Towards a Modal Logic for the Global Calculus[*]

Marco Carbone Thomas T. Hildebrandt
Hugo A. López

IT University of Copenhagen, Copenhagen, Denmark
{carbonem,hilde,lopez}@itu.dk

Abstract

We explore logical reasoning for the global calculus, a coordination model based on the notion of choreography, with the aim to provide a methodology for the specification and verification of structured communications. Starting with an extension of Hennessy-Milner logic, we propose a proof system for the logic that allows for verification of properties among participants in a choreography. Additionally, some examples of properties on service specifications are drawn, and we provide hints on how this work can be extended towards a full verification framework.

1 Introduction

Due to the continuous growth of technologies, software development is recently shifting its focus on communication, giving rise to various research efforts for proposing new methodologies dealing with higher levels of complexity. A new software paradigm, known as *choreography*, has emerged with the intent to ease programming of communication-based protocols. Intuitively, a choreography is a description of the global flow of execution of a system where the software architect just describes which interactions *might* take place. This idea differs from the standard approach where the communication primitives are given for each single entity separately. A good illustration can be seen in the way a soccer match is planned: the coach has an overall view of the team, and organizes (a priori) how players will interact in each play (the role of a choreography); once in the field, each player performs his role by interacting with each of the members of his team by throwing/receiving passes. The way each player synchronize with other members of the team represents the role of an orchestration.

The work in [3] formalises the notion of choreography in terms of a calculus, dubbed the *global calculus*, which pinpoints the basic features of the choreography paradigm. Although choreography provides a good abstraction of the system being designed allowing to *forget* about common problems that can arise when programming communication e.g. races over a channel, it can still have complex structures hence being often error prone. Additionally, choreography can be non-flexible in early design stages where the architect might be interested in designing only parts of a system as well as specifying only parts of a protocol (e.g. initial and final interactions). In this view, we believe that a logical approach can allow for more modularity in designing systems e.g. providing partial specification of a system using the choreography paradigm.

In this document, we attempt to provide a link between choreography and logics. Starting with an extension of Hennessy-Milner logic [6], we provide the syntax and the semantics of a logic for the global calculus as well as several examples pointing out the usefulness of the method. Moreover, we provide a proof system that allows for property verification of choreographies and show that it is sound.

2 The Global Calculus

The Global Calculus (GC) [3, 4] allows for the description of choreographies as interactions between participants by means of message exchanges. The description of such interactions is centered around the notion of *session*, in which two interacting parties first establish a private connection k, often referred to as *session channel*, and then interact on k, possibly interleaved with other sessions. Each session k is unique, therefore communication activities will be clearly separated.

[*]Authors are listed alphabetically by last name.

2.1 Syntax.

The syntax of GC [3, 4] is given by the following grammar.

$$
\begin{array}{rlr|lr}
C ::= & A \rightarrow B : a(k).\, C & \text{(init)} & (\nu k)\, C & \text{(newL)} \\
\mid & A \rightarrow B : k\langle e, y\rangle .\, C & \text{(com)} & 0 & \text{(inaction)} \\
\mid & A \rightarrow B : k[l_i : C_i]_{i\in I} & \text{(choice)} & & \\
\mid & C_1 \mid C_2 & \text{(par)} & & \\
\mid & \textbf{if}\, e@A \textbf{ then } C_1 \textbf{ else } C_2 & \text{(cond)} & &
\end{array}
$$

Terms $C, C', \ldots$ are called *choreographies*; $A, B, \ldots$ range over *participants*; $k, k', \ldots$ are *session channels*; $a, b, c, \ldots$ *shared channels*; $v, w, \ldots$ variables; $X, Y, \ldots$ process variables; $l, l_i, \ldots$ labels for branching; and $e, e', \ldots$ denote arithmetic and other first-order expressions over variables and some values.

Above, (init) denotes a session initiation by A via B's service channel a, with fresh session channels k and continuation C. (com) denotes an in-session communication of e over a session channel k. Note that y does not bind in C. (choice) denotes a labelled choice over session channel k and set of labels I. $C_1 \mid C_2$ denotes the parallel product between C_1 and C_2. $(\nu k)\ C$ works the same as the name restriction operator in the π -calculus, binding k in C. Since such a hiding is only generated by session initiation, we assume that a hiding never occurs inside a prefix or a conditional. In the standard conditional operator (cond), $e@A$ indicates that e is located at participant A. 0 denotes termination. The free and bound session channels and term variables are defined in the usual way. The calculus is equipped with a standard structural congruence $\equiv$ [3].

2.2 Semantics.

We equip GC with a standard labelled transition semantics obtained by enriching the one in [3, 4]. Formally, actions in the semantics are defined using the notation $(\sigma, C) \xrightarrow{\ell} (\sigma', C')$ which says that a choreography C in a state σ (which maps participants to variable assignments) executes an action ℓ and evolves into C' with a new state σ' . Actions (or labels) are defined as $\ell = \{\mathsf{init}\ A \rightarrow B\ \mathsf{on}\ a(k),\ \mathsf{com}\ A \rightarrow B\ \mathsf{over}\ k,\ \mathsf{sel}\ A \rightarrow B\ \mathsf{over}\ k : l_i\}$, denoting initiation, in-session communication and branch selection. We write $C \longrightarrow C'$ when the states σ, σ' and ℓ are irrelevant, and $\longrightarrow^*$ for its transitive closure. A variable x located at A's is written as $x@A$. The same variable name labelled with different participant names denotes different variables (hence $\sigma@A(x)$ and $\sigma@B(x)$ may differ). The transition relation $\longrightarrow$ is defined as the minimum relation on pairs state/interaction satisfying the rules of Table 1.

$$
\text{(G-INIT)}\quad \frac{}{(\sigma,\, A \rightarrow B : a(k).\, C) \xrightarrow{\mathsf{init}\ A\rightarrow B\ \mathsf{on}\ a(k)} (\sigma, (\nu k)\, C)}
\qquad
\text{(G-COM)}\quad \frac{\sigma' = \sigma[x@B \mapsto v] \qquad \sigma \vdash e@A \Downarrow v}{(\sigma,\, A \rightarrow B : k\langle e, x\rangle .\, C) \xrightarrow{\mathsf{com}\ A\rightarrow B\ \mathsf{over}\ k} (\sigma',\, C)}
$$

$$
\text{(G-CHOICE)}\quad \frac{}{(\sigma,\, A \rightarrow B : k[l_i : C_i]_{i\in I}) \xrightarrow{\mathsf{sel}\ A\rightarrow B\ \mathsf{over}\ k:l_i} (\sigma,\, C_i)}
\qquad
\text{(G-IFT)}\quad \frac{\sigma \vdash e@A \Downarrow \mathtt{tt} \qquad (\sigma, C_1) \xrightarrow{\ell} (\sigma', C_1')}{(\sigma, \textbf{if}\, e@A \textbf{ then } C_1 \textbf{ else } C_2\,) \xrightarrow{\ell} (\sigma', C_1')}
$$

$$
\text{(G-STRUCT)}\quad \frac{C \equiv C'' \qquad (\sigma,\, C) \xrightarrow{\ell} (\sigma',\, C') \qquad C' \equiv C'''}{(\sigma,\, C'') \xrightarrow{\ell} (\sigma',\, C''')}
\qquad
\text{(G-IFF)}\quad \frac{\sigma \vdash e@A \Downarrow \mathtt{ff} \qquad (\sigma, C_2) \xrightarrow{\ell} (\sigma', C_2')}{(\sigma, \textbf{if}\, e@A \textbf{ then } C_1 \textbf{ else } C_2\,) \xrightarrow{\ell} (\sigma',\, C_2)}
$$

$$
\text{(G-RES)}\quad \frac{(\sigma,\, C) \xrightarrow{\ell} (\sigma',\, C') \qquad u \in \{a, k\}}{(\sigma, (\nu u)\, C) \xrightarrow{\ell} (\sigma',\, (\nu u)\, C')}
\qquad
\text{(G-PAR)}\quad \frac{(\sigma,\, C_1) \xrightarrow{\ell} (\sigma',\, C_1')}{(\sigma, C_1 \mid C_2) \xrightarrow{\ell} (\sigma',\, C_1' \mid C_2)}
$$

Table 1: Operational Semantics for the Global Calculus

In (G-Init), after A initiates a session with B on service channel a, A and B share k locally. This is denoted by the restriction of k over the continuation C. As for communication, in (G-Com), the expression e is evaluated into v in the A-portion of the state σ and then assigned to the variable x located at B resulting in the new state $\sigma[x@B \mapsto v]$. (G-Choice) chooses the evolution of a choreography resulting from a labelled choice over a session key k. (G-IfT) and (G-IfF) show the possible paths that a deterministic evolution of a choreography can produce. (G-Par), (G-Res) (G-Rec) and (G-Struct) behave as the standard rules for parallel product, restriction, recursion and structural congruence.

Remark 1 (Global Parallel)**.** Parallel composition in the global calculus differs from the notion of parallel found in standard concurrency models based on input/output primitives. In the latter, a term $P_1 \mid P_2$ may allow *interactions* between P_1 and P_2. However, in the global calculus, the parallel composition of two choreographies $C_1 \mid C_2$ concerns two parts of the described system where *interactions* may occur in C_1 and C_2 but never across the parallel operator $\mid$. This is because an interaction $A \to B \ldots$ abstracts from the actual end-point behaviour i.e. how A sends and B receives. In our model, dependencies between two choreographies can be expressed by using variables (in the state σ).

Example 1 (Online Booking)**.** We consider a simplified version of the online booking scenario presented in [9]. Here, the customer establishes a session with the airline company AC using service *ob* (online booking) and creating session keys k_1, k_2. Once sessions are established, the customer will request the company about a flight offer with his booking data, along the session key k_1. The airline company will process the customer request and will send a reply back with an offer using the session key k_2. The customer will eventually accept the offer, sending back an acknowledgment to the airline company using k_1. The following specification in the global calculus represents the protocol:

$$\begin{aligned} C_{OB} = \; & Cust \to AC : ob(k_1,k_2).\\ & Cust \to AC : k_1\langle booking, x\rangle.\\ & AC \to Cust : k_2\langle offer, y\rangle.\\ & Cust \to AC : k_1\langle accept, z\rangle.\ 0 \end{aligned} \tag{1}$$

2.3 Session Types for the Global Calculus.

We use a generalisation of session types [7] for global interactions, first presented in [4]. Session types in *GC* are used to structure sequence of message exchanges in a session.

A typing judgment has the form $\Gamma \vdash C : \Delta$, where Γ, Δ are *service type* and *session type* environments, respectively. Typically, Γ contains a set of type assignments of the form $a@A : (\vec{k})\alpha$, which says that a service a located at participant A may be invoked with a fresh $\vec{k}$ followed by a session α. Δ contains type assignments of the form $\vec{k}[A,B] : \alpha$ which says that a vector of session channels $\vec{k}$, all belonging to the same session between participants A and B, has the session type α when seen from the viewpoint of A. The typing rules are omitted, and we refer to [5] for the full account of the type discipline. Returning to the example presented in Equation 1, the service type of the airline company at channel *ob* can be described as:

$$ob@AC : (k_1,k_2).\ k_1 \downarrow booking(\texttt{string}).\ k_2 \uparrow offer(\texttt{int}).\ k_1 \downarrow accept(\texttt{int}).\ \texttt{end} \tag{2}$$

3 A Logic for the Global Calculus

In this section we provide the main ingredients of our logic for choreographies *GL* (Global Logic). In Section 3.1, we introduce the reader to the syntax of *GL* and give several examples. In Section 3.2, we present the semantics of *GL*, inspired by the modal logic presented in [1]. The logical language comprises assertions for equality, value/name passing and existential quantifiers, plus modalities for timed execution of actions.

3.1 Syntax and Examples.

Choreography assertions (ranged over by $\phi, \chi, \ldots$) give a logical interpretation of the global calculus introduced in the previous section. The grammar of assertions used in *GL* is defined as follows:

$$\phi,\chi ::= \langle\ell\rangle\phi \mid \exists t.\ \phi \mid \texttt{tt} \mid e_1 = e_2 \mid \phi \wedge \chi \mid \neg\phi \mid \circ\phi \mid \Diamond\phi \mid \phi \mid \chi$$

In $\exists t.\ \phi$, the variable t is meant to range over service and session channels, participants, labels and basic placeholders for expressions. Accordingly, it works as a binder in ϕ. In addition to the standard operators, we include an unspecified (decidable) equality on expressions ($e_1 = e_2$) as in [1]. Our operators depend on the labels of the labelled transition system of the global calculus: $\langle\ell\rangle\phi$ represents the execution of a labelled action ℓ followed by the assertion ϕ; $\circ\phi$ and $\Diamond\phi$ denote the standard next and evenutally operators respectively. The parallel operator in $\phi \mid \chi$ denotes composition of formulae: because of the unique nature of parallel composition in choreographies, we use the symbol $\mid$ in order to stress the fact that there is no interference between two choreographies running in parallel. As usual, we can get the full account of the logic by deriving the standard set of modal operators from the syntax presented above. For example, $\texttt{ff} = \neg\texttt{tt}$, $(e_1 \neq e_2) = \neg(e_1 = e_2)$, $\phi \vee \chi = \neg(\neg\phi \wedge \neg\chi)$, $\phi \Rightarrow \chi = \neg\phi \vee \chi$, $\forall x.\ \phi = \neg\exists x.\ \neg\phi$, $\Box\phi = \neg\Diamond\neg\phi$, $[\ell]\phi = \neg\langle\ell\rangle\neg\phi$.

Example 2 (Availability, Service Usage and Coupling)**.** *The logic above allows to express that, given a service invoker (known as A in this setting) requesting the service a, there exists another participant (called B in the example) providing a with A invoking it. This can be formulated in GL as*

$$\exists B.\ \langle \mathsf{init}\, A \to B\ \mathsf{on}\ a(k)\rangle\texttt{tt}$$

Assume now, that we want to ensure that services available are actually used. We can use the dual property for availability i.e. for a service provider B offering a, there exists someone invoking a:

$$\exists A.\ \langle \mathsf{init}\, A \to B\ \mathsf{on}\ a(k)\rangle\texttt{tt}$$

Verifying that there is a service pairing two different participants in a choreography can be done by existentially quantifying over the shared channels used in an initiation action. A formula in GL representing this can be seen below:

$$\exists a.\ \langle \mathsf{init}\, A \to B\ \mathsf{on}\ a(k)\rangle\texttt{tt}$$

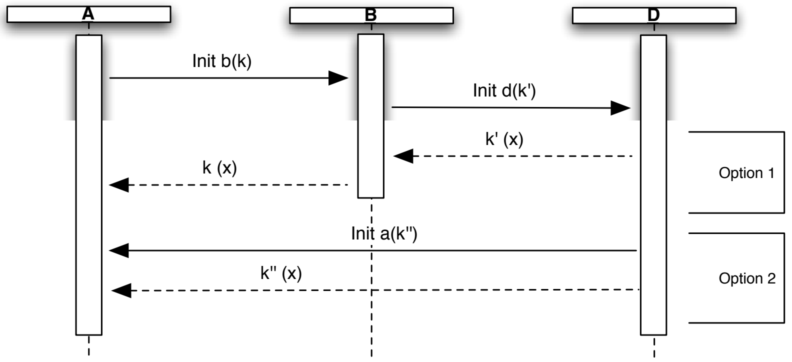
□

Example 3 (Causality Analysis)**.** *We can use the modal operators of the logic in order to perform studies of the causal properties that our specified choreography can fulfill. For instance, we can specify that given an expression e evaluated to true at participant A, there is an eventual firing of a choreography that satisfies property ϕ_1, and ϕ_2 will never be satisfied. Such a property can be specified as follows:*

$$(e@A = \texttt{tt}) \wedge \Diamond(\phi_1) \wedge \Box\neg\phi_2$$

□

Example 4 (Response Abstraction)**.** *An interesting aspect of our logic is that it allows for the declaration of partial specification properties regarding the interaction of the participants involved in a choreography. Take for instance the interaction diagram below:*



Above, participant A invokes service b at B's and then B invokes D's service d. At this point, D can send the content of variable x to A in two different ways: either by using those originally established sessions, or by invoking a new service at A's. However, at the end of the either computation path, variable z (located at A's) will contain the value of x. In the global calculus:

$$C \;=\; A \to B : b(k).\, B \to D : d(k').\ \mathbf{if}\, e@D\ \mathbf{then}\, C_1\ \mathbf{else}\, C_2 \quad \textit{where} \quad \begin{array}{ll} C_1 = & D \to B : k'\langle x, y_B\rangle.\, B \to A : k\langle y_B, z\rangle \\ C_2 = & D \to A : a(k'').\, D \to A : k''\langle x, z\rangle \end{array}$$

We argue that, under the point of view of A, both options are sufficiently good if, after an initial interaction with B is established, there is an eventual response that binds variable z. Such a property can be expressed in the logic by the formula:

$$\exists X, k''.\ \langle \mathsf{init}\, A \to B\ \mathsf{on}\ a(k)\rangle\ \Diamond\Big(\langle \mathsf{com}\, X \to A\ \mathsf{over}\ k''\rangle\ (z@A = x@D)\Big) \tag{3}$$

Note that a third option for the protocol above is to use delegation. However, the current version of the global calculus does not feature such an operation and we leave it as future work. □

Example 5 (Connectedness)**.** *The work in [4] proposes a set of criteria for guaranteeing a safe end-point projection between global and local specifications (note that the choreography in the previous example does not respect such properties). Essentially, a valid global specification have to fulfill three different criteria, namely Connectedness, Well-threadedness and Coherence. It is interesting to see that some of this criteria relate to global and local causality relations between the interactions in a choreography, and can be easily formalized as properties in the choreography logic here presented. Below, we consider the notion of connectedness and leave the other cases as future work. Connectedness dictates a global causality principle in interaction. If A initiates any action (say sending messages, assignment, etc) as a result of a previous event (e.g. message reception), then such a preceding event should have taken place at A. In the following, let* $\mathsf{Interact}(A,B)\phi$ *be a predicate which is true whenever* $\langle \ell \rangle \phi$ *holds for some* ℓ *with an interaction from A to B. Connectedness can then be specified as follows:*

$$\forall A, B.\ \Box\Big(\mathsf{Interact}(A,B)\mathtt{tt}\ \Rightarrow\ \exists C.\ \big(\mathsf{Interact}(A,B)\ \mathsf{Interact}(B,C)\ \mathtt{tt} \vee\ \mathsf{Interact}(A,B)\ \neg\langle \ell \rangle \mathtt{tt}\big)\Big)$$

□

3.2 Semantics of the Logic.

We now give a formal meaning to the assertions introduced above with respect to the semantics of the global calculus introduced in the previous section. In particular we introduce the notion of satisfaction.

$C \models_\sigma e_1 = e_2$	$\iff$	$\sigma(e_1) = \sigma(e_2)$	
$C \models_\sigma \mathtt{tt}$	$\iff$	tt	
$C \models_\sigma \phi \wedge \chi$	$\iff$	$C \models_\sigma \phi$ and $C \models_\sigma \chi$	
$C \models_\sigma \neg\phi$	$\iff$	$C \not\models_\sigma \phi$	
$C \models_\sigma \exists t.\ \phi$	$\iff$	$C[w/t] \models \phi$ (for some appropriate w)	
$C \models_\sigma \Diamond\phi$	$\iff$	$(\sigma, C) \longrightarrow^* (\sigma', C')$ and $C' \models_{\sigma'} \phi$	
$C \models_\sigma \langle\ell\rangle\phi$	$\iff$	$(\sigma, C) \xrightarrow{\ell} (\sigma', C')$ and $C' \models_{\sigma'} \phi$	$\ell \notin \{\mathsf{init}\ A \to B\ \mathsf{on}\ a(k)\}$
$C \models_\sigma \circ\phi$	$\iff$	$(\sigma, C) \longrightarrow (\sigma', C')$ and $C' \models_{\sigma'} \phi$	
$C \models_\sigma \langle\ell\rangle\phi$	$\iff$	$(\sigma, C) \xrightarrow{\ell} (\sigma', C')$ and $C' \models_{\sigma'} (\nu)\ k\phi$	$\ell = \mathsf{init}\ A \to B\ \mathsf{on}\ a(k)$
$C \models_\sigma (\nu k)\ .\ \phi$	$\iff$	$C \equiv (\nu k)C'$ and $C' \models_\sigma \phi$	
$C \models_\sigma \phi \mid \chi$	$\iff$	$C \equiv C_1 \mid C_2$ s.t. $C_1 \models_\sigma \phi$ and $C_2 \models_\sigma \chi$	

Table 2: Assertions of the Choreography Logic

We write $C \models_\sigma \phi$ whenever an environment σ and a choreography C satisfy a *GL* formula ϕ. The relation $\models_\sigma$ is the maximum relation satisfying the rules given in Table 2.

Above, we assume that variables occurring in an expression e are always located e.g. $x@A$. In the $\exists t.\ \phi$ case, w should be an appropriate value according to the type of t e.g. a participant if t is a participant placeholder. A formula ϕ is a "logical consequence" of a formula χ if every interpretation that makes ϕ true also makes χ true. In this case one says that χ is logically implied by ϕ ($\phi \Rightarrow \chi$). Moreover, let ϕ and χ be two formulae in the choreography logic. We say that ϕ is semantically equivalent to χ (denoted by $\phi \equiv_\models \chi$) if $\phi \models_\sigma \chi$ if and only if $\chi \models_\sigma \phi$. A formula is satisfiable if there is some choreography under which it is true. A formula with free variables is said to be satisfied by a choreography if the formula remains true regardless which participants, names/values are assigned to its free variables. A formula ϕ is *valid* if it is true in every choreography, that is $C \models_\sigma \phi$ for any C.

Proof System. Here, the proof system is presented. In order to reason about judgments $C \models_\sigma \phi$, we propose a proof (or inference) system for assertions of the form $C \vdash \phi$. Intuitively, we want $C \vdash_\sigma \phi$ to be as approximate as possible to $C \models_\sigma \phi$ (ideally, they should be equivalent). We write $C \vdash_\sigma \phi$ for the provability judgement where C is a process and ϕ contains a choreography formula.

We say that a choreography C *exhibits* a formula ϕ under an environment σ (written $C \vdash_\sigma \phi$) iff the assertion $C \vdash_\sigma \phi$ has a proof in the proof system given in Table 3.

Let us now describe some of the inference rules of the proof system. $\mathsf{P}_{\mathsf{inact}}$ and is the standard rule for inaction $\mathsf{P}_{\mathsf{sel}}$ can be explained as follows: suppose we are given a process $P = A \to B : k[l_i : C_i]_{i \in I}$, a set of branch labels $\{l_i\}$ (determined by typing) and we are given a proof that each C_i satisfies ϕ_i, then we certainly have a proof saying that every derivation of P should satisfy a guard l_i followed by a formula ϕ_i. The initiation and interaction rule $\mathsf{P}_{\mathsf{init}}, \mathsf{P}_{\mathsf{com}}$ behave similarly to $\mathsf{P}_{\mathsf{sel}}$: given an initiation/communication process in P and a proof that its continuation satisfies the proof term ϕ, we can derive a proof that P will first exhibit an initiation/communication action followed by ϕ. The rules for existential quantification, evaluation of expressions and conditional operators $\mathsf{P}_\exists$, $\mathsf{P}_{\mathsf{exp}}$, P_{if} are standard. The subsumption rule $\mathsf{P}_{\mathsf{sub}}$ is the standard consequence rule as found in Hoare logic. The rules for parallel composition and hiding are represented in $\mathsf{P}_{\mathsf{par}}$ and $\mathsf{P}_{\mathsf{res}}$ respectively, and they do not indicate the behaviour of a given choreography, but hint information about the structure of the process: $\mathsf{P}_{\mathsf{par}}$ juxtaposes the behaviour of two processes and combines their respective formulae by the use of a separation operator, $\mathsf{P}_{\mathsf{res}}$ hides a variable x in a formula ϕ; the intuition is that since $(\nu k)\ C$ is a choreography C with x restricted, then if

$$\mathsf{P}_{\mathsf{init}}\quad \frac{C \vdash_\sigma \phi}{A \to B : a(k).\, C \vdash_\sigma \langle \mathsf{init}\ A \to B\ \mathsf{on}\ a(k)\rangle \phi}$$

$$\mathsf{P}_{\mathsf{sel}}\quad \frac{\forall_{i\in I}\quad C_i \vdash_\sigma \phi_i}{A \to B : k[l_i : C_i]_{i\in I} \vdash_\sigma \bigwedge_{i\in I}\langle \mathsf{sel}\ A \to B\ \mathsf{over}\ l_i : s\rangle \phi_i}$$

$$\mathsf{P}_{\mathsf{com}}\quad \frac{C \vdash_\sigma \phi}{A \to B : k\langle e, y\rangle.\, C \vdash_\sigma \langle \mathsf{com}\ A \to B\ \mathsf{over}\ s\rangle \phi}$$

$$\mathsf{P}_{\mathsf{if}}\quad \frac{C_1 \vdash_\sigma e \Rightarrow \phi \qquad C_2 \vdash_\sigma \neg e \Rightarrow \phi}{\mathbf{if}\, e\ \mathbf{then}\, C_1\ \mathbf{else}\, C_2\ \vdash_\sigma \phi}$$

$$\mathsf{P}_{\mathsf{and}}\quad \frac{C \vdash_\sigma \phi \qquad C \vdash_\sigma \chi}{C \vdash_\sigma \phi \wedge \chi}$$

$$\mathsf{P}_{\mathsf{exp}}\quad \frac{\sigma(e_1) = \sigma(e_2)}{C \vdash_\sigma e_1 = e_2}$$

$$\mathsf{P}_{\mathsf{Inact}}\quad \frac{}{0 \vdash \mathtt{tt}}$$

$$\mathsf{P}_{\mathsf{res}}\quad \frac{C \vdash_\sigma \phi \qquad k \text{ is fresh}}{(\nu k)\, C \vdash_\sigma \nu k.\ \phi}$$

$$\mathsf{P}_{\mathsf{sub}}\quad \frac{C \vdash_\sigma \phi \qquad \phi \Rightarrow \chi}{C \vdash_\sigma \chi}$$

$$\mathsf{P}_{\mathsf{par}}\quad \frac{\forall_{i\in\{1,2\}}\quad C_i \vdash_\sigma \phi_i}{C_1 \mid C_2 \vdash_\sigma \phi_1 \mid \phi_2}$$

$$\mathsf{P}_{\exists}\quad \frac{C \vdash_\sigma \phi\{w \mapsto t\}}{C \vdash_\sigma \exists t \phi}$$

Table 3: Proof system for the Global Calculus

C proves ϕ and k is a fresh, then $(\nu k)\ C$ should satisfy ϕ with hidden x.

To guarantee the correctness of our logic, we shall prove the correspondence between the assertion semantics and the proof system. The details of the proof here presented can be found at [8].

Theorem 1 (soundness)**.** *for any given choreography C, if $C \vdash \phi$, then $C \models \phi$.*

4 Conclusion and Related Work

The ideas hereby presented constitutes just the first step towards a verification framework of structured communications. As a future work, our main concerns relate to establishing a completeness relation between the choreography logic and its proof system and the ability of integrating our framework into other end-point models and logical frameworks for the specification of sessions. In particular, our next step will focus on relating the logic to the end-point projection [4], the process of automatically generating end-point code from choreography. Other improvements to the system proposed include the use of fixed points, essential for describing state-changing loops, and auxiliary axioms describing structural properties of a choreography.

This work can be fruitfully nourished by related work in types and logics for session-based communication. In [9] the authors proposed a mapping between the calculus of structured communications and concurrent contraint programming, allowing them to establish a logical view of session-based communication and formulae in First-Order Temporal Logic. In [1], Berger et al. presented proof systems characterizing May/Must testing preorders and bisimilarities over typed π -calculus processes. The connection between types and logics in such system comes in handy to restrict the shape of the processes one might be interested, allowing us to consider such work as a suitable proof system for the calculus of end points. Finally, [10] studies a logic for choreographies in a model without services and sessions while [2] proposes notion of global assertion for enriching multiparty session types with simple formula describing changing in the state of a session.

Acknowlegments

This research has been partially supported by the Trustworthy Pervasive Healthcare Services (TrustCare) project. Danish Research Agency, Grant # 2106-07-0019 (www.TrustCare.eu).

References

[1] Martin Berger, Kohei Honda, and Nobuko Yoshida. Completeness and logical full abstraction in modal logics for typed mobile processes. In Luca Aceto, editor, *ICALP'08*, number 5126 in LNCS, pages 99–111. Springer-Verlag, Berlin Germany, 2008.

[2] Laura Bocchi, Kohei Honda, Emilio Tuosto, and Nobuko Yoshida. A theory of design-by-contract for distributed multiparty interactions. Available at `http://www.cs.le.ac.uk/people/lb148/AssertedTypes/assertedTypesExtended.pdf`, January 2010.

[3] M. Carbone, K. Honda, and N. Yoshida. A calculus of global interaction based on session types. In *2nd Workshop on Developments in Computational Models (DCM)*, ENTCS, 2006.

[4] M. Carbone, K. Honda, and N. Yoshida. Structured communication-centred programming for web services. In *(ESOP'2007)*, volume 4421 of *LNCS*, pages 2–17. Springer, Berlin Heidelberg, March 24–April 1 2007.

[5] M. Carbone, K. Honda, N. Yoshida, R. Milner, G. Brown, and S. Ross-Talbot. A Theoretical Basis of Communication-Centred Concurrent Programming. *Web Services Choreography Working Group mailing list, to appear as a WS-CDL working report*, 2009.

[6] M. Hennessy and R. Milner. On Observing Nondeterminism and Concurrency. In *Proceedings of the 7th Colloquium on Automata, Languages and Programming*, pages 299–309. Springer-Verlag London, UK, 1980.

[7] K. Honda, V.T. Vasconcelos, and M. Kubo. Language Primitives and Type Discipline for Structured Communication-Based Programming. In *ESOP'1998*, volume 1381 of *LNCS*, pages 122–138. Springer-Verlag London, UK, 1998.

[8] Hugo A. López. Formal models for trustworthy process and service oriented systems. Master's thesis, IT University of Copenhagen, Copenhagen, January 2009.

[9] Hugo A. López, Carlos Olarte, and Jorge A. Pérez. Towards a Unified Framework for Declarative Structured Communications. In *Programming Language Approaches to Concurrency and Communication-cEntric Software: PLACES'09*, February 2009.

[10] Carlo Montangero and Laura Semini. A logical view of choreography. In *COORDINATION*, pages 179–193, 2006.