Tiered Logic for Agents in Contexts

Rosalito Perez Cruz and John Newsome Crossley

Faculty of Information Technology
Monash University, Australia, 3800
lcruz@sleekersoft.com, John.Crossley@infotech.monash.edu.au
http://www.csse.monash.edu.au/∼jnc, http://www.sleekersoft.com

Abstract. We introduce a new kind of logic for agents in different localities, 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.

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 take the locations as specifications. 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.

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

1 Introduction

It is well established 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 in the spirit of sharing. However, human nature ensures 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. So it seems more appropriate to conceive of pockets of communities sharing their ontologies and coping with any differences. It is more realistic 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:[1] a first order logic (FOL) of localities, **Tiered FOL**, which we use as a basis, then we extend this technique to a language **Tiered CASL**, where the localities are architectural specifications in the Common Algebraic Specification Language, CASL, see [6,2]. We prove completeness results for both these logics.

In the field of AI and, by association, Logic, there are two major styles of embedding localities[2] in a logical system. The first is in the Propositional Logic of Context (PLC)

[1] We use natural deduction systems throughout.

[2] We use “locality” rather than “context” because the latter is so ambiguous.

J. Filipe, A. Fred, and B. Sharp (Eds.): ICAART 2009, CCIS 67, pp. 191–204, 2010.

of Buvač-Mason [5] and their extension of this to FOL. The second is the Local Models Sematics/MultiContext Systems (LMS/MCS) of [11,12]. By no means do we imply that these are the only two possible styles: there are others such as in [1].

One example of an LMS approach in the field of Description Logic (DL) is that taken by Borgida and Serafin, who describe a Distributed DL in [3]. A major problem has been the transfer of knowledge between localities. Bridge rules (see Section 3) were introduced in [11], 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 locality. The idea is to align ontologies (or knowledge bases) 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, for example subsumption of concepts between ontologies. To do this, Borgida and Serafini extend the usual DL formulation, taking their cue from the Distributed First Order Logic (DFOL) of [11]. In their formulation, a DL statement is preceded by a label that stands for the ontology. Then they state *bridge rules*, which relate a concept in one ontology to another one in a different ontology (see [3]). Thus they have semantic mappings in the system.

Serafini, Borgida and Ghidini take their technique from Giunchiglia's LMS which they call the *compose-and-conquer* way of dealing with differences of languages in contexts. PLC uses a divide-and-conquer technique and since we take our cue from PLC, Tiered Logic is a divide-and-conquer technique, though the terminology may not be entirely appropriate as there are similarities to both.

Gabbay's Fibring of Logics, see e.g. [10], is clearly related to our work but does not permit the up and down interaction between local and global systems that we have.

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 by 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."[3]
> "The President said that there are weapons of mass destruction in Iraq."

Here the references 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.[4] There is no contradiction between the quotations, but there is between the two men.

In the media there would be an indication of the locality, i.e. country. 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 in that locality.

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

[4] The reference to weapons of mass destruction was more problematic because we did not know whether there were any in Iraq!

For agents in localities we again have the problem of them communicating across different languages. This paper provides a basic method of formalizing such situations by allowing the inclusion of powerful *bridge rules and axioms*.

We give the first presentation of what we call "Tiered Logic". In our logic, statements made in a local language are tagged with that local locality and then become "atomic" statements or basic propositions in a higher tier of what we call the *global* logic. With bridge rules or axioms any statement in one locality can have consequences in another. So information can be conveyed, or simply translated, from one locality to another.

We provide soundness and completeness proofs for two varieties of our underlying idea of tiered logic. For simplicity we assume that all our localities have the same underlying logic, but different languages. This restriction is not essential but a completely general approach would be notationally horrendous. The complications in our presentation come from the interactions between the tiers: when a sentence from one locality is used in a different locality, one has to refer back to the previous locality in order to determine the semantics.

Additionally we use Saša Buvač's notion of *flatness* (see e.g. [5] and Section 2). This entails that once a statement has been made (and its semantics determined for its own locality) then the truth or falsehood of the statement is unaffected by reporting it in another locality. 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 weapons of mass destruction in Iraq. 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).

2 Tiered FOL

First we consider the informal semantics. We have a number of localities, think of France, the USA, etc., each with its own local theory. In our first example we simply use first order logic at each locality. These comprise Tier 0. At each locality we have a traditional model of the local theory, that is to say, a first order model. We collect these together to form a model for the global (tier 1) language. The underlying semantics at tier 1 is the standard semantics of propositional calculus except that traditional propositional letters are replaced by what we call "basic" global formulae.

However, we also have interaction between the global scene and the localities. So 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. Intuitively: a formula is interpreted in its local locality, 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.

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

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

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

Fig. 1. The transfer rules for Tiered FOL. Note that A and B must be local formulae of l for the (Exit) and (Enter) rules. (Of course this includes global formulae).

localities. At each locality $l \in \mathbb{L}$ we have a first order logic with a language $\mathcal{L}^l$ as usual. These generate the *strictly local formulae*, which we denote by φ, ψ, etc. Going up to the global level (tier 1) we define the basic global formulae as strictly local sentences tagged by their locality, e.g. φ^l. These are combined as in an ordinary propositional calculus and we denote global formulae by Φ, Ψ, etc. But now we can take these back down to the local level, where they interact with formulae already there (including *strictly* local formulae). We then take the inductive closure in the usual way, to get the set of local formulae at that locality.

Thus local formulae and global formulae are 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.

Examples. 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: $\forall x P(x)^l$, $((\forall x P(x))^l \to (\exists x P_1(x))^k)$, $(\exists y P_2(y))^k$. Notice that the localities are superscripts in the global formulae. 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, $(\forall x P(x))^l$.

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 the rules in Figure 1 which are essentially due to Buvač [5]. We read $\Gamma \vdash_\gamma A$ as "Γ globally proves A"and $\Gamma \vdash_l A$ as "Γ proves A in the locality l".

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) and (DT), in which we follow Buvač, when used together with the (Exit) and (Enter) rules, ensure that the propositional connectives commute with moving between the tiers.[5] The rule (Flat),

[5] 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.

see [5], ensures that once a statement has been made in one locality its truth-value is unchanged when it is taken into another locality. (Flat-0), which is our addition to the ideas of Buvač, ensures consistency between local and global versions of a statement. cf. footnote 5 above.

Remark 1. If Ξ is the strictly local theory in the locality l, then we define the *lifting* of Ξ to the global tier to be $\Xi^l = \{\varphi^l : \Xi \vdash_l \varphi\}$.

Lemma 1. *1. If Φ is a global formula, then $\Gamma \vdash_\gamma (\Phi \leftrightarrow \Phi^l)$ for any locality l.*
2. $\Xi \vdash_l \varphi$ is equivalent to $\Xi^l \vdash_\gamma \varphi^l$.
3. If Φ and all formulae in Γ are global formulae, and $\Gamma \vdash_l \Phi$, then $\Gamma \vdash_\gamma \Phi$.

The proofs of these and all other results may be found in [9].

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

Proof. First show that every global formula is globally provably equivalent to a propositional combination of basic global formulae, and then, as usual, put this into conjunctive normal form. □

Formal Semantics. We first define a strictly local model for a locality l as a model in the usual first order logic sense, and we denote such models as $\mathfrak{m}_l$. These are the tier 0 models. Then a model for the *global* system, or *tier* 1 *model*, is a set of such models: $\mathfrak{M} = \{\mathfrak{m}_l : l \in \mathbb{L}\}$.

Remark 2 (Overlap requirements). It is possible to have overlaps in the languages at the different localities. Then we impose the requirement that if two atomic *sentences* from different localities, are syntactically identical, then they are semantically identical also. This will then carry over to more complicated formulae in the usual way.

In order to define global satisfaction we need simultaneously to define local satisfaction, so we have a double inductive definition. The reader should be warned that the formal definitions, which may be found in [9] look much more forbidding than they are in practice. He or she should refer back to the beginning of this section, and here we shall only give an intuitive picture.

Given a *basic global sentence* φ^l (which means φ is a strictly local *sentence* of locality l), then 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 $\mathfrak{M} \models_l \varphi^l$.[6]

If Φ is a *global sentence*, then we use the usual rules of propositional calculus to compute its truth value. This also covers local satisfaction.

This only partly defines global satisfaction, for it only defines it for propositional combinations of *basic* global sentences.

It remains to define local satisfaction for local formulae that are not global formulae. Such formulae may contain free variables from a particular locality. We simply do this in the obvious way, except that, because global formulae are sentences and have no free

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

variables, we can simply use the truth values of any global sentences contained in such a formula. Thus a local 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".

To determine global satisfaction of a global formula put the formula into conjunctive normal form by Theorem 1, then determine the truth value of each basic global *sub*-formula φ^l by determining the local truth value of φ in l. Finally compute the global truth value from these truth values.

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. For example, we say that a set of global formulae Γ is *globally consistent* if $\Gamma \nvdash_\gamma \bot$and that a set, Γ_l, of formulae local in l is *locally consistent in* l if $\Gamma_l \nvdash_l \bot$. It then follows that if Σ is a set of strictly local formulae then Σ is strictly locally consistent if, and only if, it is locally consistent; 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; and that 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.

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

Theorem 2. *1. The axioms and rules for Tiered FOL are both globally sound, and locally sound for any locality* l.

2. The rules and axioms for Tiered FOL are consistent. □

3 Bridge Axioms and Rules

Having discussed the syntax and semantics of our system, we turn to the concept of *lifting axioms* which was introduced by McCarthy and Buvǎc found in [16]. Lifting axioms relate truth in one context to truth in another. McCarthy and Buvǎc use this term to mean "inferring" or "lifting" truth in one context to truth in another context. An example of this is when two global formulas are related to each other, e.g.: $\exists x P(x)^k \to \exists x Q(x)^l$, but the connection may also be as complicated as: $(\forall x P_1(x) \to \forall y P_2(y))^k \to ((\exists z Q_1(z) \to (\exists w Q_2(w))^j \wedge \exists v Q_3(v))^l$.

Bouquet and Serafini are advocates of LMS (see Section 1), and in [19] they opt for the concept of *bridge rules*. Bridge rules say that an assertion in one context leads to a conclusion in another context. For example:

(*br*) From φ^i infer ψ^j.

They construct these bridge rules as inference rules that lie outside the theories that operate as the contexts. In LMS's presentation of its system and in [4] (see their Definition 7), it is depicted as: $MS = \langle\{C_i = \langle L_i, \Omega_i, \Delta_i\rangle\}, \Delta_{br}\rangle$, where C_i is a theory having L_i as a language with axioms in Ω_i and its inference rules are in Δ_i. Finally Δ_{br} is the set of bridge rules. As can be observed, Δ_{br} is not part of the theory and stands outside it.

In the present work, we prefer to use the terms *Bridge Axioms* and *Bridge Rules*, because they are "bridges" relating truth in one context to truth in another context. We are using this phrase to recall that Tiered Logic was specifically designed to support navigation among heterogenous knowledge bases, especially ontologies, which may be viewed as contexts or localities. They are axioms because they are part of the system: they do not lie outside it.

Our idea of bridging is a cross between worlds–PLC and LMS. Lifting axioms and bridge rules are easily seen to be special cases of our bridge axioms and rules, respectively. Both may be embedded, and not just simulated, in Tiered Logic. To see this, take for example PLC: then Γ may be seen as the trivial context i.e., the context where there are no strictly local formulas at all, only global formulas, i.e. only superscripted formulas. Take now LMS; then the bridge rules (br) can simply be added to the global system as new rules of inference. Indeed, looking at the composition of MS, we may see TLM as $\langle C_i, \Gamma\rangle$ except unlike MS, our Γ and C_i interact with one another and are integrated in the reasoning machinery. Of course, the issue of consistency arises. However, this issue is not unique to Tiered Logic alone, but it is common to any system, be it PLC or LMS, etc. Consistency checking is a standard process necessary in the use of any logic system so this is expected and should not surprise us.

When, as in LMS, bridge rules are outside the system, independently existing reasoning engines will usually have to be modified and expanded to accommodate these rules. With Tiered Logic, the global system employs a labelling mechanism and so minimal changes to existing reasoning engines are required for the Tiered Logic system to work.

When we turn to description logic, suppose we have concepts, C and D, in localities k and l, respectively, then, our version of the rules in [3] would mean we would write $C^k \sqsubseteq D^l$ which corresponds to the informal sentence $\forall x(C^k(x) \rightarrow D^l(x))$. However, we cannot model this directly in our system.[7] Nevertheless we can certainly 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 locality influencing another, but more than one. We can have bridge axioms of the form $\varphi^k \wedge \psi^l \rightarrow \chi^m$ 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. Further examples of bridge axioms involving quantification are: $\forall x P(x)^k \rightarrow \exists x Q(x)^l$, and
$(\forall x P_1(x) \rightarrow \forall y P_2(y))^k \rightarrow ((\exists z Q_1(z) \rightarrow \forall w Q_2(w) \wedge \exists v Q_3(v))^l$.

[7] For an implementation of our scheme using description logic see the first author's forthcoming thesis [8].

Completeness and Decidability. In order to prove the completeness of our system under the tier scheme, we follow the technique of Leon Henkin [13]. Given a set, Γ, of consistent global formulae, we extend this to a maximal consistent set, Γ^∞, and show this has a model.[8] The main difference from the classical scheme is that we make maximal consistent sets of sentences both at the global level, Γ^∞, and at each locality.[9]

Now consider the strictly local sentences in $(\Gamma^\infty)_l = \{\varphi : \varphi^l \in \Gamma^\infty\}$. These include the atomic (strictly) local sentences and it is just these that are used, in the standard Henkin way, to build a *local* model, $\mathfrak{m}_l$. Then we collect these into $\mathfrak{M} = \{\mathfrak{m}_l : l \in \mathbb{L}\}$ as a global model for Γ^∞.

The only unusual part is to show that each $(\Gamma^\infty)_l$ is maximal consistent. Consistency has been treated above, so suppose A is a local sentence of l and A is *not* in $(\Gamma^\infty)_l$. Then we cannot have A^l in Γ^∞ by (Enter) and (Exit). Hence $\neg(A^l)$ is in Γ^∞, and by rule (DT), $(\neg A)^l$ is in in Γ^∞. Finally by (Enter), $\neg A$ is in $(\Gamma^\infty)_l$. The rest of the Henkin style completeness proof follows as before.

Theorem 3 (Completeness). *The system of rules and axioms for Tiered FOL is complete (both locally and globally).*

For decidablility we restrict ourselves to systems in which the first order logics in every locality are decidable and there is only a finite number of localities in our system.

Theorem 4. *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 axioms and rules, them the global system is decidable.*

Proof. To decide whether

$$\Gamma \vdash_\gamma \bigwedge\{\Xi^l : l \text{ is a locality}\} \to \Phi$$

express the sentence as a propositional combination of basic global sentences.[10] Now use the truth values of these basic global *sentences* to compute the value of the sentence.

4 CASL

In the previous part of the paper there was no direct interaction between localities except in the presence of bridge axioms or rules, or overlapping languages (cf. Remark 2). There are other possibilities dealing with structured localities [10]. Here we consider algebraic specifications as the localities and build new specifications from old ones.

Each locality l will now be a specification described in a language such as CASL [6,2]. There is no necessity for these specifications to be finite but in practice we would expect them to be so.

[8] The restriction to global formulae is merely for convenience. Replace local formulae Δ in a locality l by the set of global formulae $\{\varphi^l : \varphi \in \Delta\}$ and use the rules (Enter) and (Exit).

[9] The proof is as usual except that we have to ensure consistency across localities. This is ensured by the model commonality requirement, see Remark 2 above.

[10] See Remark 1 for the definition of Ξ^l.

CASL stands for "Common Algebraic Specification Language", see [6,2]. 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, 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 [18] but we give a very brief review of CASL here. From an ontology point of view, there is a strong reason to use CASL-type languages as ontology languages, primarily because the operations provided by CASL flow over to the operations one might want to do to ontologies, e.g. translate one to another (**with** operation), combine them (**and** operation), hide some parts (**hide** operation), or extend them (**then** operation).

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 Σ 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. Models for CASL specifications are ordinary many-sorted models for first order logic. We denote the set of models of SP by $Mod(\text{SP})$.

CASL Algebraic Operations. 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 locality or ontology [15], then 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.

Structured specifications are ways of combining basic specifications. Fuller details of all our constructions may be found in the CASL Manual [6] or [18].

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..[11] This is written in CASL as SP **with** ρ.

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 of the two given specifications[12] and, moreover, the models of the union are always such that they have reducts that are models of the originally given specifications, see e.g. [18] or [7].

Formally we proceed as follows. The *amalgamated union* of two specifications, written SP_1 **and** SP_2 is defined as the pushout in the following diagram.

[11] If symbols are in 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.

[12] I.e. no new sentences in the language of either specification are provable from the theory of the union specification.

$$\begin{tikzcd}
\text{SP} \arrow[r, "i_1"] \arrow[d, "i_2"'] & \text{SP_1} \arrow[d, "inl"] \\
\text{SP_2} \arrow[r, "inr"] & \text{SP_1 } \textbf{and} \text{ SP_2}
\end{tikzcd}$$

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 SP **then** SP_EXT For examples, see [18].

Hiding may perhaps be regarded as an opposite of taking extensions. Given a SP and a symbol list SL, the operation SP **hide** SL cuts down the signature of SP to SP **hide** SL which is that of SP omitting the symbol list 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 signature of SP **hide** SL, see [18].

5 The Tiered CASL System

Syntax. We use architectural specifications as localities and we recall that a specification has a language inside it and this we designate as the "local language". We then follow the same model as before (see Section 2). 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.

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; $\forall x : s \bullet (P_1(x) \to P_2(x))$ in the locality SP_2; and $\exists y : s \bullet P_2(y)$ in the locality SP_1.

Global formulae: $(\forall x : s \bullet P(x))^{\text{SP\_1}}$, and
$((\forall x : s \bullet P(x))^{\text{SP\_1 } \textbf{and} \text{ SP\_2}} \to (\exists x : s \bullet P_1(x))^{\text{SP\_2}})$, $(\exists y : s \bullet P_2(y))^{\text{SP\_2}}$.

Local formulae for the locality SP_2:
$(\forall x : s \bullet P(x))^{\text{SP\_1}}$, $\forall x : s \bullet (P_1(x) \to P_2(x))$, $\exists y : s \bullet ((\forall x : s \bullet P(x))^{\text{SP\_1}} \to P_2(y))$, $(\forall x : s \bullet P(x))^{\text{SP\_1}} \to (\forall x : s \bullet Q(x))^{\text{SP\_3}}$ 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, $(\forall x : s \bullet P(x))^{\text{SP\_1}}$. 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.

Examples of bridge axioms: $(\forall x : s \bullet P(x))^{\text{SP\_1}} \to (\forall x : s \bullet Q(x))^{\text{SP\_3}}$, $(P(a))^{\text{SP\_1}} \to (Q(b))^{\text{SP\_3}}$, and $(\exists x : s \bullet P(x))^{\text{SP\_1}} \leftrightarrow (\exists x : s \bullet Q(x))^{\text{SP\_3}}$.

We define derivations as before using the same schemata, but add rules for structured specifications, viz., the rules of the global system Tiered CASL are given by first order logic at the local level and propositional calculus at the global level with the slightly modified transfer rules in Figure 2, and the structural rules in Figure 3.

Consistency, strictly local, global and local is defined exactly as above in Section 2, and as before we assume that all of the basic specifications, $\textsc{Sp}$, in our system are consistent.[13]

$$\frac{\Gamma \vdash_\gamma A^{\textsc{Sp}}}{\Gamma, \textsc{Sp} \vdash_\lambda A} \text{ (Enter)} \qquad \frac{\Gamma, \textsc{Sp} \vdash_\lambda A}{\Gamma \vdash_\gamma A^{\textsc{Sp}}} \text{ (Exit)}$$

provided A is a local $\textsc{Sp}$ formula and Γ is a set of global formulae.

Fig. 2. The transfer rules for Tiered CASL: going from global to local and *vice versa*

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

If SL is any symbol list

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

provided the signature of $\{A\} \cup \textsc{Sp}$ does not contain SL.

$$\frac{\Gamma \vdash_\gamma A^{\textsc{Sp\_1}}}{\Gamma \vdash_\gamma inl(A)^{\textsc{Sp\_1}\ \&\ \textsc{Sp\_2}}} \text{ (union}_1\text{)} \qquad \frac{\Gamma \vdash_\gamma A^{\textsc{Sp\_2}}}{\Gamma \vdash_\gamma inr(A)^{\textsc{Sp\_1}\ \&\ \textsc{Sp\_2}}} \text{ (union}_2\text{)}$$

$$\frac{\Gamma \vdash_\gamma A^{\textsc{Sp\_1}}}{\Gamma \vdash_\gamma inl(A)^{\textsc{Sp\_1}\ \textbf{then}\ \textsc{Sp\_ext}}} \text{ (ext}_1\text{)} \qquad \frac{\Gamma \vdash_\gamma A^{\textsc{Sp\_ext}}}{\Gamma \vdash_\gamma inr(A)^{\textsc{Sp\_1}\ \textbf{then}\ \textsc{Sp\_ext}}} \text{ (ext}_2\text{)}$$

Fig. 3. The structural rules involving specifications

Semantics. Again we define the semantics of our system, strictly local, global and local, exactly as in Section 2, except that the models we are now considering are many-sorted. Global models $\mathfrak{M}$ will now be sets of models $\mathfrak{m}_{\textsc{Sp}}$ such that $\textsc{Sp}$ is a specification in our system. However, because of the structural rules of Figure 3, such a global model $\mathfrak{M}$ must also include models for all the specifications constructed from the basic specifications using translation, union, extensions and hiding.

The soundness of Tiered CASL is proved as before, except that we also have to consider the structural rules and the specifications that can be constructed from the basic ones. We take (union$_1$) as an example.

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

[13] The categorical nature of the construction of the non-basic specifications guarantees that all of the specifications constructed are consistent (provided the basic ones are!).

The initial idea of the completeness proof was inspired by that in Section 3. However, because changes in basic specifications cause changes in any structural specification constructed from them, we have to modify our strategy.

First recall that localities (i.e. specifications) may be built from other localities, so when we 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, but it carrries over to constructed specifications, so that for SP_1 and SP_2 we now have extensions SP_1+ and SP_2+, to which we further add new constants to obtain (SP_1+ **and** SP_2+)+. Similarly for specifications using the other operations of Section 4: extension, hiding and translation.

In the procedure leading to the construction of the model, the cases for the basic sets of rules go 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 we have $\neg(inl(A)^{(\text{SP\_1+ \& SP\_2+})+}) \in \Gamma_{\infty}$ by maximality. Therefore $inl(\neg A)^{(\text{SP\_1+ \& SP\_2+})+} \in \Gamma_{\infty}$ since negation commutes with the locality, by rules (DT) and (K), and also commutes 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 (partial) inverse of inl. Finally using (DT) once more $\neg(A^{\text{SP\_1+}}) \in \Gamma_{\infty}$ which is a contradiction. □

Now, for each specification SP+ we construct a local model $\mathfrak{m}_{\text{SP+}}$ (which will automatically give a model for SP) as in Section 3, and then the global model is $\mathfrak{M} = \{\mathfrak{m}_{\text{SP+}} : \text{SP is a specification}\}$.

Theorem 5 (Completeness of Tiered CASL). *The system of rules and axioms for Tiered CASL 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.*

6 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 *bridge axioms* and *bridge rules*. The new range of bridging allows for much more complex interactions than those in e.g. [11] and [3], since two (or more) localities may affect what happens in another locality.[14] Further, the lifting axioms of [16] and the bridge rules of [11] (and cf. also [4]) are expressible within our systems and do not need to be extraneous as they were in that earlier work.

We have proved completeness and consistency results for a basic system, Tiered First Order Logic, and also for a system, Tiered CASL, which allows the localities to be structured specifications in 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.

[14] In the thesis of the first author [8] the bridge rules based on [11] and [3] have been directly simulated, but also strengthened in a description logic context.

There remains one general area that particularly requires further investigation. How do we do quantification at the global level? Quantification over localities was developed in [5], 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 , but it does not seem to make sense to write $\forall x(C(x)^k \rightarrow\ D(x)^l)$ since some elements in locality k may not be in locality l. So we remain like the ancient Chinese mathematician, Liú Huī (see p. 74 of [14]), "...not daring to guess, [we] wait for a capable person to solve it."

References

1. Akman, V., Surav, M.: Steps toward formalizing context. AI Magazine 17(3), 55–72 (1996)
2. Bidoit, M., Mosses, P.D.: CASL User Manual: Introduction to Using the Common Algebraic Specification Language. Springer, Heidelberg (2004)
3. Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. In: Spaccapietra, S., March, S., Aberer, K. (eds.) Journal on Data Semantics I. LNCS, vol. 2800, pp. 153–184. Springer, Heidelberg (2003)
4. Bouquet, P., Serafini, L.: On the Difference between Bridge Rules and Lifting Axioms. In: Blackburn, P., Ghidini, C., Turner, R.M., Giunchiglia, F. (eds.) CONTEXT 2003. LNCS, vol. 2680, pp. 80–93. Springer, Heidelberg (2003)
5. Buvač, S., Buvač, V., Mason, I.A.: Metamathematics of contexts. Fundamenta Informaticae 23(2/3/4), 263–301 (1995)
6. COFI: CASL, The Common Algebraic Specification Language, Summary. CoFI Language Design Task Group on Language Design (March 25, 2001), `http://www.brics.dk/Projects/CoFI/-Documents/-CASL/-Summary/` (2001)
7. Cengarle, M.V.: Formal Specifications with Higher-Order Parametrization. PhD thesis, Ludwig-Maximilians-Universität, München (1994)
8. Cruz, R.P.: Tiered Logic with Applications to Contextualizing Logics. PhD thesis, Monash University, Melbourne, Australia (in preparation)
9. Cruz, R.P., Crossley, J.N.: Tiered logic method for assisting agents. Technical Report Technical Report 2008/232. Monash University, Clayton School of Information Technology, Clayton, Victoria, Australia, `http://www.csse.monash.edu.au/publications/2008/tr-2008-232-abs.html`
10. Gabbay, D., Nossum, R.T.: Structured contexts with fibred semantics. In: Working Papers of the AAAI Fall Symposium on Context in Knowledge Representation and Natural Language, pp. 48–57. American Association for Artificial Intelligence, Menlo Park (1997)
11. Ghidini, C., Serafini, L.: Distributed first order logics. In: Baader, F., Schultz, K.U. (eds.) Frontiers of Combining Systems 2, pp. 121–139. Research Studies Press/Wiley, Berlin (1998)
12. Giunchiglia, F., Ghidini, C.: Local models semantics, or contextual reasoning = locality + compatibility. Artificial Intelligence 127(2), 221–259 (2000)
13. Henkin, L.: The completeness of the first-order functional calculus. Journal of Symbolic Logic 14, 159–166 (1949)
14. Yǎn, L., Shírán, D.: Chinese Mathematics: A Concise History. Oxford University Press, Oxford (1987); Translated from the Chinese by J.N. Crossley and A.W.-C. Lun

15. Lüttich, K., Mossakowski, T.: Specification of ontologies in CASL. In: Varzi, A.C., Vieu, L. (eds.) Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS 2004), pp. 140–150. IOS Press, Amsterdam (2004)
16. McCarthy, J., Buvač, S.: Formalizing Context (Expanded Notes). In: Buvač, S., Iwańska, Ł. (eds.) Working Papers of the AAAI Fall Symposium on Context in Knowledge Representation and Natural Language, pp. 99–135. American Association for Artificial Intelligence, Menlo Park (1997)
17. Nossum, R.T.: A decidable multi-modal logic of context. J. Applied Logic 1, 119–133 (2003)
18. Poernomo, I.H., Crossley, J.N., Wirsing, M.: Adapting proofs-as-programs. Springer, New York (2005)
19. Serafini, L., Bouquet, P.: Comparing formal theories of context in AI. Artificial Intelligence 155, 41–67 (2004/2005))