Interface Theory for Relational Nets

Eda Baykan, Wojciech Galuba, Ali Salehi

Abstract

In this paper we show that a maximally expressive interface theory can be obtained for any component algebra by inverting the hierarchy relation. It is demonstrated that including the full I/O relation of the components in the interfaces might be necessary if the interface theory is to be able to express all the behaviors of relational nets. An instance of a minimally expressive interface theory for relational nets is also constructed.

1 Introduction

Interface models and component models are powerful ways of describing the structure and functionality of systems. Component model specifies how each part of the system behaves in any environment and the interface model specifies how each part of the system interacts with the environment.

Interface models and component models were formalized in [1]. Building upon that work we will focus on the problem of finding an interface theory for relational nets. We propose several types of interface theories for this component algebra. In section 2.1 we'll present an interface theory created by inverting the hierarchy relation relation of relational nets, and subsequently in section 2.2 show that such interface theory is also maximally expressive. We will motivate the need for such theory in section 3. Further on, in section 4 we will show a minimally expressive interface theory satisfying certain non-triviality constraints. Finally, in section 5, we will explore the space of interfaces that are neither maximally expressive nor minimally expressive.

2 Problem statement

Unless otherwise stated we will use the terminology introduced by [1]. The original problem was defined as follows:

P1 Find an interface theory for relational nets.

As it will be shown later there exist trivial solutions to P1. To rule out any trivial solutions the problem definition was modified as follows:

P2 Find an interface theory for relational nets that is as expressible for the subset of total nets as PD interfaces. Moreover, each component should have at least one interface.

2.1 Hierarchy inversion solves P1

In this section we will first formulate and prove a general theorem that allows to generate an interface theory for an arbitrary component algebra. Subsequently we will apply that theorem to solve P1.

Theorem 1 (Hierarchy inversion). *Given a block algebra* $A = (S, comp, conn, \preceq_A)$ *and* $B = (S, comp, conn, \preceq_B)$ *, where* S *is a set of blocks,* $comp$ *is a composition function,* $conn$ *is a connection function and* $\preceq_A$ *and* $\preceq_B$ *are hierarchy relations, if* $\forall_{x,y} x \preceq_A y \Rightarrow y \preceq_B x$ *then:*

a. If A *is an interface algebra then* B *is a component algebra and an implementation* $\lhd$ *defined between* A *and* B *as* $x \lhd y \Leftrightarrow x \preceq_A y$ *is compositional.*

b. If A *is a component algebra then* B *is an interface algebra and an implementation* $\lhd$ *defined between* A *and* B *as* $x \lhd y \Leftrightarrow x \preceq_B y$ *is compositional.*

Proof. We will show that claim a is true. We know that A is an interface algebra. It therefore has to satisfy two properties:

IA1. For all blocks F,G and F', if $F \parallel G$ is defined and $F' \preceq_A F$, then $F' \parallel G$ is defined and $F' \parallel G \preceq_A F \parallel G$

IA2. For all blocks F and F', and all interconnects θ, if $F\theta$ is defined and $F' \preceq_A F$, then $F'\theta$ is defined and $F'\theta \preceq_A F\theta$

Knowing that A and B have identical sets of blocks and the same connection and composition functions as A and using the fact that $\forall_{x,y} x \preceq_A y \Rightarrow y \preceq_B x$ we obtain:

1. For all blocks F,G and F', if $F \parallel G$ is defined and $F \preceq_B F'$, then $F' \parallel G$ is defined and $F \parallel G \preceq_B F' \parallel G$
2. For all blocks F and F', and all interconnects θ, if $F\theta$ is defined and $F \preceq_B F'$, then $F'\theta$ is defined and $F\theta \preceq_B F'\theta$

Therefore B is a component algebra.

We need to show that the implementation $\lhd$ defined between A and B as $x \lhd y \Leftrightarrow x \preceq_A y$ is compositional. For an implementation $\lhd$ of A by B to be compositional, it has to satisfy three conditions:

CI1. For all components f and g of B, and all interfaces F and G of A, if $f \lhd F$ and $g \lhd G$ and $F \parallel G$ is defined, then $f \parallel g$ is defined and $f \parallel g \lhd F \parallel G$.

CI2. For all components f of B, all interfaces F of A, and all interconnects θ, if $f \lhd F$ and $F\theta$ is defined, then $f\theta$ is defined and $f\theta \lhd F\theta$.

CI3. For all components f of B and all interfaces F' and F of A, if $f \lhd F'$ and $F' \preceq_A F$, then $f \lhd F$

CI1 and CI2 are trivially satisfied due to reflexivity of the implementation relation that we defined and to the fact the composition and connection functions are the same for A and B. Using the definition of our implementation relation $x \lhd y \Leftrightarrow x \preceq_A y$, we transform CI3 to: For all components f of B and all interfaces F' and F of A, if $f \preceq_A F'$ and $F' \preceq_A F$, then $f \preceq_A F$. This is always true because $\preceq_A$ is transitive.

The proof of the second part of theorem is analogous. $\square$

Hierarchy inversion theorem is universal and can be applied to any component algebra to produce a corresponding interface algebra for it along with a compositional implementation. In other words, for any component algebra we can generate an interface theory for it by inverting the hierarchy relation. From which it follows that such interface theory can be provided for relational nets in the same manner. This solves the problem P1.

2.2 Hierarchy inversion solves P2

To show that hierarchy inversion solves P2 we will need to prove that:

a. for every component there exists an interface

b. our new interface theory is as expressive for total nets as PD interfaces are.

Condition a is trivially satisfied because our implementation relation is defined based on a reflexive hierarchy relation. We will show condition b by first proving a more general theorem.

Theorem 1 (Maximal Expressiveness theorem). *Given a component algebra $B = (S, comp, conn, \preceq_B)$ and interface theory $(A_2, \triangleleft_2)$, where $A_2 = (S, comp, conn, \preceq_{A_2})$, $\forall_{x,y}\ x \preceq_B yR\rightarrow\ y \preceq_{A_2} x$ and $x \triangleleft_2 y \Leftrightarrow x \preceq_{A_2} y$, then $(A_2, \triangleleft_2)$ is as expressive for B as any other interface theory.*

Proof. The fact that $(A_2, \triangleleft_2)$ is a valid interface theory for B follows from the hierarchy inversion theorem. We need to show that $(A_2, \triangleleft_2)$ is as expressive as any other interface theory for B, $(A_1, \triangleleft_1)$. In other words we need to show a function α from the interfaces of A_1 to the interfaces of A_2 such that the following conditions hold:

EXP1 For all interfaces F, G of A_1, if $F \parallel G$ is defined, then $\alpha(F) \parallel \alpha(G)$ is defined and equal to $\alpha(F \parallel G)$.

EXP2 For all interfaces F of A_1, and all interconnects θ, if $F\theta$ is defined, then $\alpha(F)\theta$ is defined and equal to $\alpha(F\theta)$.

We assume the following algorithm for creating the function α: The value of α for F is not set until an assignment is performed : $\alpha(F) = x$.

1. For all pairs of interfaces $(F \in A_1, G \in A_1)$ and all interconnects θ
   - if $F \parallel G$ is defined and $\alpha(F \parallel G)$ is not set then:
     - if $\alpha(F)$ is not set then set $\alpha(F) := f$ so that $f \triangleleft F$
     - if $\alpha(G)$ is not set then set $\alpha(G) := g$ so that $g \triangleleft G$
     - $\alpha(F \parallel G) := \alpha(F) \parallel \alpha(G)$
   - if $F\theta$ is defined and $\alpha(F\theta)$ is not set then:
     - if $\alpha(F)$ is not set then set $\alpha(F) := f$ so that $f \triangleleft F$
     - $\alpha(F\theta) := \alpha(F)\theta$

Using structural induction we will show that the following invariants hold after execution of every step in the algorithm:

EXP1cond For all interfaces F, G of A_1, if $F \parallel G$ is defined and $\alpha(F)$ and $\alpha(G)$ and $\alpha(F \parallel G)$ are set, then $\alpha(F) \parallel \alpha(G)$ is defined and equal to $\alpha(F \parallel G)$.

EXP2cond For all interfaces F of A_1, and all interconnects θ, if $F\theta$ is defined and $\alpha(F)$ and $\alpha(G)$ and $\alpha(F\theta)$ are set , then $\alpha(F)\theta$ is defined and equal to $\alpha(F\theta)$.

Take EXP1cond, we know that A_1 is an interface theory therefore it must satisfy CI1. If $\alpha(F \parallel G)$ was set then it must have been set to $\alpha(F) \parallel \alpha(G)$ and we know that such composition is possible because CI1. Take EXP2cond, we know that A_1 is an interface theory therefore it must satisfy CI2. If $\alpha(F\theta)$ was set then it must have been set to $\alpha(F)\theta$ and we know that such interconnect is possible because CI2.

After the algorithm tertminates then all $\alpha(F)$ values that are referenced to in *EXP1* and *EXP2* were set. Then because *EXP1cond* and *EXP2cond* are invartiants, *EXP1* and *EXP2* are true.

□

The above theorem shows how to obtain a maximally expressive interface theory for any given component algebra. If we apply that theorem to relational nets we will have an interface theory that is as expressive as PD interfaces on total nets, which solves P2.

3 Why we might need the most expressive interface theory

By using the hierarchy inversion to generate an interface theory for relational nets we force the interfaces to be as complex as the components. More specifically the interfaces include the full I/O relation of the component. Is it necessary? We will demonstrate here that if it is required that the interface theory should be capable of expressing all relational nets then we need to include the full I/O relation in the interface.

Assume a Relational Net that is the composition of the following components: Component f with one input port x and one output port y. Components $g_0, g_1, ...$ each with one output port. Components $h_0, h_1, ...$ each with one input port. Assume that all port types are natural numbers. The predicates of components are set to $\rho_{g_i} = (output = i)$, $\rho_{h_i} = (input = i)$ and an arbitrary ρ_f. Assuming that $(A, \triangleleft)$ is an interface theory for relational nets. We require that every component should have an interface: $h_i \triangleleft H_i$ and $g_i \triangleleft G_i$ for all $i \in N$ and $f \triangleleft F$. We define $x = g_0 \parallel g_1 \parallel ... \parallel h_0 \parallel h_1 \parallel ... \parallel f$ and $x \triangleleft X$. Additionally we will also define $f' \neq f$ such that it has the same set of input and output ports and $\rho_{f'} \neq \rho_f$, $f' \triangleleft F'$, $x' = g_0 \parallel g_1 \parallel ... \parallel h_0 \parallel h_1 \parallel ... \parallel f'$ and $x' \triangleleft X'$.

Lemma 1. *Consider sets $\sigma(x) = \{\theta : x\theta$ is defined $\}$. If $f' \neq f$ then $\sigma(x') \neq \sigma(x)$.*

Proof. Consider a pair (k, l) that belongs to the relation R_f but does not belong to $R_{f'}$. Take an interconnect consisting of two pairs: { (output port of g_k, input port of f),(output port of f, input port of h_l) }. Then $\sigma(x)$ will contain that interconnect but $\sigma(x')$ will not. □

$$
\begin{array}{ccc}
g_0 & & h_0 \\
g_1 & & h_1 \\
\vdots & f & \vdots \\
g_k & (k,l) \in F_f & h_l \\
\vdots & & \vdots
\end{array}
$$

Figure 1:

From Lemma 1 it follows that if there is a difference in the I/O relation between the components then also the set σ of all possible interconnects will be different. In other words σ is determined by ρ. Any interface theory that would like to fully express all relational nets and their interconnects would have to include ρ as part of the interface.

4 The least expressive solution to P2

In the previous chapter we have shown a solution that has the maximum expressiveness, there exists no interface theory for relational nets that is more expressive. A question arises if there exists an interface theory with the minimum expressiveness, i.e. as expressive as PD interfaces for total nets but no more. We will try to construct an instance of such interface theory, however we will not prove that no less expressive theory exists.

Consider PD interface theory, we add a special interface E to the set of interfaces. We disallow compositions of E with any other interface including itself and we disallow any interconnects on E. E is in a hierarchy relation only with itself. We then take the implementation relation from PD interface theory and additionally define E to be the interface for all relational nets except total nets. Such augmented PD interface theory is an interface theory for relational nets.

CI1, CI2, CI3 follow trivially from the fact the PD interface theory is a valid interface theory and that no connections and compositions are allowed on E and that E is in hierarchy relation only with itself and no other interface.

The fact that the augmented PD interfaces are as expressive as PD interfaces can be shown by assuming an α function that maps all the interfaces from the original PD interface theory to the corresponding interfaces in the augmented PD interface theory.

5 Constructive Approach

So far we have shown the two extremes in the solution space for problem P2: minimally and maximally expressive. Useful interface theories, however, probably lie somewhere in between. It is known that Port Dependency Interfaces (PD-Interfaces) can compositionally implement total nets, which are a subset

$\psi : x = 1$ $\rho : v = \neg u$ $\phi : z = 1$

Figure 2: Counterexample

of relational nets. We also know that A/G Interfaces can compositionally implement rectangular nets, another subset of relational nets.

By trivial counterexamples it can be shown that neither PD interfaces nor A/G interfaces can compositionally implement all relational nets. It can also be shown that an interface algebra that is an intersection of PD interfaces and A/G interfaces also cannot implement relational nets.

In order to find a solution which doesn't include ρ in the interface itself (non-trivial solution) we considered combining the PD and A/G Interfaces, but in the *figure 2*, a counterexample is given which simply means the PD plus A/G don't sufficiently express the relational nets (The example clearly satisfies the requirements of both aforementioned interfaces but doesn't satisfy the requirements introduced by relational nets). In order to handle this, we considered about identifying an interface by introducing the set of all possible θs, and this set can easily be obtained from the relational nets, but we got the intuition that the set of θs approach can't solve the problem (The set of θs is explained in section 5). In fact the main problem for designing such an Interface is that, all above proposals are always restricting the next adjacent interfaces, and this is not enough for an interface that is compositionally implemented by relational nets. It's necessary to add some sort of global view for the interfaces. The intuition for this is because, adding an interconnection between two interfaces can possible break the requirements of relational nets at some other blocks (e.g. in above figure, any combination of two components can exist without any problem but the whole three couldn't exist together).

So we tried to add new constrains in the form of either A/G or PD interfaces to define a new interface. However the 3rd requirement of Compoistinal Implementation definition states that, if $f \triangleleft F'$ and $F' \preceq F$ then $f \triangleleft F$ and if we assume F' is the interface which has the new required constrains, and F is the interface without that constrains, then the system can use F instead of F' and this can lead to violation of relational nets. We need to add some constrains that are not in the form of A/G and PD interfaces and as we discussed earlier these new constrains should provide global view of the system. The idea of global view is already presented in the PD interfaces (The new connection shouldn't make loop) and this simply shows that it might be possible to find an interface theory for relational nets that doesn't contain full details of component. The new issue arises when we want to define what is exactly mean that we don't have ρ in the interface ? This issue is not addressed in the lectures but used explicitly for arguing about the solutions. We got the idea that sometimes the A/G interfaces could be as expressive as ρ itself which means sometime the predicates of A/G interfaces implies ρ. For example a component with one input port and one output port and the ρ that simply accepts anything from input and produces 1 in the output. An A/G interface for this component has output predicate $\psi : (output = 1)$. So there are some cases that interface could be as expressive as ρ itself.

6 Conclusion

We have shown how simple hierarchy inversion can generate an interface theory for any component algebra. Moreover, such interface theory is maximally expressive. It is shown that there exists a simple interface theory for relational nets that is also minimally expressive within the constraints of P2.Both of the theorems, maximal expressiveness and hierarchy inversion,are general enough to be used with arbitrary component algebras. However, the two solutions proposed for P2: the maximally and minimally expressive interface theories, are not adequate for practical design of relational nets. In order to find a meaningful interface theory for relational nets, the subset of the behaviors of relational nets that designer requires , must be explicitly defined. Only then a proper interface theory can be constructed.

References

[1] L. de Alfaro and T.A. Henzinger. *Interface theories for component-based design.* In *Embedded Software*, volume 2211 of Lect. Notes in Comp. Sci., pages 148-165. Springer, 2001.