpha_{\mathbb{S}_n} \circ \ldots \circ \alpha_{\mathbb{S}_1}$.
- *Combined Propagation:* exhaustive application of $\pi := \pi_{\mathbb{I}} \circ \pi_{\mathbb{S}_n} \circ \ldots \circ \pi_{\mathbb{S}_1}$ on g.
  I.e. fix point combinator *Fix* on $F = \lambda f.\lambda g.(\mathit{if}\ (g = \pi(g))\ g\ \mathit{else}\ f(\pi(g))$
- *Combined Projection:* $\iota := \iota_{\mathbb{S}_n} \circ \ldots \circ \iota_{\mathbb{S}_1}(g)$.

$$
\begin{tikzcd}
(D_1' \uplus \ldots \uplus D_n', S, \emptyset) \arrow[r, "\text{abstraction } \alpha"] & (D_1' \uplus \ldots \uplus D_n', S', \emptyset) \arrow[d, "\text{propagation } \pi"] \\
(D_1'' \uplus \ldots \uplus D_n'', S'', I) & (D_1' \uplus \ldots \uplus D_n', S'', \emptyset) \arrow[l, "\text{projection } \iota"']
\end{tikzcd}
$$

Realization

- Implemented the *GMoc* system on top of the graph rewriting system GrGen `www.grgen.net`
- Syntax for declaring document models and interaction models in configuration files
- Syntax for document collections and impact annotations
- Functionalities:
  - semantic difference analysis, annotation, change impact analysis, management of change

Document Model

```
<DocumentModel name="Guests">
 <suffix name="gxml"/>
 <equivspec filepath="Guests.eq"/>
 <graphmodel filepath="Guests.gm"/>
 <rulesystems>
  <abstraction top="guestAbs" filepath="GuestsAbstr.gri"/>
  <propagation top="guestProp" filepath="GuestsProp.gri"/>
  <projection top="guestProj" filepath="GuestsProj.gri"/>
 </rulesystems></DocumentModel>
```

Interaction Model

```
<InteractionModel name="GuestAndSeatingInteractionModel1">
 <partner name="Guests"/>
 <partner name="Seats"/>
 <graphmodel filepath="Guests2Seats.gm"/>
 <rulesystems>
 <propagation top="gsProp" filepath="GuestsSeatsProp.gri"/>
</rulesystems></InteractionModel>
```

Realization

- Implemented the *GMoc* system on top of the graph rewriting system GrGen `www.grgen.net`
- Syntax for declaring document models and interaction models in configuration files
- Syntax for document collections and impact annotations
- Functionalities:
    - semantic difference analysis, annotation, change impact analysis, management of change

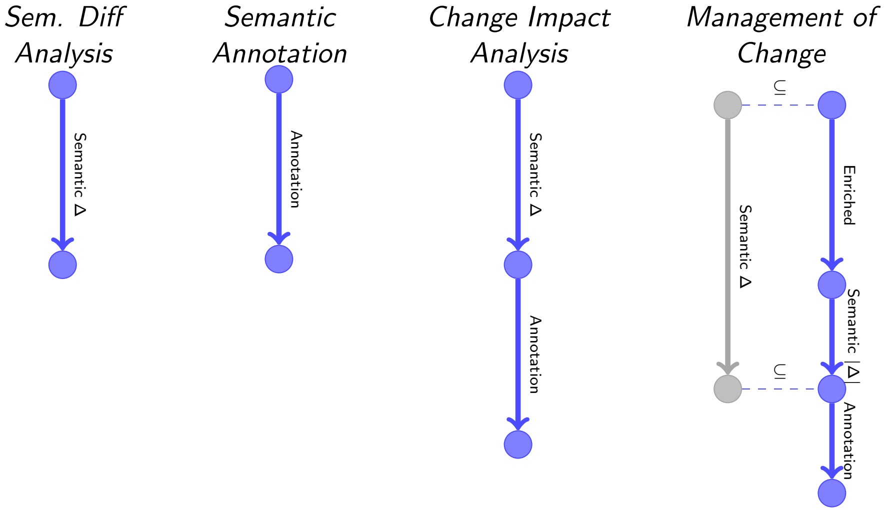
DocumentPlans for Input

```
<DocumentPlan>
 <Document id="guests" filename="guests.gxml"/>
 <Document id="seating" documentmodel="Seats"
           filename="seating.sxml"/>
 <exclude model="GuestAndSeatingInteractionModel1"/>
</DocumentPlan>
```

. . . and Output

```
<DocumentPlan>
 <Document id="guests" filename="guests.gxml"/>
 <Document id="seating" documentmodel="Seats"
           filename="seating.sxml">
   <Impacts>
     <Impact name="invalid seat assignment"
             xpath="/seatings/table[1]/chair [5]">
       Assigned person not confirmed</Impact></Impacts>
 </Document>
 <exclude model="GuestAndSeatingInteractionModel1"/>
</DocumentPlan>
```

Supported Scenarios / Functionalities



Application of GMoC to Hets

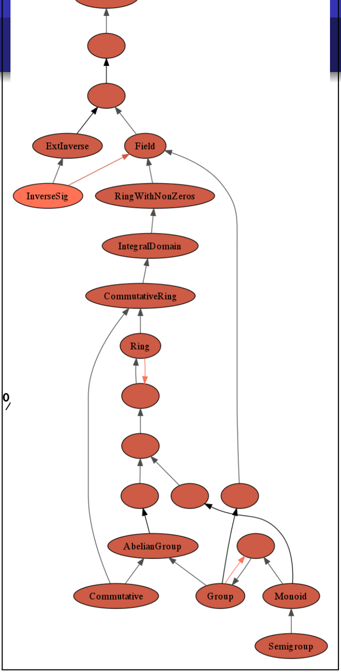
- Development graph calculus decomposes proof obligations
- Theorem provers discharge local proof obligations
- In case of change, compute which proofs of proof obligations are affected

Hets Specification

```
spec Commutative =
     sort Elem
     op __ * __, f: Elem * Elem -> Elem, comm

      forall x : Elem; y : Elem . x * y = y * x
end

spec Semigroup =
     sort Elem
     op __ * __: Elem * Elem -> Elem, assoc

      forall x : Elem; y : Elem; z : Elem . (x * y) * z = x * (y * z)
end

spec Monoid = Semigroup
then
     ops e:Elem;
         __ * __: Elem * Elem -> Elem, unit e;

      forall x : Elem . x * e = x
      forall x : Elem . e * x = x
end
```

Hets Specification

```
spec Ring =
     AbelianGroup with ops __ * __ |-> __ + __,
                           inv     |-> - __,
                           e       |-> 0
and
     Monoid with op e |-> 1
then
      forall x,y,z:Elem
     . (x + y) * z = (x * z) + (y * z)
%(distr1_Ring)%
     . z * ( x + y ) = (z * x) + (z * y) %(distr2_Ring)%
then %implies
     var x:Elem
     . 0 * x = 0 %(left zero)%
     . x * 0 = 0 %(right zero)%
end

spec CommutativeRing = Ring and Commutative

spec IntegralDomain =
     CommutativeRing
then
```



Hets Development Graph XML Representation

- Initial DGXML obtained from parsing specification
  - Proof obligations status: open
- Proof rule application in Hets changes proof status and adds theorems and new links
- Change of specification: new DGXML from parsing without proof information
- Management of Change scenario:
  - Compute edit-script on DG XML obtained from parsing
  - Apply edit-script on extended representation

```
<DGNode name="Ring__E2" refname="Ring'
  <Axioms>
    <Axiom>forall x : Elem; y : Elem; z : Ele
. (x + y) * z = (x * z) + (y * z);
Ring)</Axiom>
    <Axiom>forall x : Elem; y : Elem; z : Ele
. z * (x + y) = (z * x) + (z * y);
Ring)</Axiom>
  </Axioms>
</DGNode>

<DGLink linkid="13" source="Ring" target="
  <Type>GlobalUnprovenThmInc</Type>
  <GMorphism name="id_HasCASL.SubPCoC
</DGLink>
```

Hets Development Graph XML Representation

- Initial DGXML obtained from parsing specification
    - Proof obligations status: open
- Proof rule application in Hets changes proof status and adds theorems and new links
- Change of specification: new DGXML from parsing without proof information
- Management of Change scenario:
    - Compute edit-script on DG XML obtained from parsing
    - Apply edit-script on extended representation

```
<DGNode name="Ring__E2" refname="Ring'
  <Axioms>
    <Axiom>forall x : Elem; y : Elem; z : Ele
. (x + y) ∗ z = (x ∗ z) + (y ∗ z);
%(distr1_Ring)%</Axiom>
    <Axiom>forall x : Elem; y : Elem; z : Ele
. z ∗ (x + y) = (z ∗ x) + (z ∗ y);
%(distr2_Ring)%</Axiom>
  </Axioms>
<Theorems>
  <Theorem status="open">forall x : Elem . 0
  <Theorem status="open">forall x : Elem . x
</Theorems>
</DGNode>

<DGLink linkid="13" source="Ring" target="
  <Type>GlobalProvenThmInc</Type>
  <Status>Proven</Status>
  <Rule>Global−Decomposition</Rule>
  <ProofBasis linkref="12"/>
  <ProofBasis linkref="28"/>
  <GMorphism name="id_HasCASL.SubPCoC
```

Hets Development Graph XML Representation

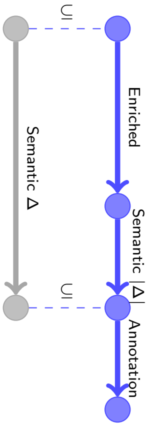
- Initial DGXML obtained from parsing specification
    - Proof obligations status: open
- Proof rule application in Hets changes proof status and adds theorems and new links
- Change of specification: new DGXML from parsing without proof information
- Management of Change scenario:
    - Compute edit-script on DG XML obtained from parsing
    - Apply edit-script on extended representation

```
<DGNode name="Ring__E2" refname="Ring'
  <Axioms>
    <Axiom>forall x : Elem; y : Elem; z : Ele
. (x + y) * z = (x * z) + (y * z);
Ring)</Axiom>
    <Axiom>forall x : Elem; y : Elem; z : Ele
. z * (x + y) = (z * x) + (z * y);
Ring)</Axiom>
  </Axioms>
</DGNode>

<DGLink linkid="13" source="Ring" target="
  <Type>GlobalUnprovenThmInc</Type>
  <GMorphism name="id_HasCASL.SubPCoC
</DGLink>
```

Hets Development Graph XML Representation

- Initial DGXML obtained from parsing specification
  - Proof obligations status: open
- Proof rule application in Hets changes proof status and adds theorems and new links
- Change of specification: new DGXML from parsing without proof information
- Management of Change scenario:
  - Compute edit-script on DG XML obtained from parsing
  - Apply edit-script on extended representation

Management of Change



Hets Development Graph XML Representation

- Initial DGXML obtained from parsing specification
  - Proof obligations status: open
- Proof rule application in Hets changes proof status and adds theorems and new links
- Change of specification: new DGXML from parsing without proof information
- Management of Change scenario:
  - Compute edit-script on DG XML obtained from parsing
  - Apply edit-script on extended representation

unordered dgnode {
  annotations = {name?}
}

unordered dglink {
  annotations = {source?, linkid ?, target ?}
}

unordered theorem {
  annotations = {name?}
  constituents = {<TEXT>}
}

Hets Abstraction Rules

- Compute relevant semantic entities
  - theory nodes, links, symbols, axioms, theorems, proof status, link decompositions
- Synchronize syntax and semantics
- 82 rules/patterns

```
pattern detectTheory(dg:DGNode) {
  a: Attribute ;
  dg <-:IsAttribute- a;
  : isAttribute (dg,a,"name");

   alternative  {
  Old {
    t:SemTheory;
    t  -:Origin-> dg;
    :wasExistingObject(dg,t);
    mark:markPreserved(t);
    modify {
     mark();
     eval { t.name = a.value; }
     emithere ("Found old theory "+a.value+"\n");
    }
  }
  New {
    negative { :CIANode -:Origin-> dg; }
    modify {
     emithere ("Found new theory "+a.value+"\n");
     t:SemTheory -:Origin-> dg;
     eval { t.name = a.value; }
    }
  }
  }
  modify {}
}
```

Hets Abstraction Rules

- Compute relevant semantic entities
  - theory nodes, links, symbols, axioms, theorems, proof status, link decompositions
- Synchronize syntax and semantics
- 82 rules/patterns

```
rule hetsdgabstraction {
  modify {
   exec ( resetLinkIdCounter );
   exec ( detectTheories ) ;
   exec ( detectLinks );
   exec ( detectSymbols );
   exec ( detectAxioms );
   exec ( detectTheorems );
   exec ( detectDecompositions );
  }
}
```

Hets Propagation Rules

- Use status (added, preserved, deleted) to detect qualitative changes
- Propagate detected qualitative changes
- Requires fine-grained theory of DGs
    - Capture signature and theory construction mechanism
    - Institutions with pre-signatures
    - 51 rules/patterns

```
alternative {
 LocallyExtended {
   addedsym:SemSymbol -:SemContainer-> th;
   :isAdded(addedsym);
   negative {
     deletedsym:SemSymbol -:SemContainer-> th;
     : isDeleted (deletedsym);
   }
   modify {
     th <-:CIAAnnotate- :SIGExtendedLocally;
     emithere ("Theory "+th.name+" has locally extended signature.\n
   }
 }
 LocallyRestricted {
   negative { addedsym:SemSymbol -:SemContainer-> th;
              :isAdded(addedsym); }
   deletedsym:SemSymbol -:SemContainer-> th;
   : isDeleted (deletedsym);

   modify {
     th <-:CIAAnnotate- :SIGRestrictedLocally;
     emithere ("Theory "+th.name+" has locally restricted signature.
   }
 }
 LocallyModified {
```

Hets Propagation Rules

- Use status (added, preserved, deleted) to detect qualitative changes
- Propagate detected qualitative changes
- Requires fine-grained theory of DGs
  - Capture signature and theory construction mechanism
  - Institutions with pre-signatures
  - 51 rules/patterns

```
    negative { addedsym:SemSymbol —:SemContainer—> th;
               :isAdded(addedsym); }
    negative { deletedsym:SemSymbol —:SemContainer—> th;
               : isDeleted (deletedsym); }

    modify {
      th <—:CIAAnnotate— :SIGUnchangedLocally;
      emithere ("Theory "+th.name+" has locally unchanged signature.\
    }
  }
  LocalllyUnchanged {
    addedsym:SemSymbol —:SemContainer—> th;
    :isAdded(addedsym);
    deletedsym:SemSymbol —:SemContainer—> th;
    : isDeleted (deletedsym);

    modify {
      th <—:CIAAnnotate— :SIGModifiedLocally;
      emithere ("Theory "+th.name+" has locally modified signature.\n
    }
  }
}
modify {}
}
```

Hets Propagation Rules

- Use status (added, preserved, deleted) to detect qualitative changes
- Propagate detected qualitative changes
- Requires fine-grained theory of DGs
  - Capture signature and theory construction mechanism
  - Institutions with pre-signatures
  - 51 rules/patterns

```
pattern theoryGlobalSigModifications (th:SemTheory) {
    alternative {
    GloballyExtended {
        : theoryAllSigNotRestricted (th);
        :theorySomeSigExtended(th);
        modify {
            th <−:CIAAnnotate− :SIGExtendedGlobally;
            emithere ("Theory "+th.name+" is globally extended by signatu
        }
    }
     GloballyRestricted {
        :theoryAllSigNotExtended(th);
        :theorySomeSigRestricted(th);
        modify {
             th <−:CIAAnnotate− :SIGRestrictedGlobally;
              emithere ("Theory "+th.name+" is globally restricted on the
        }
      }
    GloballyUnchanged {
        :theoryAllSigNotExtended(th);
        : theoryAllSigNotRestricted (th);
        modify {
          th <−:CIAAnnotate− :SIGUnchangedGlobally;
          emithere ("Theory "+th.name+" is globally unchanged on the sign
```

Hets Propagation Rules

- Use status (added, preserved, deleted) to detect qualitative changes
- Propagate detected qualitative changes
- Requires fine-grained theory of DGs
  - Capture signature and theory construction mechanism
  - Institutions with pre-signatures
  - 51 rules/patterns

```
        emithere ("Theory "+th.name+" is globally unchanged on the signa
      }
    }
    GloballyModified {
      :theorySomeSigRestricted(th);
      :theorySomeSigExtended(th);
      modify {
        th <-:CIAAnnotate- :SIGModifiedGlobally;
         emithere ("Theory "+th.name+" is globally modified (extended ar
      }
    }
  }
  modify {}
}
```

Hets Projection Rules

- Propogate Semantic Properties as Impact Annotations back along Origin links (e.g. `SIGRestrictedLocally`)
    - 1 generic rule
    - Extract Impacts as XML document
- Also allow change of document itself (DGXML)
    - Adjust linkids and maxlinkid

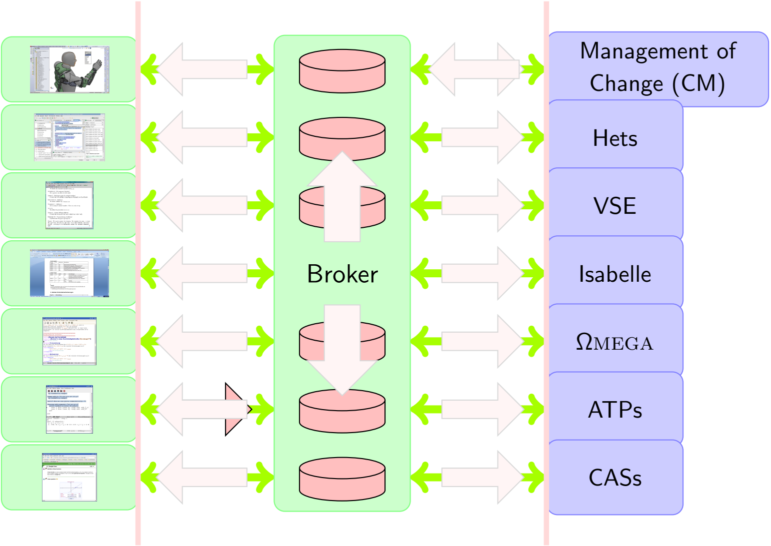
```
pattern projectAnnotations {
  iterated {
    th:CIANode -:Origin-> o:GmocNode;
    iterated {
      p:CIAProperty -:CIAAnnotate-> th;
      negative { if { p.description == ""; }}
      modify {
        i:Impact -:affects-> o;
  eval { i.name = p.description; i.value = "true"; }
  emithere ("Projected "+p.description+"\n");
      }
    }
   modify {}
  }
  modify {}
}
```

Hets Projection Rules

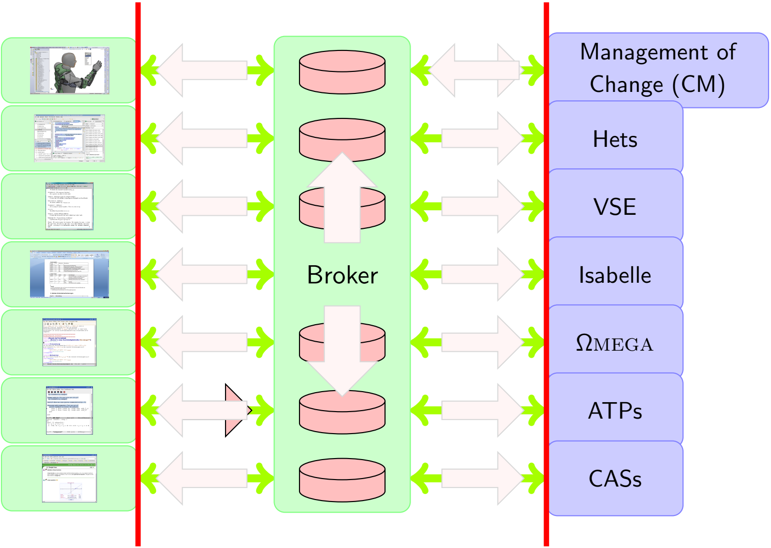
- Propogate Semantic Properties as Impact Annotations back along Origin links (e.g. `SIGRestrictedLocally`)
    - 1 generic rule
    - Extract Impacts as XML document
- Also allow change of document itself (DGXML)
    - Adjust linkids and maxlinkid

```
rule projectAdjustedLinkids {
  l:Link -:Origin-> dg:DGLink;
  a: Attribute ;
  : isAttribute (dg,a,"linkid");
  negative { if { a.value == l.linkid ; }}
  modify {
   emit("Adapting linkid in Syntax for link "+l.linkid+"\n");
   eval { a.value = l. linkid ; }
  }
}

rule projectMaxLinkId {
  c:LinkIdCounter;
  d:DGraph;
  a: Attribute ;
  : isAttribute (d,a,"nextlinkid");
  negative { if { a.value == (string) c.value; }}
  modify {
    emit ("Saving new value of nextlinkid "+c.value+" in Syntax\n");
    eval { a.value = (string) c.value; }
  }
}
```

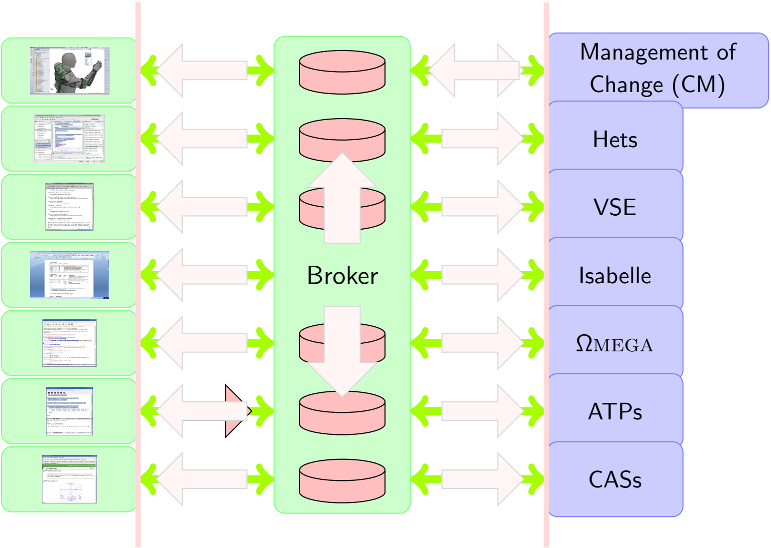
Document and Tool Integration Platform DocTIP



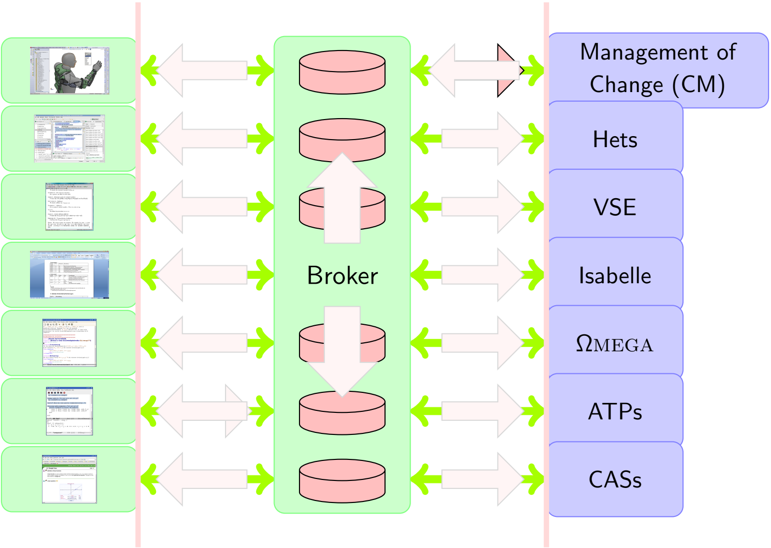
Document and Tool Integration Platform DocTIP



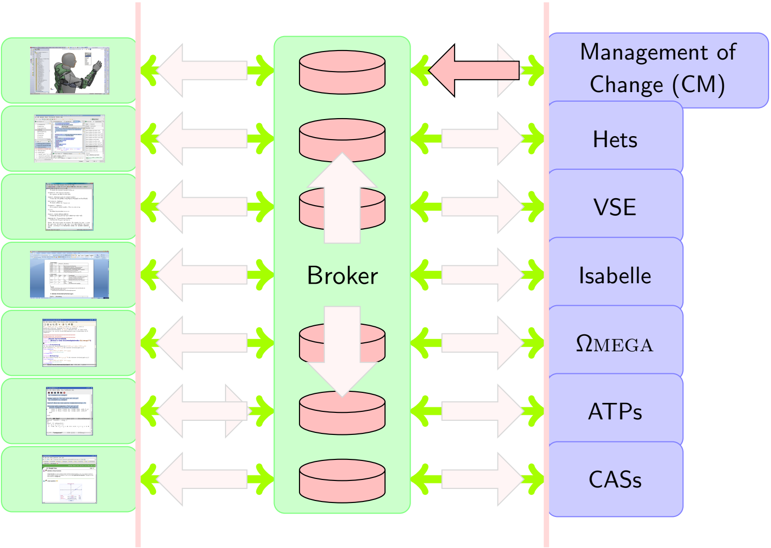
Document and Tool Integration Platform DocTIP



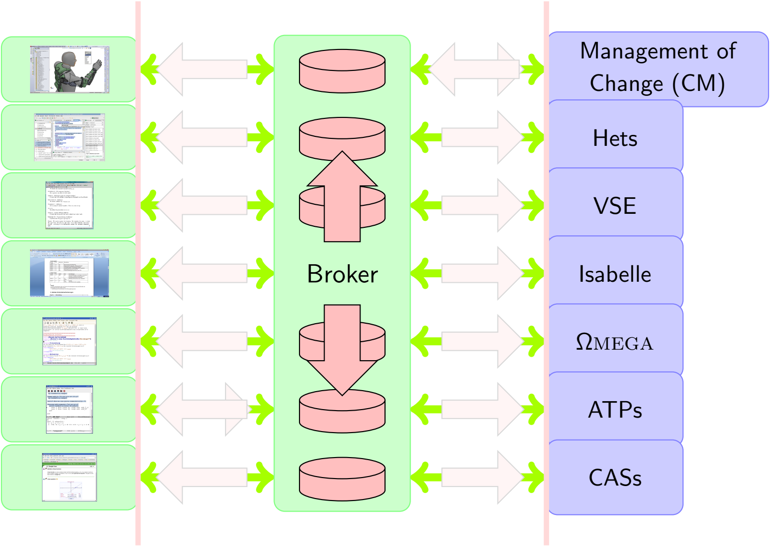
Document and Tool Integration Platform DocTIP



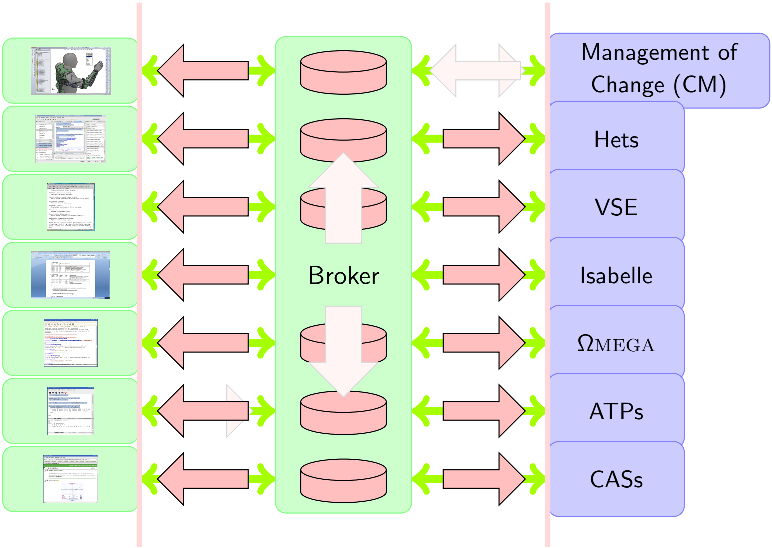
Document and Tool Integration Platform DocTIP



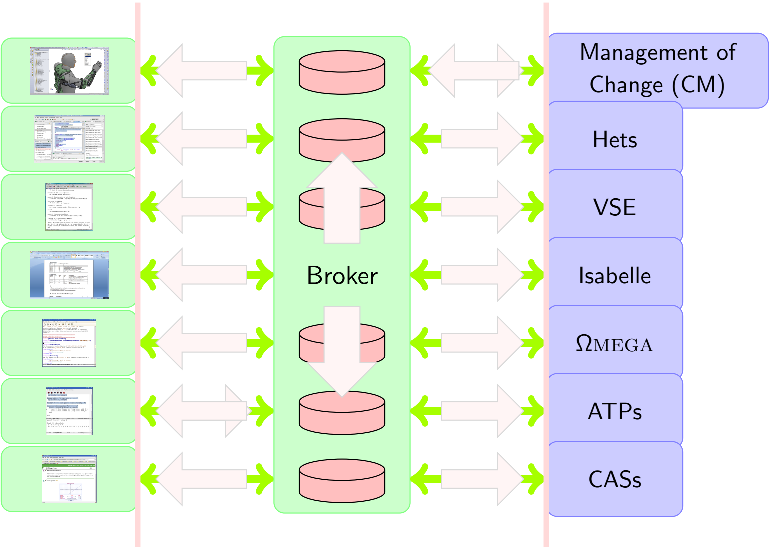
Document and Tool Integration Platform DocTIP



Document and Tool Integration Platform DocTIP



Document and Tool Integration Platform DocTIP



Related Work

Change impact analysis

- Lots of methods to determine software change impacts based on modeling of data, control, and component dependencies.
- Restricted to specific document kinds, but do not support interaction with others

Requirements traceability

- tracing requirements over different levels of refinement
- systems like DOORS, no link between requirements and software artifacts

Conclusion

- provided a framework for change management of heterogeneous specifications
- pre-signatures and pre-signature morphisms allow us to specify theories in a completely modular way.
- DG proof rules make use of this modularity: restrict focus of rule application to some few nodes and their relations in the development graph
- smart replay mechanism anticipates the result of applying a rule in a changed setting
- implemetation using GMoC's *change-aware* graph rewriting strategies
- `www.dfki.de/sks/hets`
- `www.dfki.de/sks/omoc/gmoc.html`

Problems/Wishes/Future

- Termination analysis
- Link with logic formalisms: as alternative reasoning mechanism (e.g, Symbolic Constraint Satisfaction), but especially specification of impact analysis strategies and prove properties thereof
- Improve interface with GrGen
- extend the framework of change management to the use of generalized theoroidal institution comorphisms