TIERED LOGIC FOR AGENTS

Rosalito Perez Cruz, John Newsome Crossley
Faculty of Information Technology, Monash University, Australia
lcruz@sleekersoft.com, John.Crossley@infotech.monash.edu.au

Keywords: Agents, Logics for agent systems, Ontologies and agent systems.

Abstract: We introduce a new kind of logic for agents in different locaities, which works in tiers or layers.
At the base are local worlds with their own logic. Above them is a global logic that takes statements from the local worlds and combines them. This allows communications between the different localities, even though they have different languages.
We give a basic example using first order logic as the local logic and propositional calculus at the global level. As a more sophisticated example we use the algebraic specification language CASL and then the locations have specifications associated with them. Moreover we then permit the combination of such specifications according to the architectural specifications of CASL.
Although we only consider two layers in the present paper, we see no reason why the approach should not be extended to any finite number of tiers.
We prove soundness and completeness proofs for our logics.

1 INTRODUCTION

It is an established fact that the work of agents in a multi-agent system is enhanced by the presence of ontologies. For an ontology to be useful, people will have to agree to its terms and usage. It has to be done in the spirit of sharing. However, we know from human nature that people will not agree nor use something like an ontology consistently. Thus the idea of arriving at a *global ontology* for a domain of application appears to be wishful thinking. For this reason, it seems more appropriate to conceive of pockets of communities sharing their ontologies and to cope with any differences they may have. It is a much better and a realistic assumption to think of communities adopting a number of ontologies, each created within their *local* community.

We shall adopt an approach which contextualizes the logics that support these ontologies, and thereby point a way for agent systems to deal with heterogenous ontologies. We shall describe two logics: a first order logic of contexts, **Tiered FOL**, which we use as a basis, then we extend this technique to a language **Tiered CASL**, where the contexts are architectural specifications in the Common Algebraic Specification Language, CASL, see (Casl, 2001). We prove completeness results for both these logics.

1.1 Context and reasoning

In the field of AI and, by association, Logic, there are two major styles of embedding contexts in a logical system. The first is in the Propositional Logic of Context (PLC) of Buvac-Mason (Buvač, 1993; Buvač et al., 1995). Buvač (Buvač, 1993) developed a system, which has since been enhanced by a number of people, see, e.g., (Ghidini and Serafini, 1998; Akman and Surav, 1996; Bouquet et al., 2004; Giunchiglia and Ghidini, 2000; Serafini and Bouquet, 2004).

There is also the Local Models Sematics/MultiContext Systems (LMS/MCS) of (Giunchiglia and Ghidini, 2000; Ghidini and Serafini, 1998). By no means do we imply that these are the only two possible styles, for there are others such as in (Akman and Surav, 1996). However, when it comes to DL and ontologies, PLC or LMS would be the best candidates to choose from.

One example of an LMS approach in the field of DL is that taken by Borgida and Serafini. They describe their formulation of a Distributed DL (DDL) in (Borgida and Serafini, 2002). A major problem has been the transfer of knowledge between localities. Bridge rules (see below, Section 2.5) were introduced in (Ghidini and Serafini, 1998) but the form of the rules was very limited and only allowed the (partial) identification of one concept as a subset of another in a different context. The idea is to align ontologies (or knowledge bases), say IS_1 and IS_2 by expressing the connections between them. The intent is that the logical system should allow the relationship of concepts to be stated in the said ontologies. An example of this is the subsumption of concepts between the ontologies. To do this, Borgida and Serafini extend the usual DL formulation, taking their cue from the Distributed First Order Logic (DFOL) of (Ghidini and Serafini, 1998). In their formulation a DL statement is preceded by a label that stands for the ontology, so they write $1 : E$ to say that E is a concept in ontology 1. Then they state *bridge rules*, which relate a concept in one ontology to another one in a different ontoloy (see (Borgida and Serafini, 2002)). Thus they have semantic mappings in the system.

1.2 An intuitive picture

In global (natural language) discourse one often sees or hears statements in a foreign language used in the middle of something in the local language, for example in a television broadcast where the spoken foreign language is accompanied with subtitles. References may then neeed to be changed or at least clarified. Consider the following two assertions:

> "Le président à dit qu'il n'y a aucune arme de destruction de masse en Irak."[1]
> "The President said that there are weapons of mass destruction in Iraq."

Here the references to Iraq are to the same country, however the reference to the president refers, in the first case, to the French one, and in the second, to the US President.[2] Note that there is no contradiction between the two quotations.[3]

In reporting these utterances in the media in the two countries there would be an indication of the locality. Thus we might have found in the USA: "The President *of France* said that there are no weapons of mass destruction in Iraq," and in France: "Aux États Unis, le Président a dit qu'il y a armes de destruction de masse en Irak". Finally, in a third country: "In the USA, the President said there are weapons of mass destruction in Iraq, but in France, its President said there are no weapons of mass destruction." Semantically we understand these utterances because we tag each utterance with its context or, as we shall say, "locality", in these cases, France and the USA, respectively. Then we interpret them according to the appropriate context.

When we turn to agents in contexts then we again have the problem of them communicationg with each other across different languages. This paper is an attempt to provide a basic method of formalizing such situations.

This paper gives the first presentation of what we call "Tiered Logic", which allows the inclusion of much more powerful bridge rules. In our logic statements made in a local language are tagged with that local context and then become "atomic" statements or basic propositions in a higher tier of what we call the *global* logic. With bridge rules any statement in one locality can have consequences in another. So information can be conveyed, or even translated, from one locality to another.

We provide soundness and completeness proofs for two varieties of our underlying idea of tiered logic. In order to simplify our presentation we assume that all our localities have the same underlying logic, but different languages. This restriction is not essential but to present a completely general approach would be notationally horrendous. The complications in our presentation come from the interactions between the tiers, since when a sentence from one context is used in a different context, one has to refer back to the first context in order to determine the semantics. Thus for a sentence φ from a locality l, a basic global statement φ^l is interpreted as being true globally if and only if the statement φ is true in the locality l. More complicated global statements build on this using the usual semantics of propositional calculus.

From the logic of Saša Buvač, see e.g. (Buvač, 1993) we use his notion of *flatness* (see below, Section 2.2). This notion reflects the idea that once a statement has been made (and its semantics determined for its own context) then the truth or falsehood of the statement is unaffected by reporting it in another context. Thus in the example above, a US newspaper reporting what had been said in the USA might include the statement that it had been reported in France that the (US) President had said there were

[1] "The President said that there are no weapons of mass destruction in Iraq."

[2] The reference to weapons of mass detruction was more problematic at the time because we did not know whether there were any in Iraq!

[3] There is, of course, a contradiction in what was said by the two men.

weapons of mass destruction in Iraq. We may roughly characterize by saying: in France, it is reported that in the US, ... The semantics here would only depend on what was said in the US, not what was reported in France (assuming that the media tell the truth).

Another idea relevant to the genesis of this paper comes from Barry Jay, who wrote on 14.x.2005 (email) in response to the second author's paper *Samsara* (Crossley, 2005):

> It provokes me to some thoughts. What might a distributed logic be? – one in which axioms or logical rules are distributed among a number of agents, and communication is limited?

This paper provides a partial response to those thoughts. However we shall actually distribute whole logics to various localities, as well as having a so-to-speak distributed logic.

2 TIERED FOL

First we consider the informal semantics. In our system we have a number of localities, think of France, the USA, etc., and each has its own local theory. these comprise Tier 0. For the models of our system, at each locality we have a natural, or perhaps better, traditional model of the local theory, that is to say, a model in the sense of first order logic. We collect these together to form a model for the global (tier 1) language. Now the underlying semantics at tier 1 is the standard semantics of propositional calculus except that here the traditional propositional letters are replaced by what we shall call basic global formulae (see below Definition 2).

However, we also have interaction between the global scene and the localities. Thus we have to specify how the semantics (the models) interact between tier 0 and tier 1. From an intuitive point of view the interaction is relatively simple and reflects our earlier informal example. This can be expressed intuitively as saying that a formula is interpreted in its local context, so that a tier 0 formula is interpreted in a traditional first order logic model (in tier 0 at a locality, l say). On the other hand a global, or tier 1, formula is interpreted using the values from the tier 0 model (or models) according to the usual rules for propositional calculus. When we go back down from tier 1 to tier 0, the semantic value is unchanged. (This depends on the fact that our formulae at tier 1 have no free variables and are therefore true or false.) The formal definitions follow the usual pattern. But first we must formally define our syntax.

2.1 Syntax

Because of going up and down between tiers the syntax looks a little complicated, however the actual formulae should be easily readable. We let $\mathbb{L}$ be a set of localities. At each locality $l \in \mathbb{L}$ we have a first order logic with a language $\mathcal{L}^l$ as usual. In order to describe our language(s) fully we have to take account of the fact that we go up and down between localities (in tier 0) and the global level (tier 1). Recall our example of reporting foreign language statements in a local medium. So we start off with strictly local languages which are standard first order languages for each locality. Then we describe the basic global formulae. These are combined as in an ordinary propositional calculus. But now we can take these back down to the local level, there they interact with formulae already there (including *strictly* local formulae) and the resulting set of formulae comprises the set of local formulae at that locality.

Definition 1 *A* strictly local formula *in the locality l is a (first order) formula of $\mathcal{L}^l$.*

So strictly local formulae look just like ordinary formulae of first order logic of $\mathcal{L}^l$.

Definition 2 *If φ is a strictly local formula in a locality l, then φ^l is a* basic global formula*.*

This definition shows how strictly local formulae can "move up" one tier to become global formulae. The next definition then constructs a standard propositional calculus from these (basic global) formulae.

Definition 3 (Global formulae)

1. *A basic global formula is a global formula.*
2. *If Φ and Ψ are global formulae, then so too are $\neg\Phi, (\Phi \wedge \Psi), (\varphi \vee \Psi)$ and $(\Phi \rightarrow \Psi)$.*
3. *If Φ is a global formula and l is* any *locality, then Φ^l is a global formula.*

However, once we have a formula in tier 1, it also becomes a formula of the tier below. Our definition of local formulae (Definition 4) is designed to reflect this. Thus the full range of local formulae in a locality l is generated by strictly local formulae of l and global formulae.

Definition 4 (Local formulae)

1. *If l is a locality and φ is a strictly local formula of locality l, then φ is a local formula of locality l.*
2. *If Φ is a global formula, then Φ is a local formula of locality l.*
3. *If A and B are local formulae, then so too are $\neg A, (A \wedge B), (A \vee B)$ and $(A \rightarrow B)$.*

$$\frac{}{\Gamma \vdash_x A}\ \text{(Axiom-I)} \qquad \frac{}{\Gamma, A \vdash_x A}\ \text{(Ass-I)}$$

where A is an axiom of Γ

$$\frac{\Gamma, A \vdash_x B}{\Gamma \vdash_x (A \to B)}\ (\to\text{-I})$$

$$\frac{\Gamma_1 \vdash_x A \quad \Gamma_2 \vdash_x (A \to B)}{\Gamma_1, \Gamma_2 \vdash_x B}\ (\to\text{-E})$$

$$\frac{\Gamma_1 \vdash_x A \quad \Gamma_2 \vdash_x B}{\Gamma_1, \Gamma_2 \vdash_x (A \wedge B)}\ (\wedge\text{-I})$$

$$\frac{\Gamma \vdash_x (A_1 \wedge A_2)}{\Gamma \vdash_x A_1}\ (\wedge\text{-E}_1) \qquad \frac{\Gamma \vdash_x (A_1 \wedge A_2)}{\Gamma \vdash_x A_2}\ (\wedge\text{-E}_2)$$

$$\frac{\Gamma \vdash_x A_1}{\Gamma \vdash_x (A_1 \vee A_2)}\ (\vee\text{-I}_1) \qquad \frac{\Gamma \vdash_x A_2}{\Gamma \vdash_x (A_1 \vee A_2)}\ (\vee\text{-I}_2)$$

$$\frac{\Gamma_1, \vdash_x A \vee B \quad \Gamma_2, A \vdash_x C \quad \Gamma_3, B \vdash_x C}{\Gamma_1, \Gamma_2 \Gamma_3 \vdash_x C}\ (\vee\text{-E})$$

$$\frac{\Gamma \vdash_x \bot}{\Gamma \vdash_x A}\ (\bot\text{-E}) \qquad \frac{\Gamma \vdash_x \neg\neg A}{\Gamma \vdash_x A}\ (\bot)$$

Figure 1: The basic propositional rules for global and local formulae. x indicates whether this is a local or a global derivation. For local derivations we set $x = l$, and Γ must be a set of formulae local to l and the formula to the right of $\vdash$ local to l. For global derivations we set $x = \gamma$, and Γ must be a set of global formulae and the fomula to the right of $\vdash$ a global formula. Likewise A, B, C are all global formulae or all formulae local to l as appropriate.

Thus local formulae and global formulae have been inductively defined using a pair of interacting inductive definitions. Notice that although global formulae are local formulae (for any locality) the reverse is definitely not the case. For example, a strictly local formula of locality l is not a global formula.

Here are examples of the various classes of formulae. We assume that the language of locality l has *only* the predicate letter P, and that the locality k has *only* the predicate letters P_1 and P_2.

- Strictly local formulae:
  $\forall x P(x)$ in the locality l; $(P_1(x) \to P_2(x))$ in the locality k; and $\exists y P_2(y)$ in the locality k.
- Global formulae:
  $\forall x P(x)^l$, $((\forall x P(x))^l \to (\exists x P_1(x))^k)$, $(\exists y P_2(y))^k$.
  1. Notice that the localities are superscripts in the global formulae;
  2. Each global example is either a superscripted local *sentence* or a propositional combination of such sentences.
- Local formulae for the locality k:
  $(\forall x P(x))^l$, $(P_1(x) \to P_2(x))$, and $\exists y((\forall x P(x))^l \to P_2(y))$.
  The first formula, $(\forall x P(x))^l$, is local (even in the locality k) because it is a global formula; the second is local in k because it is a strictly local formula of k; and the third is local in k, because it is a first order logic combination of a strictly local (and therefore also local) formula, $P_2(y)$, of k and a global (therefore also local) formula, $(\forall x P(x))^l$, (by clause 3 of Definition 4).

We have already introduced the notation $\mathcal{L}^l$ for the strictly local (tier 0) language of the locality l. We use the notation L^l for the local language of the locality l.

2.2 An axiom system

Our axiom system is designed from reflecting on the semantics. The (strictly) local syntax is simply first order logic in the language $\mathcal{L}^l$ for tier 0 and propositional calculus for tier 1. In addition to these we have certain rules for moving up and down between the tiers. We shall read $\Gamma \vdash_\gamma A$ as "Γ globally proves A"and $\Gamma \vdash_l A$ as "Γ proves A in the locality l". For convenience we give the (obvious) logical rules for local (including strictly local) and global formulae in Figure 1 and Figure 2.

In addition to these logical rules we have the transfer rules given in Figure 3. These rules are essentially due to (Buvač, 1993).

The (Exit) and (Enter) rules allow us to move up and down between the tiers, provided we appropriately tag or untag the formula. The rules (K), (D) and (T),[4] when used together with the (Exit) and (Enter) rules, ensure that the propositional connectives commute with moving between the tiers. Note that we write $\neg A^l$ for $\neg(A^l)$.

Remark 1 *If we did not have the* (Exit) *and* (Enter) *rules we would be able to have, say, the apparent inconsistency of having $\neg A$ at the local level and yet A^l at the global level.*

The idea of flatness was introduced in (Buvač, 1993). The rule (Flat) is used to ensure that once a statement has been made in one locality its truth-value is unchanged when it is subsequently taken into another locality. Consistency between local and global versions of a statement is ensured by (Flat-0), cf. Remark 1 above.

[4] We have retained Buvač's arrangement and labels.

$$\frac{\Gamma \vdash_l A}{\Gamma \vdash_l \forall x.A} \ (\forall\text{-I})$$

$$\frac{\Gamma \vdash_l \forall x.A}{\Gamma \vdash_l A[t/x]} \ (\forall\text{-E})$$
x is free in A, not free in Γ

$$\frac{\Gamma \vdash_l A[t'/y]}{\Gamma \vdash_l \exists y.A} \ (\exists\text{-I})$$

$$\frac{\Gamma_1 \vdash_l \exists y.A \quad \Gamma_2, A[x/y] \vdash_l C}{\Gamma_1, \Gamma_2 \vdash_l C} \ (\exists\text{-E})$$
where y is not free in C

Figure 2: The basic quantifier rules for local (resp. strictly local) formulae (i.e. tier 0). Γ is a set of local (resp. strictly local) sentences and the formula to the right of $\vdash$ is a local (resp. strictly local) formula. Likewise A, B, C are all local (resp. all strictly local) formulae local to l. x, y are individual variables, and t and t' are individual terms. $A[a/x]$ is read "A with a for x" and denotes the formula A with a substituted for the variable x.

$$\frac{\Gamma \vdash_l A}{\Gamma \vdash_\gamma A^l} \ (\text{Exit}) \qquad \frac{\Gamma \vdash_\gamma A^l}{\Gamma \vdash_l A} \ (\text{Enter})$$

$$\frac{\Gamma \vdash_\gamma (A \to B)^l}{\Gamma \vdash_\gamma (A^l \to B^l)} \ (\text{K})$$

$$\frac{}{\Gamma \vdash_\gamma (\neg A)^l \to \neg(A^l)} \ (\text{D}) \qquad \frac{}{\Gamma \vdash_\gamma \neg A^l \to (\neg A)^l} \ (\text{T})$$

$$\frac{}{\Gamma \vdash_\gamma (A^l)^k \leftrightarrow A^l} \ (\text{Flat}) \qquad \frac{}{\Gamma \vdash_l A \leftrightarrow A^l} \ (\text{Flat-0})$$

Figure 3: The transfer rules. Note that A and B must be local formulae of l for the (exit) and (Entry) rules. (Of course this includes global formulae.)

Together the rules give a natural deduction logical system in the usual way, cf. (Szabo, 1969) or our recent work (Poernomo et al., 2005). We illustrate this with a lemma and a theorem.

Lemma 1 *If Φ is a global formula, then $\Gamma \vdash_\gamma (\Phi \leftrightarrow \Phi^l)$ for any locality l.*

Proof. If Φ is of the form φ^k where φ is a strictly local formula of k, then we have $\Gamma \vdash_\gamma (\varphi^k \leftrightarrow (\varphi^k)^l)$ by (Flat).

If Φ is of the form $(\Psi \to \Theta)$, then by the induction hypothesis, $\Gamma \vdash_\gamma (\Psi \leftrightarrow \Psi^l)$ and $\Gamma \vdash_\gamma (\Theta \leftrightarrow \Theta^l)$. Hence $(\Psi \to \Theta)$ is globally provably equivalent to $(\Psi^l \to \Theta^l)$. But by (K), we then have that it is also globally provably equivalent to $(\Psi \to \Theta)^l$.

If Φ is of the form $\neg\Psi$, then by the induction hypothesis $\Gamma \vdash_\gamma (\Psi \leftrightarrow \Psi^l)$. Hence $\neg\Psi$ is globally provably equivalent to $\neg(\Psi^l)$. But by (D) and (T), it then follows that it is globally provably equivalent to $(\neg\Psi)^l$.

Since every propositional formula can be obtained using $\neg$ and $\to$, the result follows for all global formulae. □

Remark 2 *If Ξ is the strictly local theory in the locality l, then we define the* lifting *of Ξ to the global tier to be $\Xi^l = \{\varphi^l : \Xi \vdash_l \varphi\}$.*

Corollary 1 *$\Xi \vdash_l \varphi$ is equivalent to $\Xi^l \vdash_\gamma \varphi^l$.*

Lemma 2 *If Φ and all formulae in Γ are global formulae, and $\Gamma \vdash_l \Phi$, then $\Gamma \vdash_\gamma \Phi$.*

Proof. If Φ is of the form φ^l where φ is a strictly local formula, and $\Gamma \vdash_l \varphi^l$, then $\Gamma \vdash_\gamma (\varphi^l)^l$ by (Exit). Hence $\Gamma \vdash_\gamma \varphi^l$ by (Flat), i.e. $\Gamma \vdash_\gamma \Phi$.

If Φ is of the form $(\Psi \wedge \Theta)$ and $\Gamma \vdash_l (\Psi \wedge \Theta)$, then by propositional logic, $\Gamma \vdash_l \Psi$ and $\Gamma \vdash_l \Theta$. Hence by the induction hypothesis, $\Gamma \vdash_\gamma \Psi$ and $\Gamma \vdash_\gamma \Theta$. Hence $\Gamma \vdash_\gamma (\Psi \wedge \Theta)$, i.e. $\Gamma \vdash_\gamma (\Psi \wedge \Theta)$.

If Φ is of the form $\neg\Psi$, then by (Exit) $\Gamma \vdash_\gamma (\neg\Psi)^l$, and hence by Lemma 1, $\Gamma \vdash_\gamma \neg\Psi$. □

Lemma 3 (Weak normal form for global sentences) *Every global sentence is globally provably equivalent to a propositional combination of basic global sentences.*

Proof by induction on the construction of the global formula Φ.

If Φ is φ^l where φ is a strictly local sentence of l, the assertion is trivial.

Now write $pc(\Psi)$ for the equivalent propositional combination of basic global sentences. If Φ is of the form $(\neg\Psi)$, then by (D) and (T), $\vdash_\gamma (\Phi \leftrightarrow \neg pc(\Psi))$,and $\neg pc(\Psi))$ is provably equivalent to a propositional combination of basic global sentences.

If Φ is of the form $(\Psi \to \Theta)$, then by (K) and Lemma 1, $\vdash_\gamma (\Phi \leftrightarrow (pc(\Psi) \to pc(\Theta)))$, so Φ is provably equivalent to a propositional combination of basic global sentences.

Since all the other propositional connectives are definable from $\neg$ and $\to$ it remains to consider quantifiers and the tagging operation $(\ )^l$.

If Φ is of the form $\exists x(\Psi \to \Theta)$ where Ψ is global, then, by first order logic reasoning, Φ is globally provably equivalent to $(\Psi \to \exists x\Theta)$ since x does not occur free in Ψ because Ψ is a sentence. Similarly if Θ is global Φ is globally provably equivalent to $(\forall x\Psi \rightarrow \Theta)$. If neither Φ nor Θ is global, then Φ is a basic global sentence. Similarly for $\forall x(\Psi \rightarrow \Theta)$.

Finally, if Φ is of the form Ψ^l, then by Lemma 1, Φ is globally provably equivalent to Ψ, and therefore also to $pc(\Psi)$. □

Theorem 1 (CNF for global formulae) *Every global formula is provably equivalent to a conjunction of disjunctions of basic global formulae.*

Proof. By Lemma 3 every global formula is globally provably equivalent to a propositional combination of basic global formulae, therefore, by ordinary propositional calculus, every global formula can be put into conjunctive normal form $\Phi_1 \wedge \ldots \wedge \Phi_n$ where each Φ_i is a disjunction of basic global formulae.

By Theorem 1 it follows that in order to determine global satisfaction of a global formula we may first put the formula into conjunctive normal form. Then we determine the truth value of each basic global formula φ^l by determining the local truth value of φ in l. Finally we compute the global truth value from these truth values. (see also Section 2.7 below.)

2.3 Formal semantics

Remark 3 *The reader should be warned that these formal definitions look much more forbidding than they are in practice. He or she should refer back to our motivating section* 2*.*

We first define a strictly local model for a locality l as being a model in the usual first order logic sense, see, e.g. (van Dalen, 2004), and we denote such models as $\mathfrak{m}^l$. Then a model for the global system is a set of such models: $\mathfrak{M} = \{\mathfrak{m}^l : l \text{ is a locality}\}$.

Definition 5 (Strictly local satisfaction for sentences) *Given a structure $\mathfrak{m}_l$ for the language of a particular locality l, we define* strictly local satisfaction at l *(for strictly local sentences) $\mathfrak{m}_l \models^a_l \varphi(x_{i_1}, \ldots, x_{i_n})$ in the usual way. We also use the locutions 'φ is (strictly locally) true in $\mathfrak{m}_l$ (at l)', and '$\mathfrak{m}_l$ is a model of (the sentence) φ'.*

Definition 6 *A* global*, or* tier 1, structure *(for the second example) is a set $\mathfrak{M}$ of tier* 0 *models, $\{\mathfrak{m}_l : l \in \mathbb{L}\}$, where each $\mathfrak{m}_l$ is a structure for the language of locality l.*

In order to define global satisfaction we need simultaneously to define local satisfaction, so we have a double inductive definition.

Definition 7 (Global and part I of local satisfaction) *If φ^l is a* basic global sentence *(so φ is a strictly local* sentence *of locality l), then we say that φ^l is* (globally) true in $\mathfrak{M} = \{\mathfrak{m}_l : l \in \mathbb{L}\}$*, written $\mathfrak{M} \models_\gamma \varphi^l$, if, and only if, $\mathfrak{m}(l) \models_l \varphi$. In this case we also say φ^l is* locally satisfied at l*, and we write this as $\mathfrak{M} \models_l \varphi^l$.*[5]

If Φ is a global sentence*, then we simultaneously extend the definitions of $\mathfrak{M} \models_\gamma \Phi$ ($\mathfrak{M}$ globally satisfies Φ) and $\mathfrak{M} \models_l \Phi$ ($\mathfrak{M}$ locally satisfies Φ at l) inductively in the usual way for the standard propositional connectives.*

If Φ is of the form $\neg\Psi$, then $\mathfrak{M} \models_\gamma \Phi$ if, and only if, $\mathfrak{M} \not\models_\gamma \Psi$.

And for the local *satisfaction of* global *formulae we have:*

If Φ is of the form $\Psi \rightarrow \Theta$, then $\mathfrak{M} \models_l \Phi$ if, and only if, $\mathfrak{M} \not\models_l \Psi$ or $\mathfrak{M} \models_l \Theta$.

If Φ is of the form $\neg\Psi$, then $\mathfrak{M} \models_l \Phi$ if, and only if, $\mathfrak{M} \not\models_l \Psi$.

Finally, in this part of the definition, if Φ is of the form Ψ^l, then $\mathfrak{M} \models_\gamma \Phi$ and $\mathfrak{M} \models_l \Phi$ if, and only if, $\mathfrak{M} \models_l \Psi$.

Thus for global satisfaction we have, for example: If Φ is of the form $\Psi \wedge \Theta$, then $\mathfrak{M} \models_\gamma \Phi$ if, and only if, $\mathfrak{M} \models_\gamma \Psi$ and $\mathfrak{M} \models_\gamma \Theta$.

Remark 4 (Overlap requirements) *It is possible to have overlaps in the languages at the different localities. Suppose that c_i and the function letters used in the terms $t_1, t_2, t_3, \ldots, t_n$ and the predicate P are common to two (or more) localities, k, l. Then in this case we shall assume 1. on the syntactic front we have the additional (bridge rules)*
$\Gamma \vdash_k P(t_1, t_2, t_3, .., c, \ldots t_n)$ *if and only if*
$\Gamma \vdash_l P(t_1, t_2, t_3, .., c, \ldots t_n)$*,*
and 2. on the semantic front, for any model $\mathfrak{M} = \{\mathfrak{m}_j : j \in \mathbb{L}\}$ $\mathfrak{m}_k \models_k P(t_1, t_2, t_3, .., c, \ldots t_n)$ if and only if $\mathfrak{m}_l \models_l P(t_1, t_2, t_3, .., c, \ldots t_n)$.

It remains to define local satisfaction for local formulae that are not global formulae. Such formulae may contain free variables from a particular locality. If A is a local sentence of the form $\exists xB$ or $\forall xB$, then local satisfaction of the (local) formula A can be defined as for strictly local satisfaction except for those subformulae Ψ of B that are global formulae. Such Ψ contain no free variables.

Recall that all global formulae are also local to any locality. Therefore, although the preceding definition defines satisfaction for some local formulae, i.e. those built from global formulae and strictly local *sentences*, formally we still need to define local satisfaction for all the other local formulae.

It seems easiest to use Theorem 1 since then we only need to consider (equivalent) formulae in our CNF.

[5] There will be no ambiguity, because *strictly* local satisfaction is not defined for such formulae.

Definition 8 (Local satisfaction, part II) *Given a global structure $\mathfrak{M}$, we define* local satisfaction at l *(for local formulae), $\mathfrak{M} \models_l A$, by extending the usual definition to all formulae in CNF.*

If A is a global formula (and therefore a global sentence) Ψ, then $\mathfrak{M} \models_l A$ if, and only if, $\mathfrak{M} \models_\gamma \Psi$.

A sentence A is locally satisfied in l *if, and only if, $\mathfrak{m}_l \models_l A$. We also use the locutions "A is (strictly locally) true in $\mathfrak{m}_l$ (at l)", and "$\mathfrak{m}_l$ is a model of (the sentence) A".*

2.4 Consistency and soundness

There are many varieties of consistency: strictly local, global and local. Happily, because of our rule system they are all essentially equivalent (but note Theorem 2, part 2.

Definition 9 (Consistency)

1. *We say that a set of global formulae Γ is* globally consistent *if $\Gamma \nvdash_\gamma \bot$.*
2. *We say that a set, Γ_l, of formulae local in l is* locally consistent in l *if $\Gamma_l \nvdash_l \bot$.*
3. *We say that a set, Γ_l, of formulae strictly local in l is* strictly locally conistent in l *if $\Gamma_l \nvdash_l \bot$ where the rules are restricted to those of propositional calculus (i.e. the transfer rules are not used).*
4. *Given a set Γ of global formulae and sets Ξ of formulae local to l for each $l \in \mathbb{L}$, then we say (see Remark 1) that $\Gamma \cup \bigcup\{\Xi^l : l \in \mathbb{L}\}$ is* consistent *if $\Gamma \cup \bigcup\{\Xi^l_l : l \in \mathbb{L}\} \nvdash_\gamma \bot$ and $\Gamma \cup \bigcup\{\Xi^l : l \in \mathbb{L}\} \nvdash_l \bot$.*

Note that $\Gamma \vdash_\gamma \bot \leftrightarrow \bot^l$ by Lemma 1. Also, if Γ comprises only global formulae and $\Gamma \vdash_l \bot$, then $\Gamma \vdash_\gamma \bot$ by Lemma 1.

Theorem 2
1. *If Σ is a set of strictly local formulae then Σ is strictly locally consistent if, and only if, it is locally consistent.*
2. *If Σ is a locally consistent set of local formulae in a locality l, then $\Sigma^l = \{A^l : A \in \Sigma\}$ is globally consistent.*
3. *If Σ is a set of global formulae, then Σ is globally consistent if, and only if it is locally consistent at some locality l if, and only if, it is locally consistent for every locality.*

Proof. 1. Trivial.

2. Immediate from Lemma 1.

3. If $\Sigma \vdash_l \bot$, then $\Sigma \vdash_\gamma \bot^l$ by (Exit). Then by Lemma 1, $\Sigma \vdash_\gamma \bot$. The converse follows using the rule (Enter). □

We define *soundness* in the obvious way: A rule $\Gamma, A, B \vdash_x C$ is *sound* if whenever Γ, A and B are satisfied (globally, or locally at l) then so is C, respectively.

Theorem 3 *The axioms and rules in Figures 3, 1 and 2 are both globally sound, and locally sound for any locality l.*

Proof. It is easy to see that the standard rules of Figure 1 preserve truth. So we consider only the transfer rules and, for convenience, we shall further assume Γ is empty. For the rule (Exit), let $\mathfrak{M} \models_l A$. Then $\mathfrak{m}_l \models A$. Therefore by the penultimate clause of Definition 8, $\mathfrak{m}_l \models A^l$. Similarly for (Enter).

Suppose, for (K), that $\mathfrak{M} \models_\gamma (A \to B)^l$, then, by Definition 8, $\mathfrak{m}_l \models (A \to B)$. Suppose further that $\mathfrak{M} \models_\gamma A^l$, then $\mathfrak{m}_l \models A$. Hence by the semantic equivalent of rule ($\to$-E) of Figure 2, $\mathfrak{m}_l \models B$. Therefore $\mathfrak{M} \models_\gamma B^l$, and finally by the definition of satisfaction for $\to$,we have $\mathfrak{M} \models_\gamma (A^l \to B)^l$.

For (T) suppose $\mathfrak{M} \models_\gamma \neg A^l$. Then $\mathfrak{M} \not\models_\gamma A^l$ and therefore by Lemma 2, $\mathfrak{M} \not\models_l A^l$. Hence $\mathfrak{m}_l \not\models_l A^l$ and therefore, by definition 8, $\mathfrak{m}_l \not\models_l A$. Hence $\mathfrak{M} \not\models_l A$ and therefore $\mathfrak{M} \models_l \neg A$, since A is a sentence. Using the final clause of Definition 8 once more, we have $\mathfrak{M} \models_\gamma (\neg A)^l$.

The remaining rules are left to the reader. □

From the above we immediately have:

Theorem 4 (Consistency) *The rules of Figures 3, 1, and 2 are consistent.*

The following theorem shows how we can put sets of local formulae together consistently.

Theorem 5 *If Σ_k is a set of local formulae in k and Σ_l is a set of local formulae in l, then $\Sigma_k \cup \Sigma_l$ is globally consistent if, and only if, $\Sigma^k_k \cup \Sigma^l_l$ is globally consistent. Similarly for any number of localities (even infinitely many).*

Proof. Immediate from Theorem 2. □

2.5 Bridge rules

Bridge rules are global formulae involving local formulae from different localities. The original rules are given by (Ghidini and Serafini, 1998) and also used in description logics (Borgida and Serafini, 2002), see (Baader et al., 2003) for a basic reference for description logics.

In description logic, we have concepts, C and D, say. If we now want to put these in a tiered system like ours, they will belong to localities k and l, respectively, then, our version of the rules in (Borgida and Serafini, 2002) would mean we would write $C^k \sqsubseteq D^l$ which corresponds to the informal sentence $\forall x(C^k(x) \to D^l(x))$. However, we cannot model this directly in our system.[6] Nevertheless we can certainly

[6]For a treatment of our scheme in description logic see the first author's forthcoming thesis (Cruz, 2008).

imitate the intent of Borgida and Serafini by adding rules of the form:

For all constants c common to localities k and l

$$\frac{\Gamma \vdash_\gamma D^l(c)}{\Gamma \vdash_\gamma C^k(c)}$$

However, our system admits very powerful rules. For example, we can have rules that depend on not just one context influencing another, but more than one. We can have bridge axioms of the form

$$\varphi^k \wedge \psi^l \to \chi^m$$

or bridge rules of the form

$$\frac{\Gamma \vdash_\gamma \varphi^l \qquad \Gamma \vdash_\gamma \psi^k}{\Gamma \vdash_\gamma \chi^m}$$

or with even more premises. (The formulae, of course, may be any global formulae. We have simply used basic global formulae to make the interaction between localities clear.) Further examples of bridge rules involving quantification are: $\forall x P(x)^k \to \exists x Q(x)^l$, and $(\forall x P_1(x) \to \forall y P_2(y))^k \to ((\exists z Q_1(z) \to \forall w Q_2(w) \wedge \exists v Q_3(v))^l$.

2.6 Completeness

We now proceed to prove the completeness of our system under the tier scheme. We follow the technique of Leon Henkin (Henkin, 1949). Given Γ a set of consistent global formulae, we shall show that if a formula φ is consistent with a set of global formulae, Γ, then $\Gamma \cup \{\varphi\}$ has a model. This is equivalent, by the usual argument, see e.g. (Crossley et al., 1972), to showing that if a formula (global or local) is true in all models then it is provable.

Remark 5 *The restriction to global formulae is merely for convenience. If we had a set of local formulae Δ in a locality l then we could instead take the set of global formulae $\{\varphi^l : \varphi \in \Delta\}$ and use the rules* (Enter) *and* (Exit).

We follow the classical scheme

1. Add witnesses at each locality;
2. Extend Γ with Henkin sentences saying that existential statements are witnessed in all localities, giving us a larger set Γ^*;
3. Saturate Γ^* with formulae or or their negations thereby making it maximally consistent into a set called Γ^∞;
4. Construct from Γ^∞ the structures $\mathfrak{m}_l$ for each l;
5. Demonstrate that the resulting global structure $\mathfrak{M} = \{\mathfrak{m}_l : l \in \mathcal{L}\}$ is a model for Γ^∞; and then it follows that
6. $\mathfrak{M}$ is also a model for Γ since $\Gamma \subseteq \Gamma^\infty$.

Re: 1 and 2. First we add distinct constants $\{c^l_i : i = 1 \ldots \infty\}$ at locality l, then we add the *local* Henkin sentences, see e.g. (Crossley et al., 1972), $\exists x_n \varphi(x_n)_n \to \varphi(c^l_n)$ at that locality. By (Exit) we can infer the *global* sentence $(\exists x_n \varphi(x_n)_n \to \varphi(c^l_n))^l$. We add this in to Γ. We prove by a double induction on the number of the sentence and the localities that these additions are locally and globally consistent.

The proof is as usual except that we have to ensure consistency across localities. This is ensured by the model commonality requirement, see Remark 4 above .

Re:3. Now we enumerate all global sentences as $\Phi_i, i = 1, 2, 3, \ldots$. Amongst these global sentences there will occur global equivalents of all local sentences. For suppose φ_l is a local sentence in locality l, then φ^l is a global sentence.

In the usual way we set $\Gamma_0 = \Gamma^*$ and set

$$\Gamma_{n+1} = \begin{cases} \Gamma_n, & \text{if } \Gamma_n \vdash_\gamma \neg\Phi_{n+1} \\ \Gamma_n \cup \{\Phi_{n+1}\}, & \text{if } \Gamma_n \nvdash_\Gamma \neg\Phi_{n+1} \end{cases}$$

We show now that this Γ^l_{n+1} is globally and locally consistent in each locality, assuming that all previous Γ_n are. For the global case the result follows by propositional logic as usual. For the local case it follows by Theorem 3. Note that if there are bridge rules then the *strictly* local consistency at l does not guarantee *strict* local consistency at k.

In the usual way, see (Crossley et al., 1972), the set $\Gamma^\infty = \bigcup\{\Gamma_i : i = 1, 2, \ldots\}$ is maximally globally consistent. At this point it is convenient to introduce the useful notation (which is a sort of "opposite" of lifting in Remark 1):

$$(\Gamma^\infty)_l = \{\varphi : \varphi \text{ is a local sentence of } l \text{ and } \varphi^l \in \Gamma^\infty\}.$$

It is easy to see that $(\Gamma_\infty)_l$ is locally consistent. However, we also have to show it is *maximally* locally consistent. If it is not a maximal consistent set of local sentences in l, then there is a (local) formula ψ such that $(\Gamma^\infty)_l \cup \{\psi\}$ and $(\Gamma^\infty)_l \cup \{\neg\psi\}$ are both consistent. But then $\Gamma^\infty \vdash_\gamma \psi^l$ and $\Gamma^\infty \vdash_\gamma \neg\psi^l$ and by rules (D), (T) and (K), $\Gamma^\infty \vdash_\gamma (\psi \wedge \neg\psi)^l$, i.e. Γ^∞ is inconsistent.

In particular we have that for every global sentence Φ we have either $\Phi \in \Gamma^\infty$ or $\neg\Phi \in \Gamma^\infty$ but not both. We now use this to build our local and global models.

Re: 4. Consider the strictly local sentences in $(\Gamma^\infty)_l$. These include the atomic[7] (strictly) local sentences and it is just these that are used in the standard Henkin construction to build the model, see e.g. (Crossley et al., 1972). Call this model for the locality l, $\mathfrak{m}^l$, and let $\mathfrak{M} = \{\mathfrak{m}^l : l \text{ is a locality}\}$. The domain of $\mathfrak{m}^l$ is $\{c_i^l : i1, 2, \ldots\}$,

Lemma 4 $\mathfrak{M} \models_\gamma \Gamma^\infty$.

Sketch of proof. It is routine to show that Γ^∞ is closed under all the usual deduction rules. This follows as usual for the propositional connectives and the quantifiers. The same applies for the rules (D), (T) and (K). We now show that it is closed under the (Enter) and (Exit) rules. Suppose A is a local sentence of l *not* in $(\Gamma^\infty)_l$. Then we cannot have A^l in Γ^∞. hence $\neg(A^l)$ is in Γ^∞, and by rule (D) $(\neg A)^l$ is in in Γ^∞ and finally by (Enter), A is in $(\Gamma^\infty)_l$, which is a contradiction.

Finally, $\mathfrak{M}$ is a global model by the definition of its semantics. □

Theorem 6 (Completeness) *The system of rules in Figure 1 and Figure 2 are complete (both locally and globally), i.e. if, for every global model* $\mathfrak{M}$ *and every global sentence* Φ *we have* $\mathfrak{M} \models_\gamma \Phi \leftrightarrow \vdash_\gamma \Phi$, *and similarly for local sentences.*

2.7 Decidability

Although most first order logics are undecidable, nevertheless many are decidable. In this section we shall only be concerned with systems in which the first order logics in every locality we shall consider are decidable. We shall also assume that we are dealing with only a finite number of localities in our system.

Theorem 7 *If 1. the global system has only a finite number of localities and the strictly local theories at each locality are decidable, and 2. there is a finite number of bridge rules, them the global system is decidable.*

Proof. Recall that Ξ^l is the global version of the strictly local theory, Ξ, at locality l lifted to the global tier (see Remark 2 and Corollary 1). Let Δ be the set of bridge rules. Given a global formula Φ, then to say the system is decidable is equivalent to saying that there is an algorithm which decides whether $\Gamma \vdash_\gamma \bigwedge\{\Xi^l : l \text{ is a locality}\} \to \Phi$. Write Ψ for the sentence $\bigwedge\{\Xi^l : l \text{ is a locality}\} \to \Phi$, then by Lemma 3, Ψ can be expressed as a propositional combination of basic global sentences. By definition each basic global formula is a first order sentence ψ^l for some locality l. Construct a truth table for the sentence Ψ whose arguments form the set of these basic global sentences. Take those rows for which Ψ gets the value "True".

Now for each row ask a question for each basic global sentence ψ^l as follows: If ψ^l is assigned the value "True", is ψ locally provable in the locality l? and if ψ^l is assigned the value "False", is $\neg\psi$ locally provable in l? If, for some row, each of these gets the answer "Yes", then that row gets the truth value "True" and Ψ is globally provable. □

[7] Recall that the atomic formulae are those of the form $P(t_1, \ldots, t_n)$.

3 CASL

In the previous part of the paper there was no direct interaction between localities except in the presence of bridge rules, or overlapping languages (cf. Remark 4). There are other possibilities and papers such as Gabbay (Gabbay and Nossum, 1997) and (Nossum, 2003) have used superscripts to identify different structured contexts. Here we consider algebraic specifications, and new specifications built from old ones, as the localities. From an ontology point of view, there is strong reason to use CASL typed languages as an ontology languages, primarily because the operations provided by CASL flows over to the operations one might want to do to ontologies, e.g. translate one to another (**with**), combine them (**and**), hide some parts (**hide**), extend them (*Then*).

Each locality l will now be a specification described in a language such as CASL (Aspinall and Sannella, 2002) (Mosses, 1998). There is no necessity for these specifications to be finite but in practice we would expect them to be so.

3.1 Recapitulating CASL

CASL stands for "Common Algebraic Specification Language", see (Casl, 2001). It was designed by the Common Framework Initiative (CoFI) for algebraic specification and development. It is a tool for specifying the modular and functional requirements of software (Astesiano et al., 2002), and has first order logic as its base language and as such it may be used for for tier 0. A good overview of CASL from an applied logic standpoint may be found in (Poernomo et al., 2005) but we give a very brief review of CASL here. We note that the constructions we use are architectural specifications, this is to ensure the uniformity of constructions and to avoid clashes of notations.

3.2 CASL Basic specifications

CASL builds other specifications from *basic specifications*. A basic specification is an ordinary first order many-sorted logic of the form $S_P = < \Sigma, Ax >$, where $\Sigma =< S, TF, P >$ is the *signature* which comprises sorts, functions and predicates, Ax is a set of axiom formulae whose members come from the set of *well formed formulae of* SP (WFF(SP)). Models for CASL specifications are ordinary many-sorted models for first order logic. Such a model M, is a Σ-structure comprising non-empty carrier sets s^M for all $s \in S$, a function $f^M$from $w^M \rightarrow s^M$ for each $f \in TF_{w,s}$, a relation $P^M \subseteq s_1^M \times ... \times s_n^M$for each $P \in P_w$ with $w = s_1 ... s_n$ as the set of all $\Sigma - models$. We also denote the set of models SP by Mod(SP).

3.3 CASL Algebraic Operations/Expressions

CASL provides algebraic operations for building specifications. One starts with basic specifications and then uses the operations of translation, union, extension and hiding, which we briefly describe below. We use the *architectural specifications* of CASL so that we preserve the categorical structuring of the set of specifications. In practice this means that we have no problems of clashes of names.

When one views a CASL specification as a description of a theory i.e. a context or ontology (Lüttich and Mossakowski, 2004), then the immediate consequence is that we readily have ontology operations at our finger tips. The operations that may be performed on CASL specifications are defined by *specification expressions* in CASL literature.

Before discussing these operations, we need a few concepts and notations found in CASL.

1. **CSig** - the category with signatures as objects and signature morphisms as morphisms.

2. **CSpec** - the category of specification expressions, i.e. the range of possible basic and structured specifications.

   Structured specifications are ways of combining basic specifications. Fuller details of all our constructions may be found in the CASL Manual (Casl, 2001) or (Poernomo et al., 2005).

3. Associated with this are two maps: **sig** : **CSpec** $\rightarrow$ **CSig** and Mod : **CSpec** $\rightarrow \{\mathfrak{m} \in Struct(\Sigma) : \Sigma \in$ **CSig**$\}$ The former gives the signature of a specification and the latter gives the set of models of the specification.

3.3.1 Translation

Translation is simply the renaming of constants, predicates and functions in a specification. Formally a translation is the inductive closure of a symbol mapping ρ, which maps the symbols of SP to another specification, preserving sorts, etc..[8] This is written in CASL as SP **with** ρ.

3.3.2 Unions of Signatures and Specifications

In CASL the union of two specifications (possibly with some amalgamation) is achieved in such a way that the union specification is a conservative extension[9] of the two given specifications and, moreover, the models of the union are always such that they have reducts, see e.g. (Poernomo et al., 2005) or (Cengarle, 1994), that are models of the originally given specifications. Formally we proceed as follows.

Definition 10 *(Amalgamated Unions)*
Consider two signatures $\Sigma_1 =< S_1, TF_1, P_1 >$ $; \Sigma_2 =< S_2, TF_2, P_2 >$; *Assume these share a sub-signature (possibly null)* $\Sigma =< S, TF, P >$. *We define the* amalgamated union *of* $\Sigma_1 +_\Sigma \Sigma_2$ *to be the pushout of* ***CSig***. *This is diagrammed as*

$$\begin{array}{ccc} \Sigma & \xrightarrow{i_1} & \Sigma_1 \\ {\scriptstyle i_2}\downarrow & & \downarrow{\scriptstyle inl} \\ \Sigma_2 & \xrightarrow{inr} & \Sigma_1 +_\Sigma \Sigma_2 \end{array}$$

i_1, i_2 are injections from $\Sigma \rightarrow \Sigma_1$ and $\Sigma \rightarrow \Sigma_2$, respectively.

The union of two specifications, written SP_1 **and** SP_2 is defined as the union of the two specifications when the signatures have been amalgamated over a specificationSP as above. Thus we may define SP_1 **and** SP_2 by the diagram

$$\begin{array}{ccc} \text{SP} & \xrightarrow{i_1} & \text{SP\_1} \\ {\scriptstyle i_2}\downarrow & & \downarrow{\scriptstyle inl} \\ \text{SP\_2} & \xrightarrow{inr} & \text{SP\_1 } \textbf{and} \text{ SP\_2} \end{array}$$

[8]If symbols are in **sig** SP but not in the domain of ρ we make the convention that they are left unchanged. However, we also insist that this is done in such a way that there is no clash of names.

[9]I.e. no new sentences in the language of the first specification are provable from the theory of the union specification, and similarly for the second one

Given two specifications with possibly some overlap in their signatures we form a union of the two specifications amalgamating the common part.

3.3.3 Extension

Extensions are defined in a very similar way to unions except that we can extend by a partial specification. The extension of SP by SP_EXT is denoted as

$$\text{SP } \textbf{then} \text{ SP\_EXT}$$

For examples, see (Poernomo et al., 2005).

3.3.4 Hiding

Hiding mayperhaps be regarded as an opposite of taking extensions.

Given a SP and a symbol list *SL* the operation

$$\text{SP } \textbf{hide } SL$$

gives $\textbf{sig}\,(\text{SP } \textbf{hide } SL) = \textbf{sig}\,(\text{SP}/SL)$. The models of SP **hide** *SL* are

$$Mod(\text{SP } \textbf{hide } SL) = \{\mathfrak{m}|_\sigma : \mathfrak{m} \in Mod(\text{SP})\}$$

where σ is the injection from Σ to $\textbf{sig}\,(\text{SP})$, see (Poernomo et al., 2005).

4 THE TIERED CASL SYSTEM

The components of our system are algebraic specifications and from them we take our local languages. Overall the language of our system comprises labeled statements whose labels are the specifications in whose language they are written. We recall that a specification has a language inside it and this we designate as the "local language". From here, we then develop the global language for our system.

4.1 Tiered CASL Syntax

We follow the same model as before (see Section 2.1). In Tiered CASL, the *strictly local formulae* are simply first order formulae in the syntax of the locality SP. *Basic global formulae* are strictly local *sentences* annotated by superscripts that are specification [names]. Thus a strictly local sentence, φ, is lifted to the global level as a basic global sentence φ^{SP}. *Local formulae* in a specification (locality) SP are the inductive closure of the strictly local formulae and the global formulae.

However, we have one addition to these formation rules: we also use architectural specifications as localities.

Examples: We assume that the language of locality SP_1 has *only* the predicate letter P, that the locality SP_2 has *only* the predicate letters P_1 and P_2, and that locality SP_3 has only the predicate letter Q.

- Strictly local formulae:
  $\forall x : s \bullet P(x)$ in the locality SP_1 and $\forall x : s \bullet P(x)$ in the locality SP_1 **and** SP_2; $\forall x : s \bullet (P_1(x) \rightarrow P_2(x))$ in the locality SP_2; and $\exists y : s \bullet P_2(y)$ in the locality SP_1.
- Global formulae:
  $(\forall x : s \bullet P(x))^{\text{SP\_1}}$,
  $((\forall x : s \bullet P(x))^{\text{SP\_1 and SP\_2}} \rightarrow (\exists x : s \bullet P_1(x))^{\text{SP\_2}})$,
  $(\exists y : s \bullet P_2(y))^{\text{SP\_2}}$.
- Local formulae for the locality SP_2:
  $(\forall x : s \bullet P(x))^{\text{SP\_1}}$, $\forall x : s \bullet (P_1(x) \rightarrow P_2(x))$, $\exists y : s \bullet ((\forall x : s \bullet P(x))^{\text{SP\_1}} \rightarrow P_2(y))$, $(\forall x : s \bullet P(x))^{\text{SP\_1}} \rightarrow (\forall x : s \bullet Q(x))^{\text{SP\_3}}$ and $[(\forall x : s \bullet P(x))^{\text{SP\_1}}]^{\text{SP\_3}}$.

  The first formula, $(\forall x : s \bullet P(x))^{\text{SP\_1}}$ is local (even in the locality SP_2) because it is a global formula; the second is local in SP_2 because it is a strictly local formula of SP_2; and the third is local in SP_2, because it is a first order logic combination of a strictly local (and therefore also local) formula, $P_2(y)$, of SP_2, and a global (therefore also local) formula, $(\forall x : s \bullet P(x))^{\text{SP\_1}}$, (by clause 3 of Definition 4). The fourth is a mixture of global formulas from SP_1 and SP_3. The last one is a local formula for it is derived from a global formula (by clause 3 of Definition 3).
- Bridge rules: $(\forall x : s \bullet P(x))^{\text{SP\_1}} \rightarrow (\forall x : s \bullet Q(x))^{\text{SP\_3}}$, $(P(a))^{\text{SP\_1}} \rightarrow (Q(b))^{\text{SP\_3}}$, $(\exists x : s \bullet P(x))^{\text{SP\_1}} \leftrightarrow (\exists x : s \bullet Q(x))^{\text{SP\_3}}$

4.2 Tiered CASL Derivability

We define derivations as before using the same schemata, but we now add rules for the structured specifications. That is to say, the rules of the global system Tiered CASL are given as in Figures 1, 2, 3, and 4. However, we now write the (Enter) and (Exit) slightly differently to take account of the relevant specifications, see Figure 5.

Remark 6 *Note that, for convenience, we now sometimes write* $\Gamma, \text{SP} \vdash_\lambda$ *as an alternative to* $\Gamma \vdash_{\text{SP}}$.

Remark 7 Niceness. *We assume that all of the basic specifications,* SP*, in our system are consistent.*

Consistency, strictly local, global and local is defined exactly as above in Section 2.4.

$$\frac{\Gamma \vdash_\gamma A^{\text{SP}'}}{\rho(\Gamma) \vdash_\gamma \rho(A)^{\text{SP}' \textbf{ with } \rho}} \text{ (trans)}$$

If *SL* is any symbol list

$$\frac{\Gamma \vdash_\gamma A^{\text{SP}'}}{\Gamma \vdash_\gamma A^{\text{SP}' \textbf{ hide } SL}} \text{ (hide)}$$

provided **sig** $\{A \cup \text{SP}\}$ does not contain *SL*.

$$\frac{\Gamma \vdash_\gamma A^{\text{SP\_1}}}{\Gamma \vdash_\gamma inl(A)^{\text{SP\_1 \& SP\_2}}} \text{ (union}_1\text{)}$$

$$\frac{\Gamma \vdash_\gamma A^{\text{SP\_2}}}{\Gamma \vdash_\gamma inr(A)^{\text{SP\_1 \& SP\_2}}} \text{ (union}_2\text{)}$$

$$\frac{\Gamma \vdash_\gamma A^{\text{SP\_1}}}{\Gamma \vdash_\gamma inl(A)^{\text{SP\_1 } \textbf{then} \text{ SP\_EXT}}} \text{ (ext}_1\text{)}$$

$$\frac{\Gamma \vdash_\gamma A^{\text{SP\_EXT}}}{\Gamma \vdash_\gamma inr(A)^{\text{SP\_1 } \textbf{then} \text{ SP\_EXT}}} \text{ (ext}_2\text{)}$$

Figure 4: The structural rules involving specifications.

$$\frac{\Gamma \vdash_\gamma A^{\text{SP}}}{\Gamma, \text{SP} \vdash_\lambda A} \text{ (Enter)}$$

$$\frac{\Gamma, \text{SP} \vdash_\lambda A}{\Gamma \vdash_\gamma A^{\text{SP}}} \text{ (Exit)}$$

provided A is a local SP formula and Γ is a set of global formulae.

Figure 5: The context changing rules in Tiered CASL: going from global to local and *vice versa*.

5 Tiered CASL Semantics

Again we define the semantics of our system, strictly local, global and local, exactly as in Section 2.3, except that the models we are now considering are many-sorted. Global models $\mathfrak{M}$ will now be sets of models $\mathfrak{m}_{\text{SP}}$ such that SP is a specification in our system. However, because of the structural rules of Figure 4, such a global model $\mathfrak{M}$ must include models for all the specifications constructed from the basic specifications using translation, union, extensions and hiding.

We recall the consistency assumption on SP i.e. Niceness assumption Remark 7, applies to the context specification which we are referring to here. The categorical nature of the construction of the non-basic specifications guarantees that all of the specifications are consistent (provided the basic ones are!).

5.1 Tiered CASL Soundness

The soundness of Tiered CASL is proved as before except that we now have also to consider the structural rules.

Theorem 8 (Soundness Theorem) *Given a context specification* SP*, if*

$$\Gamma \vdash_\gamma \varphi^{\text{SP}} \quad \text{(or equivalently } \Gamma, \text{SP} \vdash_\lambda \varphi)$$

then $\Gamma \models_\gamma \varphi^{\text{SP}}$.

Proof. Since the other rules are treated in the usual way, we only need consider the structural rules.

$$\frac{\Gamma \vdash_\gamma A^{\text{SP\_1}}}{\Gamma \vdash_\gamma inl(A)^{\text{SP\_1 \& SP\_2}}} \text{ (union}_1\text{)}$$

Assume $\mathfrak{M} \models_\gamma A^{\text{SP\_1}}$, then the local model $\mathfrak{m}_{\text{SP\_1}}$ in $\mathfrak{M}$ is such that $\mathfrak{m}_{\text{SP\_1}} \models_\lambda A$. Let $\mathfrak{m}_{\text{SP\_2}}$ be any model of SP_2. Then the amalgamated union of $\mathfrak{m}_{\text{SP\_1}}$ and $\mathfrak{m}_{\text{SP\_2}}$ is a model of $inl(A)$. Since this is true for all such pairs of models we have $\Gamma \models_\gamma inl(A)^{\text{SP\_1 \& SP\_2}}$. The other cases are similar.

□

6 Tiered CASL Completeness

The initial idea of the completeness proof was inspired by that in Section 2.6. However, because changes in basic specifications cause changes in any structural specification constructed from them, we have to modify our strategy.

Corresponding to the steps in Section 2.6, and using the same numbering, we proceed as follows:

First recall that localities (i.e. specifications) may be built from other localities.

1. Add witnesses to each basic specification, SP to get a new basic specification SP+. This expands the specification at that locality in a trivial way. However, we then carry these over into constructed specifications, so that where before we had SP_1 and SP_2 we now have SP_1+ and SP_2+. So when we go to the locality SP_1+ **and** SP_2+ we also add new constants to obtain (SP_1+ **and** SP_2+)+. We do the

same for specifications using the other operations of Section 3.3: extension, hiding and translation.

2. Next extend *every* locality (specification) by Henkin sentences. Note that, for example, because of the structural rules (see Figure 4), (SP_1+ **and** SP_2+)+ will include Henkin sentences from SP_1+ and SP_2+ as well as its own Henkin sentences.

3. Now, given the set, Γ^*, of global formulae obtained after adding all the (liftings of) the Henkin sentences we saturate it as before to get a set Γ^∞ of global sentences. We then need to show that Γ^∞ is closed under *all* the rules, including the new structural ones. The cases for the basic sets of rules proceed as before. We give just one example for the structural rules.

(union$_1$) Assume that $A^{\text{SP\_1+}} \in \Gamma^\infty$. We now test if $inl(A)^{(\text{SP\_1+ \& SP\_2+})+} \in \Gamma_\infty$. Suppose not, then because Γ^∞ is maximal, we have $\neg(inl(A)^{(\text{SP\_1+ \& SP\_2+})+}) \in \Gamma_\infty$. Therefore $inl(\neg A)^{(\text{SP\_1+ \& SP\_2+})+} \in \Gamma_\infty$ since negation commutes with the locality (by rules (D), (T) and (K)) and with *inl* by the definition of *inl*. But then by (hide) $inl(\neg A)^{inl(\text{SP\_1+})} \in \Gamma_\infty$ and $(\neg A)^{\text{SP\_1+}} \in \Gamma_\infty$ by (trans) using the map $(inl)^{-1}$. Finally using (D) and (T) once more $\neg(A^{\text{SP\_1+}}) \in \Gamma_\infty$ which is a contradiction. □

The other cases are similar.

4. Now, for each specification SP+ we construct a local model $\mathfrak{m}_{\text{SP+}}$ following exactly the same technique as in Section 2.6.

5. Since SP+ only extends SP by individual constants, the reduct of $\mathfrak{m}_{\text{SP+}}$ obtained by ignoring the constants is a model of SP. Now let $\mathfrak{M} = \{\mathfrak{m}_{\text{SP+}} :$ for all localities SP$\}$.

As before we then have:

6. $\mathfrak{M} \models \Gamma^\infty$, whence $\mathfrak{M} \models \Gamma$.

So we have shown that if Γ is (globally) consistent then it has a (global) model and in the usual way we then have the desired result.

Theorem 9 (Completeness of Tiered CASL) *The system of rules in Figures 1, 2, 3, and 4 is complete (both locally and globally), i.e. if, for every global model $\mathfrak{M}$ and every global sentence Φ we have $\mathfrak{M} \models_\gamma \Phi \leftrightarrow \vdash_\gamma \Phi$, and similarly for local sentences for each specification.*

7 CONCLUSIONS AND FUTURE WORK

We have described a scheme that provides for global communication between agents in different localities, possibly with different logics, but certainly with different languages. in doing so we have allowed one locality to influence another by means of *bridge rules*. These build on earlier work in (Ghidini and Serafini, 1998) and (Borgida and Serafini, 2002). The new range of such rules is much more complex than in previous forays into this area since two (or even more) localities may affect what happens in another locality.[10]

We have proved completeness and consistency results for a basic system and also for a system, Tiered CASL, which allows the localities to be structured specifications in the Common Algebraic Specification Language, CASL.

For a practical implementation of our scheme we have built software where the local logic is PROLOG and the global logic is propositional calculus.

There remains one general area that particularly requires further investigation. we have found ourselves restricted to propositional logic at the global level. How do we do quantification there? Buvač (Buvač, 1996) developed quantification over contexts and we see no difficulty in extending our work in that direction. However we would like to imitate Borgida's $C^k \sqsubseteq D^l$ directly in first order logic. It does not seem to make sense to write $\forall x(C(x)^k \to D(x)^l)$ since some elements in locality k may not be in locality l. The very form of this sentence means that we have to have a universal structure including all the elements from all the localities, whereas one advantage of our scheme is that we can always add another locality with its own local theory – provided, of course, that we maintain consistency.[11] We could add existence predicates for each locality k but then this raises the question of what locality they are in.

So we remain like the ancient Chinese mathematician, Liu Hui, see p. 74 of (Li Yan and Du Shiran, 1987), "... not daring to guess, [we] wait for a capable person to solve it."

REFERENCES

Akman, V. and Surav, M. (1996). Steps toward formalizing context. *AI Magazine*, 17(3):55–72.

Aspinall, D. and Sannella, D. (2002). From specifications to code in casl. In *AMAST*, pages 1–14.

Astesiano, E., Bidoit, M., Kirchner, H., Krieg-Brückner, B., Mosses, P. D., Sannella, D., and Tarlecki, A. (2002).

[10] In the thesis of the first author (Cruz, 2008) the bridge rules based on (Ghidini and Serafini, 1998) and (Borgida and Serafini, 2002) have been directly simulated, but also strengthened in a description logic context.

[11] Bridge rules may cause conflicts and it is the responsibility of the user to avoid this, just as it is his or her responsibility when modifying any specification.

Casl: The common algebraic specification language. *Theoretical Computer Science*.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Borgida, A. and Serafini, L. (2002). Distributed description logics: Directed domain correspondences in federated information sources. In Meersman, R. and Tari, Z., editors, *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, volume 2519 of *LNCS*, pages 36–53. Springer Verlag, London.

Bouquet, P., Giunchiglia, F., Harmelen, H., Serafini, L., and Stuckenschmidt, S. (2004). Contextualizing ontologies. Technical Report DIT-04-013, University of Trento, Department of Information and Communication Technology, Povo - Trento(Italy).

Buvač, S. (1993). Propositional logic of context. In Fikes, R. and Lehnert, W., editors, *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 412–419, Menlo Park, CA. AAAI Press.

Buvač, S. (1996). Quantificational logic of context. In Shrobe, H. and Senator, T., editors, *Thirteenth AAAI National Conference on Artificial Intelligence*, volume 1, pages 600–606. AAAI Press/MIT Press.

Buvač, S., Buvač, V., and Mason, I. A. (1995). Metamathematics of contexts. *Fundamenta Informaticae*, 23(2/3/4):263–301.

Casl (2001). *CASL, The Common Algebraic Specification Language, Summary, 25 March 2001*. CoFI Language Design Task Group on Language Design.

Cengarle, M. V. (1994). *Formal Specifications with Higher-Order Parametrization*. PhD thesis, Ludwig-Maximilians-Universität, München.

Crossley, J. N. (2005). Samsara. In Gabbay, D., Goncharov, S. S., and Zakharyaschev, M., editors, *Mathematical Problems from Applied Logic. New Logics for XXI Century*, International Mathematical Series, pages 234–276. Springer, Berlin.

Crossley, J. N., Brickhill, C., Ash(†), C. J., Stillwell, J. C., and Williams, N. H. (1972). *What is mathematical logic?* Oxford University Press. Latest edition, Dover, 1990.

Cruz, R. P. (2008). *Tiered Logic with Applications to Contextualizing Logics*. PhD thesis, Monash University, Melbourne, Australia. In preparation.

van Dalen, D. (2004). *Logic and Structure*. Springer.

Gabbay, D. and Nossum, R. T. (1997). Structured contexts with fibred semantics. In Buvač, S. and Iwańska, Ł., editors, *Working Papers of the AAAI Fall Symposium on Context in Knowledge Representation and Natural Language*, pages 48–57, Menlo Park, California. American Association for Artificial Intelligence.

Ghidini, C. and Serafini, L. (1998). Distributed first order logics. In Baader, F. and Schultz, K. U., editors, *Frontiers of Combining Systems 2*, pages 121–139, Berlin. Research Studies Press/Wiley.

Giunchiglia, F. and Ghidini, C. (2000). Local models semantics, or contextual reasoing = locality + compatibility. *Artificial Intelligence*, 127(2):221–259.

Henkin, L. (1949). The completeness of the first-order functional calculus. *Journal of Symbolic Logic*, 14:159–166.

Li Yan and Du Shiran (1987). *Chinese Mathematics: A Concise History*. Oxford University Press, Oxford. Translated from the Chinese by J.N. Crossley and A.W.-C. Lun.

Lüttich, K. and Mossakowski, T. (2004). Specification of ontologies in CASL. In Varzi, A. C. and Vieu, L., editors, *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS-2004)*, pages 140–150, Amsterdam. IOS PRess.

Mosses, P. D. (1998). CASL: A guided tour of its design. In *Workshop on Algebraic Development Techniques*, pages 216–240.

Nossum, R. (2003). A decidable multi-modal logic of context. *J. Applied Logic*, 1(1-2):119–133.

Poernomo, I. H., Crossley, J. N., and Wirsing, M. (2005). *Adapting proofs-as-programs*. Springer, New York.

Serafini, L. and Bouquet, P. (2004). Comparing formal theories of context in ai. *Artificial Intelligence*, 155:41–67.

Szabo, M. E., editor (1969). *The collected papers of Gerhard Gentzen*. North-Holland Pub. Co., Amsterdam.