Heterogeneous Logical Environments for Distributed Specifications*

Till Mossakowski[1,2] and Andrzej Tarlecki[3,4]

[1] Institute for Computer Science, Albert-Ludwigs-Universität Freiburg
[2] Safe and Secure Cognitive Systems, DFKI GmbH, Bremen
[3] Institute of Informatics, University of Warsaw
[4] Institute of Computer Science, Polish Academy of Sciences

Abstract. We use the theory of institutions to capture the concept of a heterogeneous logical environment as a number of institutions linked by institution morphisms and comorphisms. We discuss heterogeneous specifications built in such environments, with inter-institutional specification morphisms based on both institution morphisms and comorphisms. We distinguish three kinds of heterogeneity: (1) specifications in logical environments with universal logic (2) heterogeneous specifications focused at a particular logic, and (3) heterogeneous specifications distributed over a number of logics.

1 Introduction

The theory of *institutions* [GB92] provides an excellent framework where the theory of specification and formal software development may be presented in an adequately general and abstract way [ST88a, ST97, Tar03]. The initial work within this area captured specifications built and developments carried out in an arbitrary but fixed logical system formalised as an institution. However, the practice of software specification and development goes much beyond this. Different logical systems may be appropriate or most convenient for specification of different modules of the same system, of different aspects of system behaviour, or of different stages of system development. This leads to the need for a number of logical systems to be used in the same specification and development project, linked by appropriate notions of morphisms between institutions [GR02]. This observation spurred a substantial amount of research work already, and motivates the research presented here.

In such a framework, one works in a *heterogeneous logical environment* formed by a number of logical systems formalised as institutions and linked with each other in a way captured by various maps between institutions. One such logical environment is the Hets family of institutions [Mos05], supported by a tool to build and work with heterogeneous specifications [MML07].

* This work has been partially supported by European projects IST-2005-015905 MOBIUS and IST-2005-016004 SENSORIA, by a visiting grant to the University of Illinois at Urbana-Champaign (AT) and by the German Federal Ministry of Education and Research (Project 01 IW 07002 FormalSafe) and by the DFG-funded SFB/TR 8 "Spatial cognition" (TM).

A. Corradini and U. Montanari (Eds.): WADT 2008, LNCS 5486, pp. 266–289, 2009.

Given a heterogeneous logical environment, there are several possible ways of using it to build heterogeneous specifications:

1. In some logical environments, we have a single logical system (usually coming with good tool support) that can be used as a *universal logic* into which all other systems are mapped. Then the maps between logics are used for mapping specifications from all logics in the environment into the universal logic, where they can be further combined then as usual. Various logical systems have been proposed and used as such universal logics, including higher-order logic in various versions [NPW02], Edinburgh LF [HHP93], rewriting logic [MOM02], fork algebra [PF06], etc.
2. *Focused heterogeneous specifications* are more liberal: parts of a specification may be written in different logics (also exploiting the availability of specialised tools for these logics). However, these parts ultimately are assembled in one logical system, where the models of interest live. This is made possible by extending the repertoire of *specification-building operations* with ones that move specifications from one logic to another using various maps between logical systems, as perhaps first mentioned in the context similar to what we use here in [ST88b] and further developed in [Tar96, Tar00, Mos03, Dia02].
3. *Distributed heterogeneous specifications* involve a number of specifications in different logical systems, with compatibility links between them given by logic maps, but not necessarily with a single specification in a particular logic providing an overall integration. *Heterogeneous development graphs* [Mos02b] offer a first hint in this direction.

While the first two methodologies have been studied in the literature, the third one seems to have attracted only little attention from the formal specification community so far, although it is clear that in frameworks like UML, distributed heterogeneous specifications arise rather naturally.

In this paper we largely set up a framework for further work, collecting the ideas, concepts and facts put forward earlier at other places (by us and others). No new big results are to be expected at this stage. However, a new overall view of heterogeneous logical environments and distributed specifications in such environment seems to be emerging here.

We introduce a notion of a heterogeneous logical environment, and study to what extent such environments can be made *uniform*, i.e., based on one kind of a mapping between institutions. We discuss various ways of building focused heterogeneous specifications in such environments. Then, given heterogeneous, inter-institutional (co)morphisms between such specifications and specification categories they define, we introduce distributed specifications as specification diagrams. These come with a natural notion of a distributed model, and so also other standard concepts like consistency, consequence, implementations, etc. Finally, we show that these concepts apply in the context of heterogeneous specification categories built over any heterogeneous logical environment.

2 Heterogeneous Logical Environments

Let us begin by recalling the notion of an institution, as a formalisation of an arbitrary logical system [GB92], assuming that the reader is familiar with all the intuitions that this notion brings in.

Definition 2.1. *An* institution $\mathcal{I}$ *consists of:*

- *a category* $\mathbf{Sign}_{\mathcal{I}}$ *of* signatures*;*
- *a functor* $\mathbf{Sen}_{\mathcal{I}}\colon \mathbf{Sign}_{\mathcal{I}} \to \mathbf{Set}$,[1] *giving a set* $\mathbf{Sen}(\Sigma)$ *of* Σ-sentences *for each signature* $\Sigma \in |\mathbf{Sign}_{\mathcal{I}}|$*, and a function* $\mathbf{Sen}(\sigma)\colon \mathbf{Sen}(\Sigma) \to \mathbf{Sen}(\Sigma')$*, denoted by* σ*, that yields* σ-translation *of* Σ*-sentences to* Σ'*-sentences for each signature morphism* $\sigma\colon \Sigma \to \Sigma'$*;*
- *a functor* $\mathbf{Mod}_{\mathcal{I}}\colon \mathbf{Sign}_{\mathcal{I}}^{op} \to \mathbf{Set}$,[2] *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* $M' \in \mathbf{Mod}_{\mathcal{I}}(\Sigma')$*:*

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

Whenever convenient, we avoid spelling out the standard notations for institution components, and allow primes, subscripts and superscripts to determine which institution is referred to.

The next concept we need is a mapping between institutions. We concentrate here on *institution morphisms* [GB92] and on *institution comorphisms* (named so in [GR02]; see "plain maps of institutions" in [Mes89] and "institution representations" in [Tar87, Tar96]).

Definition 2.2. *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* $\mu^{Sen}\colon \mu^{Sign} ; \mathbf{Sen}' \to \mathbf{Sen}$,[3] *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* $\mu^{Mod}\colon \mathbf{Mod} \to (\mu^{Sign})^{op} ; \mathbf{Mod}'$*, 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* $M \in \mathbf{Mod}(\Sigma)$*:*

---

[1] The category **Set** has all sets as objects and all functions as morphisms.

[2] To keep things simple, we work with the version of institutions where morphisms between models, not needed here, are disregarded. To capture standard examples, we should allow here for the use of classes, rather than just sets of models — but again, we will disregard such foundational subtleties here.

[3] We write composition of morphisms in any category in the diagrammatic order and denote it by ";" (semicolon).

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

Institution morphisms compose in the obvious, component-wise manner. The category of institutions with institution morphisms is denoted by $\mathcal{INS}$.

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* $\rho^{Sen}\colon \mathbf{Sen} \to \rho^{Sign} ; \mathbf{Sen}'$*, 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* $\rho^{Mod}\colon (\rho^{Sign})^{op} ; \mathbf{Mod}' \to \mathbf{Mod}$*, 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* $M' \in \mathbf{Mod}'(\rho^{Sign}(\Sigma))$*:*

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

Institution comorphisms compose in the obvious, component-wise manner. The category of institutions with institution comorphisms is denoted by co$\mathcal{INS}$.

Whenever no confusion may arise, the superscripts identifying the components of an institution morphism will be omitted, so that all components of an institution morphism μ will be written as μ, and similarly for institution comorphisms.

Even though the only essential difference between institution morphisms and comorphisms is in the direction of sentence and model translations w.r.t. signature translation, the intuition they capture is quite different. Very informally, an institution morphism $\mu\colon \mathcal{I} \to \mathcal{I}'$ shows how a "richer" institution $\mathcal{I}$ is "projected" onto a "poorer" institution $\mathcal{I}'$ (by removing some parts of signatures and models of $\mathcal{I}$ to obtain the simpler signatures and models of $\mathcal{I}'$, and by embedding simpler $\mathcal{I}'$-sentences into more powerful $\mathcal{I}$-sentences). Then, an institution comorphism $\mathcal{I} \to \mathcal{I}'$ shows how a "simpler" institution $\mathcal{I}$ is represented in a "more complex" institution $\mathcal{I}'$ (by representing the simpler signatures and sentences of $\mathcal{I}$ as signatures and sentences of $\mathcal{I}'$, and extracting simpler $\mathcal{I}$-models from more complex $\mathcal{I}'$-models).[4]

Given the two possible ways to link institutions with each other, a notion of a *heterogeneous logical environment* may be formalised as a collection of institutions linked by institution morphisms and comorphisms.

Definition 2.3. *A* heterogeneous logical environment $\mathcal{HLE}$ *is a collection of institutions and of institution morphisms and comorphisms between them, that is, a pair of diagrams* $\langle \mathcal{HLE}^\mu\colon \mathcal{G}^\mu \to \mathcal{INS}, \mathcal{HLE}^\rho\colon \mathcal{G}^\rho \to co\mathcal{INS}\rangle$[5] *in the category*

[4] Variants of comorphisms are also used to encode "more complex" institutions into "simpler" ones: e.g. in [GR02], a so-called simple theoroidal comorphism is used to code first-order logic with equality in first-order logic without equality. See also [MDT09] for discussion of relative strength of logical systems in a similar context.

[5] We assume that $\mathcal{G}^\mu$ is a graph that gives the shape of the diagram; its nodes $n \in |\mathcal{G}^\mu|$ carry institutions $\mathcal{HLE}^\mu(n)$ linked by institution morphisms $\mathcal{HLE}^\mu(e)\colon \mathcal{HLE}^\mu(n) \to \mathcal{HLE}^\mu(m)$ for each edge $e\colon n \to m$ in $\mathcal{G}$. Similar notation is used for diagrams in other categories.

$\mathcal{INS}$ 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 $n \in |\mathcal{G}^\mu| \cap |\mathcal{G}^\rho|$, $\mathcal{HLE}^\mu(n) = \mathcal{HLE}^\rho(n)$.

We write $\mathcal{G}$ for the union of $\mathcal{G}^\mu$ and $\mathcal{G}^\rho$, and w.l.o.g. assume that all the nodes of the underlying graphs are common, $|\mathcal{G}| = |\mathcal{G}^\mu| = |\mathcal{G}^\rho|$.

Such a heterogeneous logical environment is morphism-uniform *if $\mathcal{G}^\rho$ is discrete (has no edges); we can then identify $\mathcal{HLE}$ with $\mathcal{HLE}^\mu \colon \mathcal{G} \to \mathcal{INS}$. Similarly, $\mathcal{HLE}$ is* comorphism-uniform *if $\mathcal{G}^\mu$ is discrete; we can identify it then with $\mathcal{HLE}^\rho \colon \mathcal{G} \to \mathcal{INS}$.*

The lack of uniformity in linking institutions in heterogeneous environments (we use both institution morphisms and comorphisms here; other kinds of maps between institutions may be considered as well) may be somewhat surprising and certainly is technically cumbersome. Morphism-uniform environments, where only institution morphisms are used, are conveniently captured by a single diagram in the appropriate institution category, and similarly for comorphism-uniform environments. These concepts coincide with what was studied in the literature as *indexed institutions* [Dia02] and *indexed coinstitutions* [Mos02a].

One way to make logical environments uniform is by noticing that in fact a link of each kind may be captured by links of the other kind, albeit in general a span of those may be needed. This has been noticed already in [Mos03] and spelled out in [Mos05]; see also [MW98] for similar ideas with institution forward morphisms (called transformations there) as a primary notion.

Definition 2.4. *Consider an institution morphism $\mu \colon \mathcal{I} \to \mathcal{I}'$. We build an "intermediate institution" by re-indexing $\mathcal{I}'$ using the signature translation: $\mathcal{I}'_0 = \langle \mathbf{Sign}, \mu^{Sign} ; \mathbf{Sen}', \mu^{Sign} ; \mathbf{Mod}', \langle \models'_{\mu^{Sign}(\Sigma)} \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$. Two comorphisms emerge then*[6]*: $\rho_{\mu,1} = \langle id, \mu^{Sen}, \mu^{Mod} \rangle \colon \mathcal{I}'_0 \to \mathcal{I}$ and $\rho_{\mu,2} = \langle \mu^{Sign}, id, id \rangle \colon \mathcal{I}'_0 \to \mathcal{I}'$. The* comorphism span for μ*, written span(μ), is the following span of institution comorphisms: $\mathcal{I} \xleftarrow{\rho_{\mu,1}} \mathcal{I}'_0 \xrightarrow{\rho_{\mu,2}} \mathcal{I}'$.*

Consider an institution comorphism $\rho \colon \mathcal{I} \to \mathcal{I}'$. We build an "intermediate institution": $\mathcal{I}'_0 = \langle \mathbf{Sign}, \rho^{Sign} ; \mathbf{Sen}', \rho^{Sign} ; \mathbf{Mod}', \langle \models'_{\rho^{Sign}(\Sigma)} \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$. Two institution morphisms emerge then: $\mu_{\rho,1} = \langle id, \rho^{Sen}, \rho^{Mod} \rangle \colon \mathcal{I}'_0 \to \mathcal{I}$ and $\mu_{\rho,2} = \langle \mu^{Sign}, id, id \rangle \colon \mathcal{I}'_0 \to \mathcal{I}'$. The morphism span for ρ*, written span(ρ), is the following span of institution morphisms: $\mathcal{I} \xleftarrow{\mu_{\rho,1}} \mathcal{I}'_0 \xrightarrow{\mu_{\rho,2}} \mathcal{I}'$.*

Informally, the span of comorphisms *span*(μ) captures exactly the same relationship between the components of $\mathcal{I}$ and $\mathcal{I}'$ as the original institution morphism $\mu \colon \mathcal{I} \to \mathcal{I}'$; and similarly, the span of morphisms *span*(ρ) captures exactly the same relationship between the components of $\mathcal{I}$ and $\mathcal{I}'$ as the original institution comorphism $\rho \colon \mathcal{I} \to \mathcal{I}'$.

[6] We write *id* for identities in any category, in particular, here for the identity functor as well as the identity natural transformations.

This essentially allows us to concentrate on heterogeneous logical environments that are uniform in the sense that only institution morphisms (or comorphisms) are used to link with each other the institutions involved.

Definition 2.5. *Let* $\mathcal{HLE} = \langle \mathcal{HLE}^\mu \colon \mathcal{G}^\mu \to \mathcal{INS}, \mathcal{HLE}^\rho \colon \mathcal{G}^\rho \to co\mathcal{INS} \rangle$ *be a heterogeneous logical environment. By* $span^\mu(\mathcal{HLE}) \colon span^\mu(\mathcal{G}) \to \mathcal{INS}$ *we denote the morphism-uniform environment obtained from* $\mathcal{HLE}$ *by replacing each comorphism* ρ *in* $\mathcal{HLE}^\rho$ *by* $span(\rho)$*. Similarly, by* $span^\rho(\mathcal{HLE}) \colon span^\rho(\mathcal{G}) \to co\mathcal{INS}$ *we denote the comorphism-uniform environment obtained from* $\mathcal{HLE}$ *by replacing each morphism* μ *in* $\mathcal{HLE}^\mu$ *by* $span(\mu)$*.*

Note that the "uniformisation" described above typically will change the shape of the graph underlying the heterogeneous logical environment: while building $span^\mu(\mathcal{HLE})$ we remove each edge in $\mathcal{G}^\rho$ adding a corresponding span of edges in $\mathcal{G}^\mu$, with a new node that carries the new "intermediate" institution, and similarly for $span^\rho(\mathcal{HLE})$.

A very rough intuition about various ways of linking institutions by spans and sinks of (co)morphisms is that in a span of comorphisms $\mathcal{I} \xleftarrow{\rho} \mathcal{I}_0 \xrightarrow{\rho'} \mathcal{I}'$ and in a sink of morphisms $\mathcal{I} \xrightarrow{\mu} \mathcal{I}_0 \xleftarrow{\mu'} \mathcal{I}'$, the intermediate institution $\mathcal{I}_0$ captures the common features of $\mathcal{I}$ and $\mathcal{I}'$, and so this relationship may be used to express some "sharing" requirements between models of $\mathcal{I}$ and $\mathcal{I}'$. Dually, in a sink of comorphisms $\mathcal{I} \xrightarrow{\rho} \mathcal{I}_0 \xleftarrow{\rho'} \mathcal{I}'$ and in a span of morphisms $\mathcal{I} \xleftarrow{\mu} \mathcal{I}_0 \xrightarrow{\mu'} \mathcal{I}'$, the intermediate institution $\mathcal{I}_0$ is richer than both $\mathcal{I}$ and $\mathcal{I}'$ and combines the features present in them, and therefore may be used to express some "consistency" properties between models of $\mathcal{I}$ and $\mathcal{I}'$.

3 Specifications and Their Heterogeneous Categories

The original purpose of introducing the notion of institution (under the name of a *language* in [BG80]) was to free the theory of specifications from dependency on any particular logical system. We follow [ST88a] and for an arbitrary institution $\mathcal{I}$ consider a class $Spec_{\mathcal{I}}$ of specifications built in $\mathcal{I}$ starting from *basic specifications* (*presentations*, which essentially consist of a signature and a set of sentences over this signature) by means of a number of *specifications-building operations*, including *union* of specifications with common signature (written $SP_1 \cup SP_2$), *translation* along a signature morphism (written $\sigma(SP)$), *hiding* (or "*derive*") w.r.t. a signature morphism (written $SP'|_\sigma$), etc. We will not dwell here on the particular choice of these operations, as usual assuming though that specifications come with their basic semantics given in terms of model classes. That is, for each specification $SP \in Spec_{\mathcal{I}}$, we have its *signature* $Sig[SP] \in |\mathbf{Sign}|$ and its class of *models* $Mod[SP] \subseteq \mathbf{Mod}(Sig[SP])$. In particular, we have $Mod[SP_1 \cup SP_2] = Mod[SP_1] \cap Mod[SP_2]$, $Mod[\sigma(SP)] = \{M' \mid M'|_\sigma \in Mod[SP]\}$, and $Mod[SP'|_\sigma] = \{M'|_\sigma \mid M' \in Mod[SP']\}$. The semantics also determines the obvious notion of specification equivalence: $SP_1 \equiv SP_2$ iff $Sig[SP_1] = Sig[SP_2]$ and $Mod[SP_1] = Mod[SP_2]$.

Working in a heterogeneous logical environment, where we have a number of institutions linked by institution morphisms and comorphisms (no uniformity assumption necessary at this stage), we can enrich the collection of specification-building operations by translation along institution comorphisms and hiding w.r.t. institution morphisms, see [ST88b, Tar96]. Somewhat less naturally, we can also define translation along institution morphisms and hiding w.r.t. institution comorphisms, but the target signature has to be given explicitly then:

Definition 3.1. *Let $\mu\colon \mathcal{I} \to \mathcal{I}'$ be an institution morphism. Given a specification $SP \in Spec_{\mathcal{I}}$, we write $SP|_{\mu}$ for a new specification in $Spec_{\mathcal{I}'}$ with the semantics given by $Sig[SP|_{\mu}] = \mu^{Sign}(Sig[SP])$ and $Mod[SP|_{\mu}] = \mu^{Mod}(Mod[SP])$ $(= \{\mu^{Mod}_{Sig[SP]}(M) \mid M \in Mod[SP]\})$.*

Given a specification $SP' \in Spec_{\mathcal{I}'}$ and signature $\Sigma \in |\mathbf{Sign}|$ such that $\mu^{Sign}(\Sigma) = Sig[SP']$, we write $\mu(SP')^{\Sigma}$ for a new specification in $Spec_{\mathcal{I}}$ with the semantics given by $Sig[\mu(SP')^{\Sigma}] = \Sigma$ and $Mod[\mu(SP')^{\Sigma}] = (\mu^{Mod})^{-1}(Mod[SP'])$ $(= \{M \in \mathbf{Mod}(\Sigma) \mid \mu^{Mod}_{\Sigma}(M) \in Mod[SP']\})$.

Let $\rho\colon \mathcal{I} \to \mathcal{I}'$ be an institution comorphism. Given a specification $SP \in Spec_{\mathcal{I}}$, we write $\rho(SP)$ for a new specification in $Spec_{\mathcal{I}'}$ with the semantics given by $Sig[\rho(SP)] = \rho^{Sign}(Sig[SP])$ and $Mod[\rho(SP)] = (\rho^{Mod})^{-1}(Mod[SP])$ $(= \{M' \in \mathbf{Mod}'(\rho^{Sign}(Sig[SP])) \mid \rho^{Mod}_{Sig[SP]}(M') \in Mod[SP]\})$.

Given a specification $SP' \in Spec_{\mathcal{I}'}$ and signature $\Sigma \in |\mathbf{Sign}|$ such that $\rho^{Sign}(\Sigma) = Sig[SP']$, we write $SP'|^{\Sigma}_{\rho}$ for a new specification in $Spec_{\mathcal{I}}$ with the semantics given by $Sig[SP'|^{\Sigma}_{\rho}] = \Sigma$ and $Mod[SP'|^{\Sigma}_{\rho}] = \rho^{Mod}(Mod[SP'])$ $(= \{\rho^{Mod}_{\Sigma}(M') \mid M' \in Mod[SP']\})$.

These new, inter-institutional specification-building operations may be arbitrarily mixed with other (intra-institutional) operations, yielding heterogeneous specifications. Parts of such specifications may be given in different institutions of the heterogeneous logical environment we work in. However, each such a specification as a whole eventually focuses on a particular institution in this environment, where its overall semantics (signature and the class of models) is given. In essence, viewed from a certain perspective, such *focused heterogeneous specifications* do not differ much from the structured specifications built within a single institution. For instance, the view of a software specification and development process as presented in [ST88b, ST97] directly adapts to the use of such specifications without much (semantic) change. For each institution $\mathcal{I}$, we still denote the class of such heterogeneous specifications focused on $\mathcal{I}$ by $Spec_{\mathcal{I}}$.

The standard notion of a specification morphism carries over to heterogeneous specifications focused at the same institution without any change: a specification morphism between specifications $SP, SP' \in Spec_{\mathcal{I}}$ is a signature morphism $\sigma\colon Sig[SP] \to Sig[SP']$ such that for all models $M' \in Mod[SP']$, $M'|_{\sigma} \in Mod[SP]$. This yields the category $\mathbf{Spec}_{\mathcal{I}}$ of specifications focused on $\mathcal{I}$, with the model-class semantics that extends to the functor $\mathbf{Mod}\colon \mathbf{Spec}^{op}_{\mathcal{I}} \to \mathbf{Set}$.

To generalise this definition to a truly heterogeneous case, with specifications involved focused on different institutions, we first have to appropriately generalise the notion of a signature morphism. Of course, we need some link (given by an institution morphism or comorphism) between the institutions involved.

Definition 3.2. *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* **Sign**. *It induces the* heterogeneous reduct $\_|_{\langle \mu, \sigma \rangle} \colon \mathbf{Mod}'(\Sigma') \to \mathbf{Mod}(\Sigma)$ *defined as the composition $\mu^{Mod}_{\Sigma'} ; \mathbf{Mod}(\sigma)$, i.e., $M'|_{\langle \mu, \sigma \rangle} = \mu^{Mod}_{\Sigma'}(M')|_{\sigma}$, for all $M' \in \mathbf{Mod}'(\Sigma')$.*

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 $\mathbf{Mod}'(\sigma') ; \rho^{Mod}_{\Sigma}$, i.e., $M'|_{\langle \rho, \sigma' \rangle} = \rho^{Mod}_{\Sigma}(M'|_{\sigma'})$, for all $M' \in \mathbf{Mod}'(\Sigma')$.*

Heterogeneous signature morphisms compose as expected: $\langle \mu_1, \sigma_1 \rangle ; \langle \mu_2, \sigma_2 \rangle = \langle \mu_2 ; \mu_1, \sigma_1 ; \mu_1^{Sign}(\sigma_2) \rangle$. For any morphism-uniform heterogeneous logical environment $\mathcal{HLE}^{\mu} \colon \mathcal{G}^{\mu} \to \mathcal{INS}$ this yields the heterogeneous category $\mathbf{Sign}(\mathcal{HLE}^{\mu})$ of signatures in institutions in $\mathcal{HLE}^{\mu}$ with heterogeneous morphisms that involve institution morphisms in $\mathcal{HLE}^{\mu}$ (and their compositions, and identities). Then model functors extend to $\mathbf{Mod}(\mathcal{HLE}^{\mu}) \colon \mathbf{Sign}(\mathcal{HLE}^{\mu})^{op} \to \mathbf{Set}$ using the reducts defined above.

Heterogeneous signature comorphisms compose as expected: $\langle \rho_1, \sigma_1 \rangle ; \langle \rho_2, \sigma_2 \rangle = \langle \rho_1 ; \rho_2, \rho_2^{Sign}(\sigma_1) ; \sigma_2 \rangle$. For any comorphism-uniform heterogeneous logical environment $\mathcal{HLE}^{\rho} \colon \mathcal{G}^{\rho} \to co\mathcal{INS}$ this yields the heterogeneous category $\mathbf{Sign}(\mathcal{HLE}^{\rho})$ of signatures in institutions in $\mathcal{HLE}^{\rho}$ with heterogeneous comorphisms that involve institution comorphisms in $\mathcal{HLE}^{\rho}$ (and their compositions, and identities). Then model functors extend to $\mathbf{Mod}(\mathcal{HLE}^{\rho}) \colon \mathbf{Sign}(\mathcal{HLE}^{\rho})^{op} \to \mathbf{Set}$ using the reducts defined above.

We stop short here of defining translation of sentences and proving the satisfaction condition. Otherwise though, the above follows the construction of the category of signatures and model reducts in the Grothendieck institution given in [Dia02] for the morphism-uniform case and in [Mos02a] for the comorphism-uniform case (heterogeneous signature (co)morphisms were called Grothendieck signature morphisms there). For full formality, signatures in the heterogeneous categories of signatures defined above should really be written as pairs $\langle \mathcal{I}, \Sigma \rangle$, marking them explicitly with the institution they come from (or even with the nodes in the institution diagram) but we continue relying on the reader's good will to decipher the institution from the context.

Note that we retain the overall informal idea that a signature morphism goes from the simpler to more complex signature — hence the contravariance with the use of institution morphisms in the definition of heterogeneous signature morphism. This intuition also dictated the choice of the placement and direction of signature morphism components in heterogeneous signature (co)morphisms.

Note also that the inter-institutional specification-building operations given in Def. 3.1 arise now as hiding w.r.t. and translation along heterogeneous signature morphisms and comorphisms (with the identities as signature morphisms).

In a (non-uniform) heterogeneous logical environment $\mathcal{HLE}$, Def. 3.2 yields two heterogeneous signature categories, one for the morphism-uniform, the other for the comorphism-uniform part of $\mathcal{HLE}$. The two categories share all objects, but have different morphisms. We can put these categories together by formally adding compositions of morphisms of the two kinds involved, modulo the expected identification of morphisms that arise from intra-institutional signature morphisms by adding identity institution (co)morphisms.

Definition 3.3. *Let $\mathcal{HLE} = \langle \mathcal{HLE}^\mu, \mathcal{HLE}^\rho \rangle$ be a heterogeneous logical environment, and consider the disjoint union of the signature categories of all institutions in $\mathcal{HLE}$, which embeds in the obvious way into both $\mathbf{Sign}(\mathcal{HLE}^\mu)$ and $\mathbf{Sign}(\mathcal{HLE}^\rho)$. We write $\mathbf{Sign}(\mathcal{HLE})$ for the* heterogeneous category of signatures *in $\mathcal{HLE}$ and their* generalised heterogeneous morphisms*, defined as the pushout (in $\mathbf{Cat}$[7]) of the two embedding functors. The model functors extend to $\mathbf{Mod}(\mathcal{HLE})\colon \mathbf{Sign}(\mathcal{HLE})^{op} \to \mathbf{Set}$ using the compositions of reducts from Def. 3.2.*

We again stop short from extending this definition to a complete construction of a *Bi-Grothendieck institution* for $\mathcal{HLE}$ with $\mathbf{Sign}(\mathcal{HLE})$ as its signature category and $\mathbf{Mod}(\mathcal{HLE})$ as its model functor, as spelled out in [Mos03].

Given the above, the definitions of inter-institutional specification morphisms are now obvious (cf. [Dia98] for a similar notion of an extra theory morphism, and the notions of specification morphisms arising in Grothendieck (co)institutions).

Definition 3.4. *Consider a heterogeneous logical environment $\mathcal{HLE}$, institutions $\mathcal{I}$ and $\mathcal{I}'$ in $\mathcal{HLE}$ and specifications $SP \in Spec_{\mathcal{I}}$ and $SP' \in Spec_{\mathcal{I}'}$.*

A generalised heterogeneous signature morphism $\zeta\colon Sig[SP] \to Sig[SP'] \in \mathbf{Sign}(\mathcal{HLE})$ is a heterogeneous specification morphism *$\zeta\colon SP \to SP'$ if for all models $M' \in Mod[SP']$, $M'|_\zeta \in Mod[SP]$. Heterogeneous specification morphisms compose, which yields the heterogeneous category $\mathbf{Spec}(\mathcal{HLE})$ of specifications focused on institutions in $\mathcal{HLE}$. The model functions extend to the functor $\mathbf{Mod}(\mathcal{HLE})\colon \mathbf{Spec}(\mathcal{HLE})^{op} \to \mathbf{Set}$, using heterogeneous reducts.*

Given a heterogeneous logical environment $\mathcal{HLE} = \langle \mathcal{HLE}^\mu, \mathcal{HLE}^\rho \rangle$, the heterogeneous category of specifications $\mathbf{Spec}(\mathcal{HLE})$ has subcategories $\mathbf{Spec}(\mathcal{HLE}^\mu)$ and $\mathbf{Spec}(\mathcal{HLE}^\rho)$, given by heterogeneous signature morphisms of the form $\langle \mu, \sigma \rangle \in \mathbf{Sign}(\mathcal{HLE}^\mu)$ and, respectively, by heterogeneous signature comorphisms of the form $\langle \rho, \sigma \rangle \in \mathbf{Sign}(\mathcal{HLE}^\rho)$.

The category $\mathbf{Spec}_{\mathcal{I}}$ of specifications focused on a particular institution in an environment is a subcategory of the heterogeneous category of specifications built in the environment (via the obvious embedding which adds identity institution (co)morphisms).

Directly from the definitions:

[7] $\mathbf{Cat}$ is the (quasi-)category of all categories, as usual.

Lemma 3.5. *Consider institutions $\mathcal{I}$, $\mathcal{I}'$ and specifications $SP \in Spec_{\mathcal{I}}$, $SP' \in Spec_{\mathcal{I}'}$.*

A heterogeneous signature morphism $\langle \mu, \sigma \rangle \colon Sig[SP] \to Sig[SP']$ (so that $\mu \colon \mathcal{I}' \to \mathcal{I}$, $\sigma \colon Sig[SP] \to \mu^{Sign}(Sig[SP'])$) is a heterogeneous specification morphism $\langle \mu, \sigma \rangle \colon SP \to SP'$ if and only if $\sigma \colon SP \to SP'|_{\mu}$ is a specification morphism (in $\mathbf{Spec}_{\mathcal{I}}$).

A heterogeneous signature comorphism $\langle \rho, \sigma' \rangle \colon Sig[SP] \to Sig[SP']$ (so that $\rho \colon \mathcal{I} \to \mathcal{I}'$, $\sigma' \colon \rho^{Sign}(Sig[SP]) \to Sig[SP']$) is a heterogeneous specification morphism $\langle \rho, \sigma' \rangle \colon SP \to SP'$ if and only if $\sigma' \colon \rho(SP) \to SP'$ is a specification morphism (in $\mathbf{Spec}_{\mathcal{I}'}$).

In all the specification categories we consider, equivalent specifications are isomorphic (identity signature morphisms being isomorphisms between them).

One observation about the heterogeneous categories of specifications as defined above is that they reveal a potential problem with making heterogeneous logical environments uniform. Replacing institution morphisms by spans of comorphisms, and vice-versa, replacing institution comorphisms by spans of morphisms, changes the inter-institutional specification-building operations that are available in the logical environment. Fortunately, in view of symmetry in Def. 3.1, this is not much of a problem:

Lemma 3.6. *Let $\mu : \mathcal{I} \to \mathcal{I}'$ be an institution morphism, and let $span(\mu)$ be $\mathcal{I} \xleftarrow{\rho_{\mu,1}} \mathcal{I}'_0 \xrightarrow{\rho_{\mu,2}} \mathcal{I}'$.*

- *$SP|_{\mu} \equiv \rho_{\mu,2}(SP|^{\Sigma}_{\rho_{\mu,1}})$ for any $SP \in Spec_{\mathcal{I}}$ with $Sig[SP] = \Sigma$.*
- *$\mu(SP')^{\Sigma} \equiv \rho_{\mu,1}(SP'|^{\Sigma}_{\rho_{\mu,2}})$ for any $SP' \in Spec_{\mathcal{I}'}$ with $\mu^{Sign}(\Sigma) = Sig[SP']$.*

Dually then, let $\rho : \mathcal{I} \to \mathcal{I}'$ be an institution comorphism, and let $span(\rho)$ be $\mathcal{I} \xleftarrow{\mu_{\rho,1}} \mathcal{I}'_0 \xrightarrow{\mu_{\rho,2}} \mathcal{I}'$.

- *$\rho(SP) \equiv \mu_{\rho,1}(SP)^{\Sigma}|_{\mu_{\rho,2}}$ for any $SP \in Spec_{\mathcal{I}}$ with $Sig[SP] = \Sigma$.*
- *$SP'|^{\Sigma}_{\rho} \equiv \mu_{\rho,2}(SP')^{\Sigma}|_{\mu_{\rho,1}}$ for any $SP' \in Spec_{\mathcal{I}'}$ with $\rho^{Sign}(\Sigma) = Sig[SP']$.*

Together with Lemma 3.5, this yields a useful characterisation of heterogeneous specification (co)morphisms, see [Mos03]:

Proposition 3.7. *Let $\mu \colon \mathcal{I} \to \mathcal{I}'$ be an institution morphism, where $span(\mu)$ is $\mathcal{I} \xleftarrow{\rho_{\mu,1}} \mathcal{I}'_0 \xrightarrow{\rho_{\mu,2}} \mathcal{I}'$. Consider specifications $SP \in Spec_{\mathcal{I}}$ and $SP' \in Spec_{\mathcal{I}'}$ and heterogeneous signature morphism $\langle \mu, \sigma \rangle \colon Sig[SP'] \to Sig[SP]$. Then $\langle \mu, \sigma \rangle \colon SP' \to SP$ is a specification morphism if and only if $\sigma \colon SP' \to \rho_{\mu,2}(SP|^{Sig[SP]}_{\rho_{\mu,1}})$ is a specification morphism (in $\mathbf{Spec}_{\mathcal{I}'}$).*

Let $\rho \colon \mathcal{I} \to \mathcal{I}'$ be an institution morphism, where $span(\rho)$ is $\mathcal{I} \xleftarrow{\mu_{\rho,1}} \mathcal{I}'_0 \xrightarrow{\mu_{\rho,2}} \mathcal{I}'$. Consider specifications $SP \in Spec_{\mathcal{I}}$ and $SP' \in Spec_{\mathcal{I}'}$ and heterogeneous signature comorphism $\langle \rho, \sigma \rangle \colon Sig[SP] \to Sig[SP']$. Then $\langle \rho, \sigma \rangle \colon SP \to SP'$ is a specification morphism if and only if $\sigma \colon \mu_{\rho,1}(SP)^{Sig[SP]}|_{\mu_{\rho,2}} \to SP'$ is a specification morphism (in $\mathbf{Spec}_{\mathcal{I}'}$).

This proposition is the central motivation for the use of spans. It means that all proof obligations arising in a mixed heterogeneous logical environment can be properly expressed already in a uniform heterogeneous logical environment, using spans (even if one cannot express them directly as a theory morphism between original specifications).

However, specification morphisms that arise from institution morphisms and spans of comorphisms that replace them are quite different, and similarly for comorphisms and spans of morphisms. In particular, given a heterogeneous specification morphism $\langle \mu, \sigma \rangle \colon SP' \to SP$, with an institution morphism $\mu \colon \mathcal{I} \to \mathcal{I}'$, there seems to be no natural way to link specifications SP and SP' by heterogeneous specification comorphisms built over the comorphism span $\mathit{span}(\mu)$ (but see Sect. 5.5 for more on this). Consequently, for a heterogeneous logical environment $\mathcal{HLE}$, the heterogeneous specification categories $\mathbf{Spec}(\mathcal{HLE})$, $\mathbf{Spec}(\mathit{span}^{\mu}(\mathcal{HLE}))$ and $\mathbf{Spec}(\mathit{span}^{\rho}(\mathcal{HLE}))$ are quite different in general.

4 Uniformity via Signature Adjunctions

The discrepancy between the categories of heterogeneous specifications built on institution morphisms and the one built on the spans of comorphisms that could replace them, pointed out in Sect. 3, may seem a bit disturbing and suggests a search for "better" ways of making heterogeneous logical environment uniform. One such possibility arises in most practical examples, when institution morphisms involve "forgetful" signature functors that have left adjoints (which restores the "forgotten" structure of signatures in the source institution in the free way) and/or when institution comorphisms involve signature functors that have right adjoints (that forget the extra structure added to signatures of the source institution to encode them in the target institution). Under such circumstances, institution morphisms can be turned into comorphisms, and vice versa, see [AF96, Dia08]:

Theorem 4.1. *Let $\mu \colon \mathcal{I}' \to \mathcal{I}$ be an institution morphism and let $\mu^{Sign} \colon \mathbf{Sign}' \to \mathbf{Sign}$ have a left adjoint $\rho^{Sign} \colon \mathbf{Sign} \to \mathbf{Sign}'$ with unit $\eta \colon id_{\mathbf{Sign}} \to \rho^{Sign} ; \mu^{Sign}$. Then $L(\mu) = \langle \rho^{Sign}, \rho^{Sen}, \rho^{Mod} \rangle$, where for $\Sigma \in |\mathbf{Sign}|$, $\rho^{Sen}_{\Sigma} = \mathbf{Sen}(\eta_{\Sigma});\mu^{Sen}_{\rho^{Sign}(\Sigma)}$ and $\rho^{Mod}_{\Sigma} = \mu^{Mod}_{\rho^{Sign}(\Sigma)} ; \mathbf{Mod}(\eta_{\Sigma})$, is an institution comorphism $L(\mu) \colon \mathcal{I} \to \mathcal{I}'$.*

Let $\rho \colon \mathcal{I} \to \mathcal{I}'$ be an institution comorphism and let $\rho^{Sign} \colon \mathbf{Sign} \to \mathbf{Sign}'$ have a right adjoint $\mu^{Sign} \colon \mathbf{Sign}' \to \mathbf{Sign}$ with counit $\varepsilon \colon \mu^{Sign} ; \rho^{Sign} \to id_{\mathbf{Sign}'}$. Then $R(\rho) = \langle \mu^{Sign}, \mu^{Sen}, \mu^{Mod} \rangle$, where for $\Sigma' \in |\mathbf{Sign}'|$, $\mu^{Sen}_{\Sigma'} = \rho^{Sen}_{\mu^{Sign}(\Sigma')} ; \mathbf{Sen}'(\varepsilon_{\Sigma'})$ and $\mu^{Mod}_{\Sigma} = \mathbf{Mod}'(\varepsilon_{\Sigma'}) ; \rho^{Mod}_{\mu^{Sign}(\Sigma')}$, is an institution morphism $R(\rho) \colon \mathcal{I}' \to \mathcal{I}$.

Moreover, R and L can be chosen so that $R(L(\mu)) = \mu$ and $L(R(\rho)) = \rho$.

Now, given any heterogeneous logical environment $\mathcal{HLE} = \langle \mathcal{HLE}^{\mu}, \mathcal{HLE}^{\rho} \rangle$, if all institution comorphisms in $\mathcal{HLE}^{\rho}$ have signature functors with right adjoints, we can build a morphism-uniform heterogeneous logical environment $\mathit{adj}^{\mu}(\mathcal{HLE})$ with each institution comorphism ρ in $\mathcal{HLE}^{\rho}$ replaced by the institution morphism $R(\rho)$. Then the specification categories $\mathbf{Spec}(\mathcal{HLE})$ and $\mathbf{Spec}(\mathit{adj}^{\mu}(\mathcal{HLE}))$ are equivalent. Similarly, if all institution morphisms in $\mathcal{HLE}^{\mu}$ have signature functors with left adjoints, we can build a comorphism-uniform heterogeneous logical environment $\mathit{adj}^{\rho}(\mathcal{HLE})$ with each institution morphism

μ in $\mathcal{HLE}^{\mu}$ replaced by the institution comorphism $L(\mu)$. Then the specification categories $\mathbf{Spec}(\mathcal{HLE})$ and $\mathbf{Spec}(adj^{\rho}(\mathcal{HLE}))$ are equivalent. These equivalences follow essentially from results in [Mos02a, Dia08], which show that switching between institution morphisms and comorphisms as in Thm. 4.1, and so between indexed institutions and indexed coinstitutions, does not affect the resulting Grothendieck institution that can be built.

More explicitly, focused heterogeneous specifications that one can build as in Def. 3.1 using an institution morphism coincide (up to equivalence) with those one can build using the institution comorphism determined by the left adjoint to the signature functor of the institution morphism, and vice versa. Moreover, in each case, heterogeneous specification morphisms determine each other (generalising the original result of [AF96] for theories).

Lemma 4.2. *Let μ and ρ be as in Thm. 4.1, with $L(\mu) = \rho$ and $R(\rho) = \mu$. Consider specifications $SP \in Spec_{\mathcal{I}}$, $SP' \in Spec_{\mathcal{I}'}$ with $Sig[SP] = \Sigma$ and $Sig[SP'] = \Sigma'$. Then:*

- $SP'|_{\mu} \equiv (SP'|_{\varepsilon_{\Sigma'}})|_{\rho}^{\mu(\Sigma')}$
- $\rho(SP) \equiv \mu(\eta_{\Sigma}(SP))^{\rho(\Sigma)}$
- $\mu(SP)^{\Sigma'_0} \equiv \varepsilon_{\Sigma'_0}(\rho(SP))$*, for any* $\Sigma'_0 \in |\mathbf{Sign}'|$ *with* $\mu^{Sign}(\Sigma'_0) = \Sigma$,
- $SP'|_{\rho}^{\Sigma_0} \equiv (SP|_{\mu})|_{\eta_{\Sigma_0}}$*, for any* $\Sigma_0 \in |\mathbf{Sign}|$ *with* $\rho^{Sign}(\Sigma_0) = \Sigma'$.

Moreover, if $\sigma\colon \Sigma \to \mu^{Sign}(\Sigma')$ and $\sigma'\colon \rho^{Sign}(\Sigma) \to \Sigma'$ are signature morphisms corresponding to each other under bijection given by the adjunction between signature categories (i.e., such that $\eta_{\Sigma} ; \mu^{Sign}(\sigma') = \sigma$) then $\langle \mu, \sigma\rangle$ is a heterogeneous specification morphism $\langle \mu, \sigma\rangle\colon SP \to SP'$ iff $\langle \rho, \sigma'\rangle$ is a heterogeneous specification comorphism $\langle \rho, \sigma'\rangle\colon SP \to SP'$.

Overall this means that then when in a heterogeneous logical environment the institution morphisms or comorphisms link signature categories by adjunctions, we can gracefully make the environment uniform by replacing institution morphisms by the corresponding comorphisms or, respectively, by replacing institution comorphisms by the corresponding morphisms, with losing neither specifications that can be built nor heterogeneous (co)morphisms between them.

5 Distributed Specifications

Heterogeneity of the logical environment was used in focused heterogeneous specifications to build various parts of specifications in various logical systems (institutions) and then put them together to end up in one logical system, where the models of interest are. However, quite often, for instance in UML [BRJ98], heterogeneous specifications are presented rather differently, by simply giving a number of specifications in various logical systems, and then (implicitly or explicitly) linking them with each other to ensure the expected compatibility properties. This leads to the idea of *distributed specifications* which we will present in this section; see also [CKTW08] for an earlier sketch of this to provide an understanding of distributed heterogeneous UML specifications.

5.1 Distributed Specifications and Their Models

We will work in the context of a *specification frame*: a category **Spec** of (abstract) specifications with semantics given by a model functor $\mathbf{Mod}\colon \mathbf{Spec}^{op} \to \mathbf{Set}$. The terminology follows [CBEO99], the concept appeared earlier as "specification logic" in [EBCO92, EBO93]). As before, functions $\mathbf{Mod}(\sigma)$, for $\sigma\colon SP \to SP'$ in **Spec**, will be called reducts and denoted by $\_|_\sigma$. Moreover, specification frames can be linked by morphisms and comorphisms much in the same way as institutions, by just leaving out the sentence translation component.

For a while we will not need to discuss how such a specification frame was built: for instance, it may be the category of specifications built in an institution (which is perhaps the prime example) or a heterogeneous category of specifications built over a heterogeneous logical framework (which are examples of interest here).

Definition 5.1. *Let* $\mathcal{F} = \langle \mathbf{Spec}, \mathbf{Mod}\colon \mathbf{Spec}^{op} \to \mathbf{Set}\rangle$ *be a specification frame. A* distributed specification *in* $\mathcal{F}$ *is a collection of specifications linked by specification morphisms, that is, a diagram* $DSP\colon \mathcal{G} \to \mathbf{Spec}$ *in* **Spec**.

A distributed model *of* DSP *is a family of models* $\langle M_n\rangle_{n\in|\mathcal{G}|}$ *that is* compatible with morphisms in DSP *(i.e., for each edge* $e\colon n \to m$ *in* $\mathcal{G}$, $M_m|_{DSP(e)} = M_n$*) and such that for each node* $n \in |\mathcal{G}|$, $M_n \in \mathbf{Mod}(DSP(n))$*. We write* $Mod[DSP]$ *for the collection of all such distributed models of* DSP.

A specification morphism between distributed specifications $DSP\colon \mathcal{G} \to \mathbf{Spec}$ *and* $DSP'\colon \mathcal{G}' \to \mathbf{Spec}$ *is a pair* (F, τ)*, where* $F\colon \mathcal{G} \to \mathcal{G}'$ *is a functor, and* $\tau\colon DSP \to F\,; DSP'$ *a natural transformation. Such morphisms compose as usual, which yields the category* $\mathbf{DSpec}(\mathcal{F})$ *of distributed specifications in* $\mathcal{F}$.

Distributed model reducts w.r.t. such morphisms are defined in the obvious way: for $\mathcal{M}' = \langle M'_n\rangle_{n\in|\mathcal{G}'|} \in Mod[DSP']$, $\mathcal{M}'|_{(F,\tau)} = \mathcal{M}$*, where* $\mathcal{M} = \langle M'_{F(m)}|_{\tau_m}\rangle_{m\in|\mathcal{G}|} \in Mod[DSP]$.

This defines a new specification frame $\mathcal{DSP}(\mathcal{F})$ *of distributed specifications in* $\mathcal{F}$ *and their distributed models.*

Similar concepts were introduced for instance already in [Cla93]. The definition of $\mathcal{DSP}(\mathcal{F})$ resembles the construction of the institution of structured theories in [DM03], but differs from it somewhat by using whole specifications (linked by specification morphisms) as the building blocks for our distributed specifications, whereas the institution of structured theories relies on the use of collections of sentences distributed over signature diagrams.

Given the notion of a distributed specification and its class of models, many concepts and terminology carry over from standard to distributed specifications. For instance, consistency: a distributed specification DSP is *consistent* if $Mod[DSP] \neq \emptyset$.

5.2 Removing Distributivity

The literature so far focused largely on specification frames that are homogeneous in some sense, for instance arise as the category of theories or of specifications built in a single institution, with the usual semantics. In such a case, the following fact will often apply and could be used to diminish the role of distributed specifications by using a corresponding standard (colimit) specification.

Proposition 5.2. *Let* $\mathcal{F} = \langle \mathbf{Spec}, \mathbf{Mod} \colon \mathbf{Spec}^{op} \to \mathbf{Set} \rangle$ *be a (finitely)* exact *specification frame — that is,* **Spec** *is (finitely) cocomplete and* **Mod** *preserves (finite) limits.*[8]

Then for any (finite) distributed specification $DSP \colon \mathcal{G} \to \mathbf{Spec}$ *in* $\mathcal{F}$*, there exists a (colimit of DSP) specification* $SP \in |\mathbf{Spec}|$ *with specification morphisms* $\langle \iota_n \colon DSP(n) \to SP \rangle_{n \in |\mathcal{G}|}$ *such that each model* $M \in Mod[SP]$ *determines uniquely a distributed model of DSP,* $\langle M|_{\iota_n} \rangle_{n \in |\mathcal{G}|} \in Mod[DSP]$*, and vice versa: each distributed model* $\langle M_n \rangle_{n \in |\mathcal{G}|} \in Mod[DSP]$ *determines a unique model* $M \in Mod[SP]$ *such that* $M_n = M|_{\iota_n}$ *for* $n \in |\mathcal{G}|$*.*

This construction can easily be turned into both a morphism and a comorphism of specification frames:

Proposition 5.3. *Let* $\mathcal{F} = \langle \mathbf{Spec}, \mathbf{Mod} \colon \mathbf{Spec}^{op} \to \mathbf{Set} \rangle$ *be an exact specification frame. Then there is a specification frame (co)morphism* $Colim \colon DSP(\mathcal{F}) \to \mathcal{F}$*, taking a distributed specification to its colimit, that is an isomorphism on model classes.*

A similar fact, although not stated explicitly there, is already present in the proof of Thm. 20 in [DM03].

The assumption necessary for Props. 5.2 and 5.3 holds for instance for specification frames given by the categories of theories or of specifications (closed under translation and union) built in any (finitely) *exact* institution, where the category of signatures is (finitely) cocomplete and the model functor preserves (finite) limits, see [GB92, ST88a, DGS93]. It is well-known that practically all institutions that capture many-sorted logics are in fact exact. For single-sorted case, the model functors tend not to preserve coproducts, but typically such institutions are *semi-exact* (signature pushouts exists and the model functor maps them to pullbacks in **Set**). Then the resulting specification frames are semi-exact in the analogous sense, which is enough to establish the fact as above for finite connected distributed specifications.

However, this is quite in contrast with specification frames typically arising in heterogeneous logical environments, where (semi-)exactness is very rare. What one essential would need then is the (semi-)exactness of Grothendieck institutions built over (uniform) heterogeneous logical environments. See [Dia02] for results than ensure this for the morphism-uniform case, and [Mos02a] for the comorphism-uniform case. Unfortunately, albeit mathematically interesting and elegant, these results tend to rely on assumptions that are rarely met by the environments arising in practice — for instance, they require the shape of the heterogeneous logical environment to be (co)complete. This essentially would imply that in our logical environment there already is a single "maximal" institution capable of expressing all the specifications built in other institutions in

[8] That is, **Mod** maps (finite) colimits in **Spec** to limits in **Set**.

the environment via a unique representation. For example, in the Heterogeneous Tool Set HETS [MML07], there is no such "maximal" institution, rather, there are "local maxima", like the logic of Isabelle/HOL, which is used to encode many other logics. But even when restricting to a subgraph of logics represented in Isabelle/HOL, each logic is typically represented in it in more than one way, and so this is not a colimit (indeed, a colimit would have to make identifications that turn it into a rather artificial institution). Moreover, not all of the comorphisms involved in HETS are exact, but this would be needed to make the Grothendieck institution exact.

This is why distributed specifications become of real interest and relevance in the context of heterogeneous logical environments.

As stated above, the exactness assumption of Prop. 5.3 is unrealistically strong. A somewhat more realistic assumption is the following:

Definition 5.4. *A specification frame* $\mathcal{F} = \langle \mathbf{Spec}, \mathbf{Mod}\colon \mathbf{Spec}^{op} \to \mathbf{Set}\rangle$ *is* quasi-exact *if each diagram* $D\colon \mathcal{G} \to \mathbf{Spec}$ *has a cocone* $\langle \iota_n\colon D(n) \to SP\rangle_{n\in|\mathcal{G}|}$ *that, moreover, is weakly amalgamable. The latter means that any compatible family of models* $\langle M_n \in \mathbf{Mod}(D(n))\rangle_{n\in|\mathcal{G}|}$ *can be amalgamated to a (not necessarily unique) model* $M \in \mathbf{Mod}(SP)$ *with* $M|_{\iota_n} = M_n$ *for* $n \in |\mathcal{G}|$.

This notion leads to a mathematically less elegant, but practically somewhat more applicable variant of Prop. 5.3:

Proposition 5.5. *Let* $\mathcal{F} = \langle \mathbf{Spec}, \mathbf{Mod}\colon \mathbf{Spec}^{op} \to \mathbf{Set}\rangle$ *be a quasi-exact specification frame. Let* $Discr(\mathcal{DSP}(\mathcal{F}))$ *be the sub-specification frame of* $\mathcal{DSP}(\mathcal{F})$ *where all non-identity specification morphisms are removed. Then there is at least one specification frame comorphism* $WeakAmalg\colon Discr(\mathcal{DSP}(\mathcal{F})) \to \mathcal{F}$ *that is surjective on models, taking a distributed specification to the tip of a weakly amalgamable cocone.*

Note that the need of the move to discrete specification categories (via $Discr(\_)$) is caused by the construction not being functorial.

5.3 Implementing Distributed Specifications

Working in a specification frame $\mathcal{F} = \langle \mathbf{Spec}, \mathbf{Mod}\rangle$, in this section we adapt to distributed specifications the standard view of the process of systematic software development, as presented using implementation steps involving constructors, see [ST88b, ST97]. Recall that for (standard) specifications SP and SP', a *constructor* from SP' to SP is simply a function $\kappa\colon Mod[SP'] \to Mod[SP]$. Given such a constructor, we say that SP' *implements* SP *via* κ, written $SP \underset{\kappa}{\rightsquigarrow} SP'$.[9] To generalise this to distributed specifications, we also have to "distribute" the constructor:

[9] The definition in the framework of an institution is a bit more delicate: κ is a partial function between model classes over the signatures of SP' and SP, respectively, and then for $SP \underset{\kappa}{\rightsquigarrow} SP'$ one requires that on models in $Mod[SP']$, κ is defined and yields models in $Mod[SP]$.

Definition 5.6. *To implement a distributed specification $DSP\colon \mathcal{G} \to$ **Spec** by $DSP'\colon \mathcal{G}' \to$ **Spec**, one needs to provide a* covering function $f\colon |\mathcal{G}| \to |\mathcal{G}'|$ *and a family of* constructors $K = \langle \kappa_n\colon Mod[DSP'(f(n))] \to Mod[DSP(n)]\rangle_{n\in|\mathcal{G}|}$.

Then DSP' implements *DSP* via *f* and *K, written $DSP \underset{f,K}{\rightsquigarrow} DSP'$, if for each distributed model $\langle M_{n'}\rangle_{n'\in|\mathcal{G}'|} \in Mod[DSP']$, the family $\langle \kappa_n(M_{f(n)})\rangle_{n\in|\mathcal{G}|}$ is compatible with morphisms in DSP.*

Of course, if $DSP \underset{f,K}{\rightsquigarrow} DSP'$ then for each distributed model $\langle M_{n'}\rangle_{n'\in|\mathcal{G}'|} \in Mod[DSP']$, $\langle \kappa_n(M_{f(n)})\rangle_{n\in|\mathcal{G}|}$ is a model of DSP.

As can be seen directly from the definition, to establish $DSP \underset{f,K}{\rightsquigarrow} DSP'$ we first have to show that for all $n \in |\mathcal{G}|$, $DSP(n) \underset{\kappa_n}{\rightsquigarrow} DSP'(f(n))$ (which is just as in the implementation steps for standard specifications) and then add that the constructors in K on the respective models from any family satisfying (and hence compatible with) DSP' yield a family of models compatible with DSP. The latter requirement is essentially new and, in general, may require new proof techniques. However, in some simple cases it can be shown using standard categorical reasoning:

Proposition 5.7. *Consider any distributed specifications $DSP\colon \mathcal{G} \to$ **Spec** and $DSP'\colon \mathcal{G}' \to$ **Spec**, and let $(F,\tau)\colon DSP \to DSP'$ be a specification morphism in* **DSpec**$(\mathcal{F})$. *Then $DSP \underset{f,K}{\rightsquigarrow} DSP'$, where f is the object (node) part of F and $K = \langle \_|_{\tau_n}\rangle_{n\in|\mathcal{G}|}$ is the family of reducts w.r.t. τ_n, $n \in |\mathcal{G}|$.*

Note that the above is just an instance (in $\mathcal{DSP}(\mathcal{F})$) of the well-known general fact that for any specification morphism $\sigma\colon SP \to SP'$, the reduct w.r.t. σ yields a correct implementation of SP by SP', $SP \underset{\_|_\sigma}{\rightsquigarrow} SP'$, cf. [ST88b].

In the case captured by the proposition above, we ensure that the family of constructors given as reducts w.r.t. specification morphisms preserves compatibility of model families in the most simple and expected categorical way. The use of reducts as constructors here may seem very restrictive, but in fact, if one works in a sufficiently rich specification frame, for instance based on institutions with "derived" signature morphisms, then reducts may cover essentially all relevant constructors. Here, a very general concept of a derived signature morphism may be used. Informally, a derived signature morphism $\delta\colon \Sigma \to \Sigma'$ maps each symbol in Σ to its definition in terms of symbols in Σ'; then for any Σ'-model which interprets the symbols in Σ', its reduct w.r.t. δ is a Σ-model built using the definitions for the symbols in Σ given by δ. In an institution-independent setting, a derived signature morphism could be defined to be an ordinary signature into a definitional extension (see Def. 5.8 below).

Finally, we should stress here that the above notion of implementation covers all possible (and necessary in the development process) changes. First, as usual, individual specifications may be refined, by adding more requirements and "implementation decisions". Second, the structure of the distributed specification may change here: ultimately, we may even arrive at a single standard specification. Finally, in the case when we are working in the heterogeneous category of specifications built in a heterogeneous logical framework, institutions in which individual specifications are built may be changed as well!

5.4 Comparing Distributed Specifications

For usual (homogeneous or focused heterogeneous) specifications, we have introduced the basic notion of equivalence as a way to identify specifications with the same model classes. For distributed specifications, this cannot be so simple. The point is that, very informally, some of the nodes in a distributed specification may play only an auxiliary role, so that in any distributed model of such a distributed specification, the individual models given for such nodes are always uniquely determined by the rest of the family. Effectively, such nodes and their corresponding individual models may be disregarded when comparing distributed models of distributed specifications. This leads to a generalisation of the notion of equivalence of specifications in any specification frame $\mathcal{F} = \langle \mathbf{Spec}, \mathbf{Mod} \rangle$.

Definition 5.8. *A specifications SP' is a* definitional extension *of a specification SP along a specification morphism $\sigma\colon SP \to SP'$ if any SP-model has a unique σ-expansion to an SP'-model, i.e., the reduct $\_|_\sigma\colon Mod[SP'] \to Mod[SP]$ is a bijection.*

Definition 5.9. *Two specifications SP_1 and SP_2 are* pre-equivalent*, written $SP_1 \cong SP_2$, iff there is a common definitional extension SP of SP_1 and SP_2. Derived equivalence*[10] *of specification is defined to be transitive closure of pre-equivalence.*

Proposition 5.10. *Derived equivalence is an equivalence. In exact specification frames, pre-equivalence is transitive, hence derived equivalence and pre-equivalence coincide.*

Now, two distributed specifications $DSP_1\colon \mathcal{G}_1 \to \mathbf{Spec}$ and $DSP_2\colon \mathcal{G}_2 \to \mathbf{Spec}$ are pre-equivalent (in $\mathcal{DSP}(\mathcal{F})$) iff there is a distributed specification $DSP\colon \mathcal{G} \to \mathbf{Spec}$ with distributed specification morphisms $(F_1, \tau_1)\colon DSP_1 \to DSP$ and $(F_2, \tau_2)\colon DSP_2 \to DSP$ such that DSP is a definitional extension of DSP_1 along (F_1, τ_1) and of DSP_2 along (F_2, τ_2). In other words, any distributed model $\mathcal{M}_1 \in Mod[DSP_1]$ extends then uniquely along (F_1, τ_1) to a distributed model of DSP, which in turn reduces (uniquely) w.r.t. (F_2, τ_2) to a distributed model of DSP_2, and vice versa, yielding a "natural" bijection between distributed models of DSP_1 and DSP_2, respectively.

5.5 Distributed Heterogeneous Specifications

The machinery developed above may now be employed to deal with distributed heterogeneous specifications in a heterogeneous logical environment $\mathcal{HLE}$,

[10] This terminology is meant to reflect the comments concerning derived specification morphisms in Sect. 5.3 above.

understood as collections of specifications in $\mathcal{HLE}$ linked by (generalised) heterogeneous specification morphisms. Formally, such a distributed heterogeneous specification is just a distributed specification in the sense of Def. 5.1 in the specification frame $\mathcal{HSF}(\mathcal{HLE}) = \langle \mathbf{Spec}(\mathcal{HLE}), \mathbf{Mod}(\mathcal{HLE})\rangle$, given by Def. 3.4. This yields the specification frame $\mathcal{DSP}(\mathcal{HSF}(\mathcal{HLE}))$ of distributed heterogeneous specifications. We can extend it further to an institution:

Definition 5.11. *Let $\mathcal{HLE}$ be a heterogeneous logical environment. Then the institution $\mathcal{DHSI}(\mathcal{HLE})$ has the category $\mathbf{DSpec}(\mathcal{HSF}(\mathcal{HLE}))$ of distributed heterogeneous specifications as its "signature" category; the model functor is inherited from $\mathcal{DSP}(\mathcal{HSF}(\mathcal{HLE}))$. Given a distributed specification $DSP\colon \mathcal{G} \to \mathbf{Spec}(\mathcal{HLE})$ in $|\mathbf{DSpec}(\mathcal{HSF}(\mathcal{HLE}))|$, a DSP-sentence is of the form $\langle n, \varphi\rangle$ for $n \in |\mathcal{G}|$ and $\varphi \in \mathbf{Sen}_{\mathcal{I}_n}(\Sigma_n)$, where $Sig[DSP(n)] = \langle \mathcal{I}_n, \Sigma_n\rangle$. A distributed model $\langle M_k\rangle_{k\in|\mathcal{G}|} \in Mod[DSP]$ satisfies such a sentence $\langle n, \varphi\rangle$ if $M_n \models \varphi$ in $\mathcal{I}_n$. For a distributed specification morphism $(F, \tau)\colon (DSP\colon \mathcal{G} \to \mathbf{Spec}(\mathcal{HLE})) \to (DSP'\colon \mathcal{G}' \to \mathbf{Spec}(\mathcal{HLE}))$ in $\mathbf{DSpec}(\mathcal{HSF}(\mathcal{HLE}))$, translation of such a sentence is given by $(F, \tau)(\langle n, \varphi\rangle) = \langle F(n), \tau_n(\varphi)\rangle$, where for each $n \in |\mathcal{G}|$, the translation $\tau_n(\varphi)$ of φ along the generalised heterogeneous specification morphism τ_n is defined by composing in the natural order the translations along the signature morphism and the institution (co)morphism involved in τ_n. The satisfaction condition follows easily.*

$\mathcal{DHSI}(\mathcal{HLE})$ leads, in the expected way, to a notion of logical *consequences* of a distributed specification (sentences that hold in all models of the distributed specifications). Spelling this out: for $DSP\colon \mathcal{G} \to \mathbf{Spec}(\mathcal{HLE})$, $n \in |\mathcal{G}|$, $Sig[DSP(n)] = \langle \mathcal{I}_n, \Sigma_n\rangle$, and $\varphi \in \mathbf{Sen}_{\mathcal{I}_n}(\Sigma_n))$, we say that $\langle n, \varphi\rangle$ is a consequence of DSP, written $DSP \models_n \varphi$, iff for all distributed models $\langle M_k\rangle_{k\in|\mathcal{G}|} \in Mod[DSP]$, $M_n \models_{\mathcal{I}_n,\Sigma_n} \varphi$. Note that such consequences include, in general properly, the usual consequences of the individual specifications involved.

$\mathcal{DHSI}(\mathcal{HLE})$ also gives a notion of a *(distributed heterogeneous) theory* of a distributed heterogeneous specification: for $DSP\colon \mathcal{G} \to \mathbf{Spec}(\mathcal{HLE})$ and $n \in |\mathcal{G}|$ with $Sig[DSP(n)] = \langle \mathcal{I}_n.\Sigma_n\rangle$, we have $Th(DSP)(n) = \{\varphi \in \mathbf{Sen}_{\mathcal{I}_n}(\Sigma_n) \mid DSP \models_n \varphi\}$. Then for each $e\colon m \to n$ in $\mathcal{G}$, the signature morphism $DSP(e)$ is a theory morphism, $DSP(e)\colon Th(DSP)(m) \to Th(DSP)(n)$, so that we get a diagram in the usual (heterogeneous) category of theories of $\mathcal{HLE}$. Note though that the individual theories $Th(DSP)(n)$ need not be in general finitely presentable in $\mathcal{I}_n$, even if all the individual specifications in DSP are finite presentations.

An interesting alternative way to present distributed heterogeneous specifications would be to first define an institution that differs from $\mathcal{DHSI}(\mathcal{HLE})$ by taking as signatures diagrams in the category $\mathbf{Sign}(\mathcal{HLE})$ of heterogeneous signatures (which would coincide with the institution of structured theories, as defined in [DM03], built for the Bi-Grothendieck institution of $\mathcal{HLE}$). It is easy to see that each distributed heterogeneous specification could be then obtained as a structured specification built in this institution. Moreover, although structured specifications in this institution would also correspond to families of specifications with their signatures linked by signature morphisms that are not necessarily specification morphisms, it can be shown that for such structured specifications, at least when their signatures are finite directed diagrams, we can always give an equivalent distributed heterogeneous specification as defined here.

If the specification frame $\mathcal{HSF}(\mathcal{HLE})$ is quasi-exact, Prop. 5.5 can be used for $\mathcal{DSP}(\mathcal{HSF}(\mathcal{HLE}))$. Moreover, it can be checked that the specification frame morphism *WeakAmalg* defined on $Discr(DSP(\mathcal{HSF}(\mathcal{HLE})))$ there can be extended to an institution comorphism from $\mathcal{DHSI}(\mathcal{HLE})$ to the Bi-Grothendieck institution build on $\mathcal{HLE}$. The importance of this fact lies in the possibility of transferring logical consequence:

Proposition 5.12. *For any institution comorphism $\rho\colon \mathcal{I} \to \mathcal{I}'$ that is surjective on models, and set of Σ-sentences $\Gamma \cup \{\varphi\}$ in $\mathcal{I}$, we have:*

$$\Gamma \models^{\mathcal{I}}_{\Sigma} \varphi \textit{ iff } \rho^{Sen}_{\Sigma}(\Gamma) \models^{\mathcal{I}'} \rho^{Sen}_{\Sigma}(\varphi)$$

That is, given any proof calculus or theorem prover capturing logical consequence in $\mathcal{I}'$, we can re-use it to capture logical consequence in $\mathcal{I}$. When combined with Prop. 5.5, this means that for quasi-exact (Bi-Grothendieck) institutions, logical consequence for distributed heterogeneous specifications can be reduced to logical consequence for focused heterogeneous specifications.

In a heterogeneous setting, the property of quasi-exactness for the specification frame (or institution) of heterogeneous specifications remains quite a strong requirement. However, if one restricts attention to distributed specifications with particular shapes of diagram (namely, so-called connected finitely bounded inf-complete diagrams), then it can be obtained under rather realistic assumptions. For details, see Corollaries 30 and 31 of [CM08].

Finally, we can return in this setting to the issue of making heterogeneous logical environments uniform. It turns out that for any heterogeneous logical environment $\mathcal{HLE}$, even though the heterogeneous specification categories $\mathbf{Spec}(\mathcal{HLE})$, $\mathbf{Spec}(span^{\mu}(\mathcal{HLE}))$ and $\mathbf{Spec}(span^{\rho}(\mathcal{HLE}))$ are quite different, the distributed specifications we can build in each of these categories are essentially the same.

Proposition 5.13. *Given a heterogeneous logical environment $\mathcal{HLE}$, consider any distributed heterogeneous specification $DSP \in |\mathbf{DSpec}(\mathcal{HLE})|$. There exists then a comorphism-uniform distributed heterogeneous specification $DSP^{\rho} \in |\mathbf{DSpec}(span^{\rho}(\mathcal{HLE}))|$ such that $DSP^{\rho} \cong DSP$. Similarly, there is a morphism-uniform distributed heterogeneous specification $DSP^{\mu} \in |\mathbf{DSpec}(span^{\mu}(\mathcal{HLE}))|$ such that $DSP^{\mu} \cong DSP$.*

The proof is related to that of Thm. 11 of [Mos03], relying on Prop. 3.7. For instance, consider an institution morphism $\mu\colon \mathcal{I} \to \mathcal{I}'$, where $span(\mu)$ is $\mathcal{I} \xleftarrow{\rho_{\mu,1}} \mathcal{I}'_0 \xrightarrow{\rho_{\mu,2}} \mathcal{I}'$, specifications $SP \in Spec_{\mathcal{I}}$, with $Sig[SP] = \Sigma$, and $SP' \in Spec_{\mathcal{I}'}$, and a signature morphism $\sigma\colon Sig[SP'] \to \mu^{Sign}(\Sigma)$. Then a heterogeneous specification morphism $SP' \xrightarrow{\langle\mu,\sigma\rangle} SP$ in a distributed heterogeneous specification may be replaced by a sequence of heterogeneous specifications comorphisms $SP' \xrightarrow{\langle id,\sigma\rangle} \rho_{\mu,2}(SP|^{\Sigma}_{\rho_{\mu,1}}) \xleftarrow{\langle\rho_{\mu,2},id\rangle} SP|^{\Sigma}_{\rho_{\mu,1}} \xrightarrow{\langle\rho_{\mu,1},id\rangle} SP$.

6 Final Remarks

The sentence part of the institution morphisms and comorphisms has rarely played any role in the considerations in this paper (that is, after sentences have been used to build basic specifications). Consequently, we could replace the use of institution morphisms and comorphisms by institution semi-morphisms and semi-comorphisms, respectively (semi-(co)morphisms are just like (co)morphisms but without the translation of sentences, and hence without caring about the satisfaction at all, see [ST88b, Tar96]). With the obvious projection from the category of institutions and their (co)morphisms to the category of institutions and their semi-(co)morphisms, essentially all we presented here would be a special case of a formally more general (but in the presentation basically identical) development using semi-morphisms and semi-comorphisms. Of course, the sentences and satisfaction start matter when it comes to consideration of consequence and proofs in the framework presented here. Remarks on theories for distributed specifications and discussion of Prop. 5.5 in Sect. 5.5 give but the first hints in this direction. Then full institution morphisms and comorphisms provide considerably more possibilities then their "semi-" versions. A proof calculus for focused heterogeneous specifications has been developed in [Mos02a, Mos05]. Using Prop. 5.5, it can be extended to distributed heterogeneous specifications under suitable conditions using weakly amalgamable cocones, which are not unrealistic to be met in practice. The exact tuning of these conditions remains a topic for further research. In cases without weak amalgamation, probably there is no better way than to resolve the proof problems on a case-by-case basis, for each specific link between institutions.

A simple analysis of possible mutual directions of translations involved in maps between institutions leads to further notions of maps between institutions, as suggested in [Tar96] and then studied in [GR02] (see also [MW98]). In particular, when all translations go in the same direction, we obtain institution *forward morphisms*, and when both sentences and models are translated contravariantly w.r.t. signatures, we obtain *forward comorphisms*. It turns out that the span construction helps here again: with spans of morphisms, we can simulate forward (co)morphisms (as well as semi-(co)morphisms) much in the same way as we have been able to simulate comorphisms, see [Mos05] for details. (A similar remark holds for spans of comorphisms.) It may be a bit more difficult to bring into the picture institution (co)morphisms in their *theoroidal* versions, where signatures of one institution are mapped to theories, rather than just signatures, of the other institution [Mes89, GR02]. A technically easy way to achieve this is to add to the heterogeneous logical environment enough infrastructure to allow for expressing theoroidal institution (co)morphisms as plain institution (co)morphisms: for each institution $\mathcal{I}$, its institution of theories $\mathcal{I}^{th}$ needs to be added, along with the obvious morphism $\mathcal{I}^{th} \to \mathcal{I}$ and comorphism $\mathcal{I} \to \mathcal{I}^{th}$.[11]

[11] Even generalised theoroidal comorphisms in the sense of [Cod] can then be expressed as semi-comorphisms between institutions of theories.

While the general theory works also for this extended heterogeneous logical environment, it remains to be checked which properties of the heterogeneous logical environment are preserved under this extension, and whether the duplication of $\mathcal{I}$ into $\mathcal{I}$ and $\mathcal{I}^{th}$ can be eliminated, possibly using techniques of [Mos96]. At least it is clear that a theorem prover for $\mathcal{I}$ can easily be lifted to $\mathcal{I}^{th}$.

While non-uniform heterogeneous logical environments naturally arise and can be used in practice, we also offer two ways to make them uniform. The first way, via adjunctions between signature categories, leaves the resulting category of heterogeneous specifications essentially untouched. However, adjunctions are not always available. The second way, via the construction involving spans, is completely general, but leads to a certain modification of the category of heterogeneous specifications. While the same focused heterogeneous specifications can be expressed, we do not directly obtain the same heterogeneous specification (co)morphisms. Nevertheless, we can capture the proof obligations that the morphisms in the non-uniform environment carry by considering logically equivalent specification diagrams. The same method shows that making a heterogeneous logical environment uniform preserves (up to equivalence) the set of distributed heterogeneous specifications.

Another possibility would be to consider an even more general category of institutions, where both morphisms and comorphisms (as well as their semi- and forward versions) can be placed together. One obvious candidate could be based on a notion of institution *relational links*, where the categories of signatures are linked e.g. by distributors (also called profunctors) [Bor94], which are a relational version of functors. Then for any two related signatures, a relation between the sentences over them and a relation between models over them would be given, natural in the related signature morphisms. Generalising the satisfaction condition for institution (co)morphisms, we would of course require these relations to preserve satisfaction. Such relational links clearly compose and cover all kinds of maps between institutions we considered. Hence, in this way we would obtain a category of institutions with relational links between them, into which each of the categories of institutions considered so far could be faithfully embedded. However, as far as we can see, such a category brings little benefit: the notion so obtained seems a bit artificial, and does not ensure any of the expected properties (e.g., entailment is in general neither preserved nor reflected by relational links, the category is neither finitely complete nor finitely cocomplete, etc).

One consequence may be that we have to live with non-uniform environments, where the maps considered do not compose in general, and so we cannot view them simply as diagrams in a category of institutions. In fact, this is what is really happening in HETS [MML07, Mos05], where both institution morphisms and comorphisms are used, while the projection (via spans) to a comorphism-uniform environment is applied for theorem proving. Future work will apply this approach to the heterogeneous logical environment arising from UML (see [CKTW08] for initial promising steps in this direction).

Acknowledgements. Many thanks to the anonymous referees for detailed comments.

References

[AF96] Arrais, M., Fiadeiro, J.-L.: Unifying theories in different institutions. In: Haveraaen, M., Dahl, O.-J., Owe, O. (eds.) Abstract Data Types 1995 and COMPASS 1995. LNCS, vol. 1130, pp. 81–101. Springer, Heidelberg (1996)

[BG80] Burstall, R.M., Goguen, J.A.: The semantics of CLEAR, a specification language. In: Bjorner, D. (ed.) Abstract Software Specifications. LNCS, vol. 86, pp. 292–332. Springer, Heidelberg (1980)

[Bor94] Borceux, F.: Handbook of Categorical Algebra I. Cambridge University Press, Cambridge (1994)

[BRJ98] Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language User Guide. Addison-Wesley, Reading (1998)

[CBEO99] Cornelius, F., Baldamus, M., Ehrig, H., Orejas, F.: Abstract and behaviour module specifications. Mathematical Structures in Computer Science 9(1), 21–62 (1999)

[CKTW08] Cengarle, M.V., Knapp, A., Tarlecki, A., Wirsing, M.: A heterogeneous approach to UML semantics. In: Degano, P., De Nicola, R., Meseguer, J. (eds.) Concurrency, Graphs and Models. LNCS, vol. 5065, pp. 383–402. Springer, Heidelberg (2008)

[Cla93] Classen, I.: Compositionality of application oriented structuring mechanisms for algebraic specification languages with initial algebra semantics. Phd thesis, Technische Universität Berlin (1993)

[CM08] Codescu, M., Mossakowski, T.: Heterogeneous colimits. In: Boulanger, F., Gaston, C., Schobbens, P.-Y. (eds.) MoVaH 2008 Workshop on Modeling, Validation and Heterogeneity. IEEE press, Los Alamitos (2008)

[Cod] Codescu, M.: Generalized theoroidal institution comorphisms. This volume

[DGS93] Diaconescu, R., Goguen, J., Stefaneas, P.: Logical support for modularisation. In: Huet, G., Plotkin, G. (eds.) Logical Environments, pp. 83–130. Cambridge Univ. Press, Cambridge (1993)

[Dia98] Diaconescu, R.: Extra theory morphisms for institutions: Logical semantics for multi-paradigm languages. J. Applied Categorical Structures 6, 427–453 (1998)

[Dia02] Diaconescu, R.: Grothendieck institutions. J. Applied Categorical Structures 10, 383–402 (2002)

[Dia08] Diaconescu, R.: Institution-independent model theory. Birkhäuser, Basel (2008)

[DM03] Durán, F., Meseguer, J.: Structured theories and institutions. Theor. Comput. Sci. 309(1-3), 357–380 (2003)

[EBCO92] Ehrig, H., Baldamus, M., Cornelius, F., Orejas, F.: Theory of algebraic module specification including behavioral semantics and constraints. In: Nivat, M., Rattray, C., Rus, T., Scollo, G. (eds.) Algebraic Methodology and Software Technology AMAST 1991, Proc. 2nd Intl. Conf., Iowa City, 1991, Workshops in Computing, pp. 145–172. Springer, Heidelberg (1992)

[EBO93] Ehrig, H., Baldamus, M., Orejas, F.: New concepts of amalgamation and extension for a general theory of specifications. In: Bidoit, M., Choppy, C. (eds.) Abstract Data Types 1991 and COMPASS 1991. LNCS, vol. 655, pp. 199–221. Springer, Heidelberg (1993)

[GB92] Goguen, J.A., Burstall, R.M.: Institutions: Abstract model theory for specification and programming. Journal of the ACM 39(1), 95–146 (1992)

[GR02] Goguen, J.A., Rosu, G.: Institution morphisms. Formal Aspects of Compututing 13(3-5), 274–307 (2002)

[HHP93] Harper, R., Honsell, F., Plotkin, G.D.: A framework for defining logics. Journal of the ACM 40(1), 143–184 (1993)

[MDT09] Mossakowski, T., Diaconescu, R., Tarlecki, A.: What is a logic translation? In: Beziau, J.-Y. (ed.) Logica Universalis, Birkhäuser, Basel (to appear, 2009)

[Mes89] Meseguer, J.: General logics. In: Logic Colloquium 1987, pp. 275–329. North Holland, Amsterdam (1989)

[MML07] Mossakowski, T., Maeder, C., Lüttich, K.: The Heterogeneous Tool Set. In: Grumberg, O., Huth, M. (eds.) TACAS 2007. LNCS, vol. 4424, pp. 519–522. Springer, Heidelberg (2007)

[MOM02] Martí-Oliet, N., Meseguer, J.: Rewriting logic: roadmap and bibliography. Theor. Comput. Sci. 285(2), 121–154 (2002)

[Mos96] Mossakowski, T.: Different types of arrow between logical frameworks. In: Meyer auf der Heide, F., Monien, B. (eds.) ICALP 1996. LNCS, vol. 1099, pp. 158–169. Springer, Heidelberg (1996)

[Mos02a] Mossakowski, T.: Comorphism-based Grothendieck logics. In: Diks, K., Rytter, W. (eds.) MFCS 2002. LNCS, vol. 2420, pp. 593–604. Springer, Heidelberg (2002)

[Mos02b] Mossakowski, T.: Heterogeneous development graphs and heterogeneous borrowing. In: Nielsen, M., Engberg, U. (eds.) FOSSACS 2002. LNCS, vol. 2303, pp. 326–341. Springer, Heidelberg (2002)

[Mos03] Mossakowski, T.: Foundations of heterogeneous specification. In: Wirsing, M., Pattinson, D., Hennicker, R. (eds.) WADT 2003. LNCS, vol. 2755, pp. 359–375. Springer, Heidelberg (2003)

[Mos05] Mossakowski, T.: Heterogeneous Specification and the Heterogeneous Tool Set. Habilitation thesis, Universität Bremen (2005)

[MW98] Martini, A., Wolter, U.: A single perspective on arrows between institutions. In: Haeberer, A.M. (ed.) AMAST 1998. LNCS, vol. 1548, pp. 486–501. Springer, Heidelberg (1998)

[NPW02] Nipkow, T., Paulson, L.C., Wenzel, M.: Isabelle/HOL — A Proof Assistant for Higher-Order Logic. LNCS, vol. 2283. Springer, Heidelberg (2002)

[PF06] López Pombo, C., Frias, M.F.: Fork algebras as a sufficiently rich universal institution. In: Johnson, M., Vene, V. (eds.) AMAST 2006. LNCS, vol. 4019, pp. 235–247. Springer, Heidelberg (2006)

[ST88a] Sannella, D., Tarlecki, A.: Specifications in an arbitrary institution. Information and Computation 76, 165–210 (1988)

[ST88b] Sannella, D., Tarlecki, A.: Toward formal development of programs from algebraic specifications: Implementations revisited. Acta Informatica 25, 233–281 (1988)

[ST97] Sannella, D., Tarlecki, A.: Essential concepts of algebraic specification and program development. Formal Aspects of Computing 9, 229–269 (1997)

[Tar87] Tarlecki, A.: Institution representation. Unpublished note, Dept. of Computer Science, University of Edinburgh (1987)

[Tar96] Tarlecki, A.: Moving between logical systems. In: Haveraaen, M., Dahl, O.-J., Owe, O. (eds.) Abstract Data Types 1995 and COMPASS 1995. LNCS, vol. 1130, pp. 478–502. Springer, Heidelberg (1996)

[Tar00] Tarlecki, A.: Towards heterogeneous specifications. In: Gabbay, D., de Rijke, M. (eds.) Frontiers of Combining Systems 2. Studies in Logic and Computation, pp. 337–360. Research Studies Press (2000)

[Tar03] Tarlecki, A.: Abstract specification theory: An overview. In: Broy, M., Pizka, M. (eds.) Models, Algebras, and Logics of Engineering Software. NATO Science Series — Computer and System Sciences, vol. 191, pp. 43–79. IOS Press, Amsterdam (2003)