

Interprétations de la composition d'activités

Soutenance d'habilitation à diriger des
recherches

Mireille Blay-Fornarino
Université de Nice-Sophia Antipolis, I3S
16 avril 2009

Evaluation Based Composition Of Activity Models


Soutenance d'habilitation à diriger des
recherches

Mireille Blay-Fornarino
Université de Nice-Sophia Antipolis, I3S
16 avril 2009

Background

- ❧ 1990 : Artificial Intelligence, Objects and Logic (I3S & INRIA)
- ❧ 1991 : Post-Doc : Model Library & Reasoning(CSTB)
- ❧ 1992 : I3S, Team working on Logic
- ❧ 1999 : Rainbow project creation with AM. Pinna-Dery
- ❧ 2006 : “Congé pour Recherches”
- ❧ 2008 : Participation to the creation of MODALIS project leaded by Johan Montagnat

Research Topics : Building Evolutive Complex Systems
Using Composition



**La Nature est un temple où de vivants piliers
Laissent parfois sortir de confuses paroles ;
L'homme y passe à travers des forêts de symboles
Qui l'observent avec des regards familiers.**

Baudelaire, Correspondances

Building Evolutive Complex Systems

A Point Of View :

Composition Of Software Activities

1- Software Activity Composition?

2- Language directed Composition

3- Meta-composition

4- Perspectives

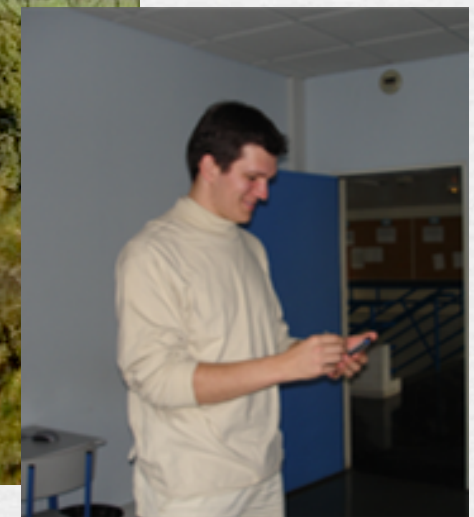
}

solutions

Software Activity Composition

What does it mean?
What are the issues?
(context)

Illustrating Activity Composition With Examples





Activity Compositions : Software Interactions From Client-side

Multi-user

Automatic composition

Composition at runtime

Component
Connector
Event Prog.
ADL
Aspects
Interactions

- ☒ Order-independent composition
- ☒ Associativity
- ☒ Condition preservation



Activity Compositions : Service Composition

 *Evolution management*
 *Static composition*

Mme Péresse : absente

12h00	2 séances + d'infos ...	4 séances + d'infos ...	2 séances + d'infos ...	2 séances + d'infos ...
14h00	2 séances + d'infos ...	4 séances + d'infos ...	2 séances + d'infos ...	2 séances + d'infos ...
15h00	2 séances + d'infos ...	4 séances + d'infos ...	2 séances + d'infos ...	2 séances + d'infos ...
16h00	2 séances + d'infos ...	4 séances + d'infos ...	2 séances + d'infos ...	2 séances + d'infos ...
19h00	2 séances + d'infos ...	4 séances + d'infos ...	2 séances + d'infos ...	2 séances + d'infos ...
20h00	2 séances + d'infos ...	4 séances + d'infos ...	2 séances + d'infos ...	2 séances + d'infos ...
21h00	2 séances + d'infos ...	4 séances + d'infos ...	2 séances + d'infos ...	2 séances + d'infos ...

Univ-Nice: Ronde des Facs 2009

Judi 26 mars 2009 Modalités de pré-inscription et d'inscription pour tous les participants (cours, marcheurs, coureurs et marcheurs, bénévoles pour l'organisation) Pré inscription gratuite par internet. Un courriel "Accusé de Réception" sera envoyé sur votre messagerie. A...

26 Mar 2009 - 11:55 am ...

CIP: Journée portes ouvertes

Si vous souhaitez nous rencontrer, rendez-vous le samedi 7 mars 2009 dans nos locaux (PolytechNice-Sophia, site des Lucioles) pour une journée portes ouvertes.

7 Mar 2009 - 10:00 am ...

Alumni: 11ème édition du Sophia Forum

Le 29 janvier, aura lieu la 11ème édition du Sophia Forum aux Espaces Antipolis, forum de l'emploi et des métiers, organisé par l'association Sophia Forum de l'école. Pour ceux qui ne connaissent pas encore, c'est un forum destiné aux étudiants et à la recherche de l'emploi ou d'un stage. Vous y trouverez des secteurs d'activité tels que le management, le marketing, les finances, etc.

Sciences Informatiques: Java EE mini-concours

Lunatch Research and the IUT de Nice Sophia-Antipolis organise un mini-concours de programmation en Java EE on Wednesday, 14 January 2009 at the IUT de Nice Sophia-Antipolis. as well as numerous details, are available on the website.

Service, Mashup
BPMN
Orchestrations
Algebra

☒ Consistency
☒ Quality of service

Personal Information

First Name	
Second Name	
Email	
Email (again)	
Affiliation	
Address (line 1)	
Address (line 2)	
City (+CEDEX, for French addresses)	
ZIP/Postal code	
Country	
Phone	
Fax	

Registration options

Registration



Activity Compositions :

Weaving Technical Functionalities

 *Separation of concerns*
 *Conflict detection*

Comp., Service,
Orchestration
Scenario
State,
Aspects

 “Functionality” preservation
 Order & condition preservation



Activity Compositions : Activity “Merging”

- Evolution management
- Semi-automatic composition (*Humatisation*)
- Conflict Detection



- ☒ Mutualization
- ☒ Quality of service
- ☒ Activity preservation
- ☒ Consistency

Univ-Nice: Ronde des Facs 2009

Jeudi 26 mars 2009 Modalités de pré-inscription et d'inscription pour tous les participants (coursiers, marcheurs, coursiers et marcheurs, bénévoles pour l'organisation) Pré inscription gratuite par internet. Un courriel "Accusé de Réception" sera envoyé sur votre messagerie. A...

26 Mar 2009 - 11:55 am ...

CIP: Journée portes ouvertes

Si vous souhaitez nous rencontrer, rendez-vous le samedi 7 mars 2009 dans nos locaux (Polytech-Nice-Sophia, site des Lucioles) pour une journée portes ouvertes.

7 Mar 2009 - 10:00 am ...

Alumni: 11ème édition du Sophia Forum

Le 29 janvier, aura lieu la 11ème édition du Sophia Forum aux Espaces Antipolis, forum de l'emploi et des métiers, organisé par l'association Sophia Forum de l'école. Pour ceux qui ne connaîtraient pas encore, c'est un forum destiné aux étudiants et jeunes diplômés (bac+2 à bac+8), à la recherche d'un emploi ou d'un stage. Vous y trouverez des secteurs aussi variés que l'électronique, l'informatique, les telecom, le management, le marketing, les finances etc... Pour plus (...)

29 Jan 2009 - 3:29 pm ...

Sciences Informatiques: Java EE mini-conference

Lunatech Research and the IUT de Nice Sophia-Antipolis have organised a free mini-conference on Java EE on Wednesday, 14 January 2009 at the IUT de Sophia-Antipolis (Google Maps). The program, as well as numerous details, are available on the event's official web page. We are expecting numerous

Orchestration Ontologies, Algebra,...





Requirements & Properties Summary

Humatisation

Evolution management

Semi-automatic composition

*Multi-user/
Separation of concerns*

Conflict detection

Automatic composition

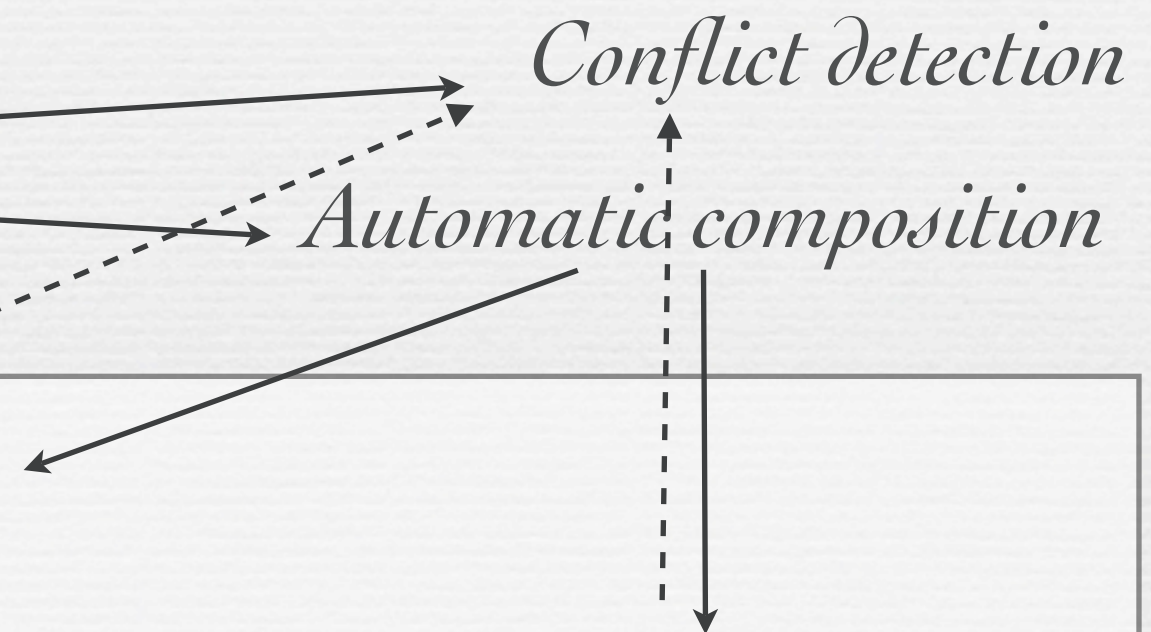
Application property &
quality of service preservation

Activity preservation /Mutualization

Order/Condition preservation

Language conformance

Consistency

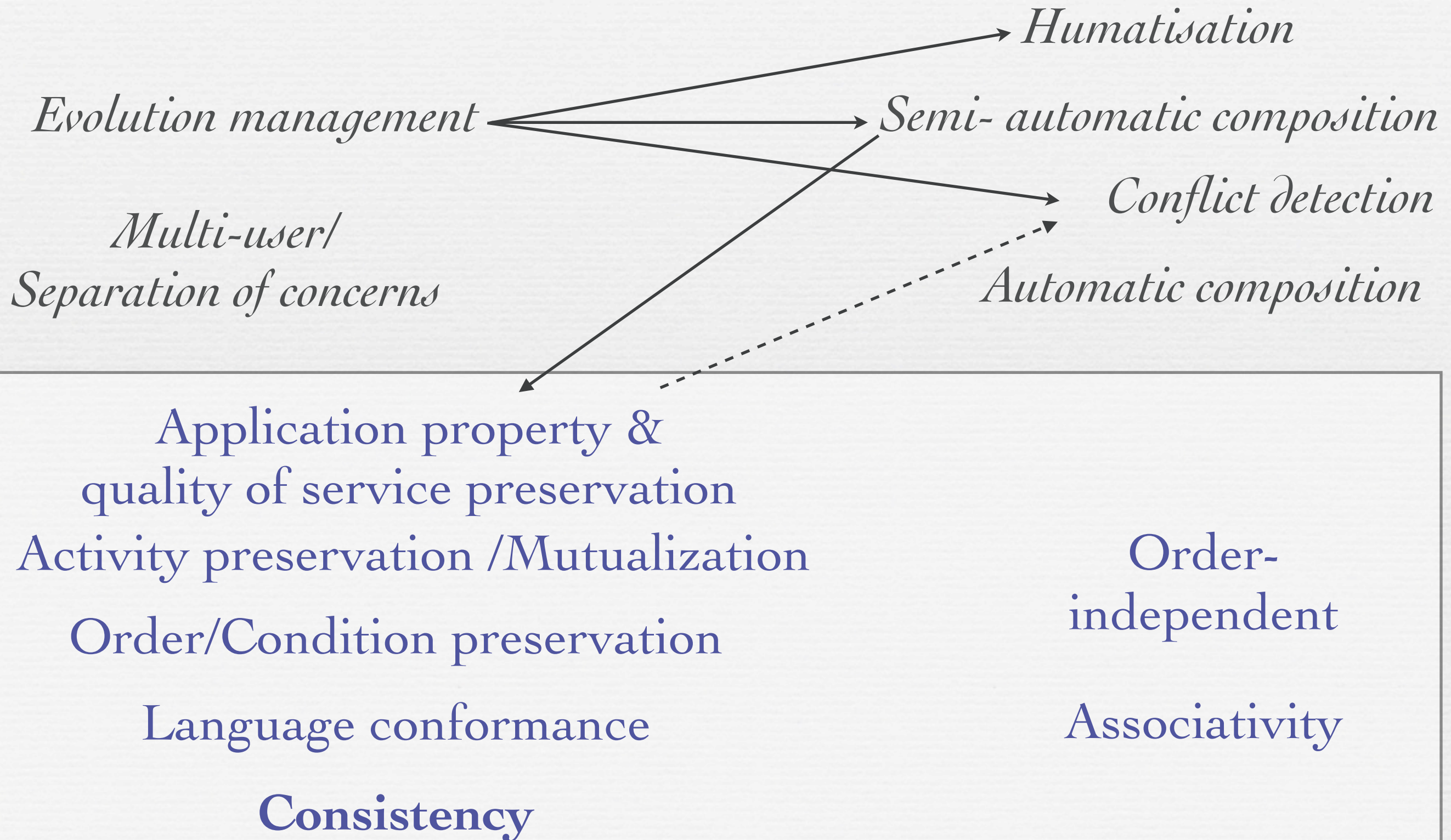


Order-
independent

Associativity



Requirements & Properties Summary



Building Evolutive Complex Systems

Language Directed Composition
(LDC)

ex : Interaction Composition

- ☒ Order-independent composition
- ☒ Associativity
- ☒ Condition preservation



Language Directed Composition : Interactions

- ❧ Language Definition :
 - ~ Interaction Specification Language (ISL)
- ❧ Composition Algorithm :
 - ~ Definition, Proof of properties (rewriting rules)
- ❧ Implementation & Validation :
 - ~ Interoperability, Architecture Pattern, ...

Interaction Composition

.. objects cooperate to perform some tasks or to maintain some invariants. We call such behaviors *inter object dependencies*. ... our work can be seen as the elaboration of the “lego game for inter object dependencies” with which the programmer may construct his own programming environment
[Ducasse 97]

~ Multiple Users \Rightarrow

- ~ Order independence & associativity
- ~ Condition and order preservation

~ RNTL ASPECT 02-04

~ MS Rotor 03-05

~ Dynamic interaction adding/retracting

~ Interactions between heterogeneous components

C#, Java RMI, .Net objects, J2EE- EJB Jonas component



: APP deposit

Stéphane Ducasse, 97 (&P. Franchi): Intégration réflexive de dépendances dans un modèle à classes...

~ L.Berger 01 & A. Ocelllo 06 avec AM. Dery, D. Cheung Foo Woo 09 avec JY Tigli & S. Lavirotte



Use Of Interactions To Manage Framework Consistency

...the use of a framework poses problems related to the respect of its structure and behavior....
[Rapicault 02]

- ❧ Babylon is a language independent tool to assist framework users.
 - ~ Active assistance (information on element to create, rename and delete...)
 - ~ Consistency checking

~ Grant IBM Eclipse 03

~ Pascal Rapicault, 02 (&JP. Rigault): Modèles et Techniques pour spécifier, développer et utiliser un framework : une approche par méta-modélisation



Use Of Interactions To Define Service Integration

The services are only complicated as they need to be [OMG]. In fact programmers spend a lot of time and effort in studying the needed services before being able to use them.[Bussard 2000]

- ❧ Platform-independent description of service integration
 - ~ based on interactions on message flow modeling
- ❧ Static automatic composition of service integrations
 - ~ Conflict detection
 - ~ Partitions of function space

~ RNTL ARCAD 01-03

~ Olivier Nano, 04 (&M. Riveill): Un modèle de réécriture pour l'intégration de services



From Interactions To Activities

- ❧ Different composition algorithms for :
 - ~ Service Integration
 - ~ GUI Composition
 - ~ Device Assembling
- ❧ Workflows : Another way to express interactions?

Building Evolutive Complex Systems By Means Of Software Activity Composition

Toward Software Activity Composition
directed by Models
(meta-composition)
ex : Activity Merging

Properties Of SA Composition: What Does It Mean, Exactly ?

- ❧ Activity Consistency : $a \in A$
- ❧ Activity Composition : $\forall a_1 \in A, \forall a_2 \in A, a_1 + a_2 \in A,$
- ❧ Order-Independent: $\forall a_1, a_2 \in A^2, a_1 + a_2 \equiv a_2 + a_1,$
- ❧ Associativity: $\forall a_1, a_2, a_3 \in A^3, (a_1 + a_2) + a_3 \equiv a_1 + (a_2 + a_3)$
- ❧ Activity Preservation : $\exists a_1' \equiv a_1, a_1' \subset a_1 + a_2$ /duplication?
- ❧ Idempotence : $\forall a_1, a_1 + a_1 = a_1$
- ❧ Order/Condition Preservation ? $a_1 \leq a_2, c \mid a, \dots$
- ❧ Application Property Preservation :

$$a_1 \in A, a_2 \in A, p_1(a_1), p_2(a_2) \Rightarrow p_1(a_1 + a_2) \wedge p_2(a_1 + a_2)$$



Abstraction & Domains

Abstraction as Free Interpretation

Domain-Free Activity Meta-Model $\equiv, \subseteq, \text{exclusion}$

Domain-Free Composition *preservation, order-independence, ...*

Evaluation Function Specification *constraints, properties ...*

Interpretations on Domain

Activity Meta-Model on Domain D $\equiv D$

Evaluation Functions on Domain *constraint checking*



Abstraction & Domains

Abstraction as Free Interpretation

Domain-Free Activity Meta-Model

$\equiv, \subseteq, \text{exclusion}$

Domain-Free Composition

preservation, order-independence, ...

Evaluation Function Specification

constraints, static properties ...

Interpretations on Domain

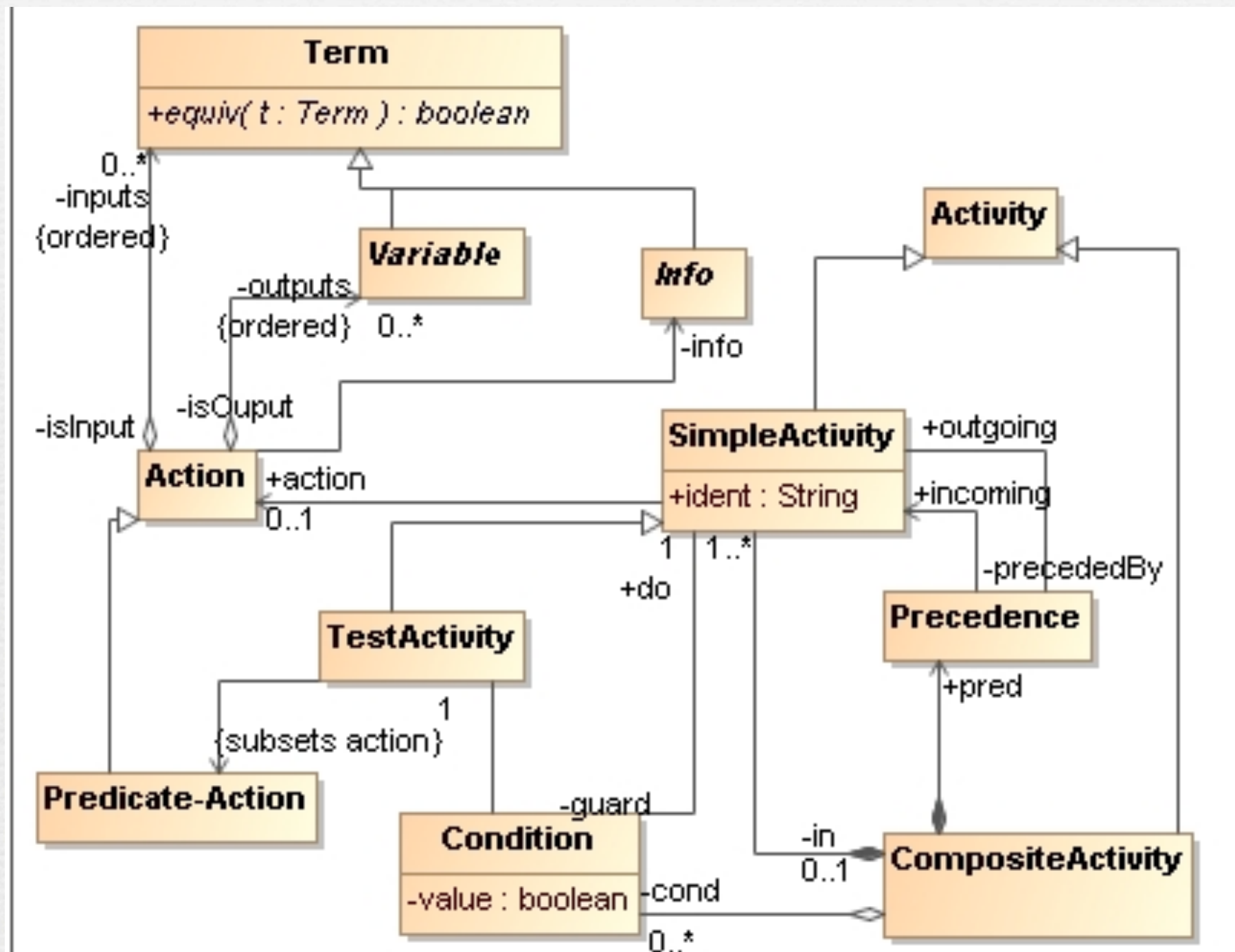
Activity Meta-Model on Domain D

\equiv_D

Evaluation Functions on Domain

constraint checking

Activity Meta-Model (MM4CA)





Activity Meta-model

- A composite Activity is an acyclic oriented graph

$G = \{A, P\}$ where

A : Set of Activities

P : Edge Set labeled with $\{\varepsilon, \text{true}, \text{false}\}$

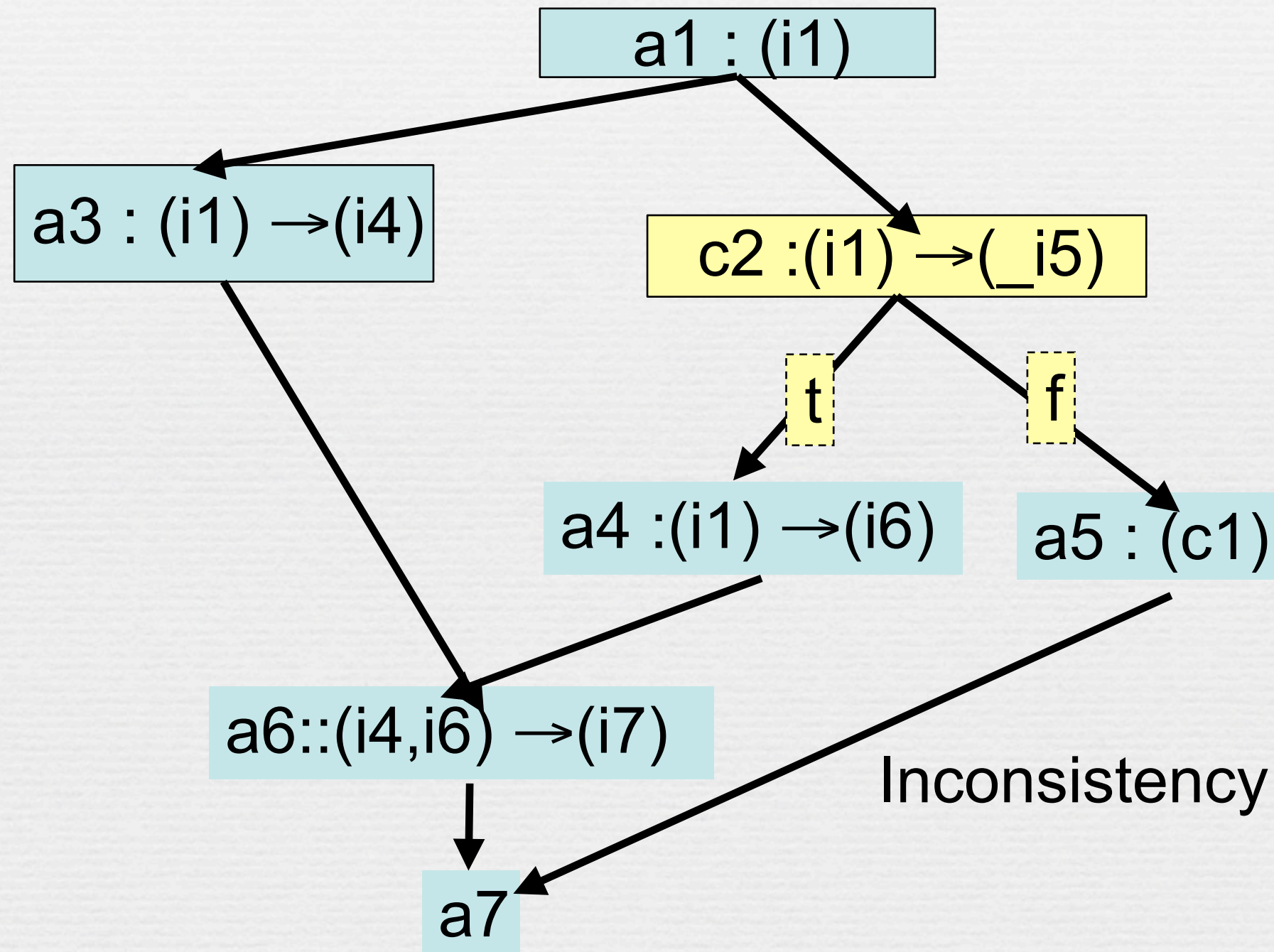
Constraints

G : Acyclic Graph

$\forall t \in A, \forall a \in A, \neg \exists ((t, a, \text{true}) \in P \ \& \ (t, a, \text{false}) \in P)$

$\forall t \in A, a1 \in A, a2 \in A, v \in \{\text{true}, \text{false}\},$
 $(a1, a2, \varepsilon), (t, a1, v) \Rightarrow (t, a2, v)$

Composite Activity : Example



Equivalence

- Two actions $a1$ et $a2$ are **equivalent**, if $\exists \sigma$,
 - $\sim \sigma(\text{inputs}(a1)) = \sigma(\text{inputs}(a2)) \wedge$
 - $\sim \sigma(\text{outputs}(a1)) = \sigma(\text{outputs}(a2)) \wedge$
 - $\sim \sigma(\text{info}(a1)) \equiv \sigma(\text{info}(a2))$ (Domain dependent).

Abstraction & Domains

Abstraction as Free Interpretation

Domain-Free Activity Meta-Model $\equiv, \subseteq, \text{exclusion}$

Domain-Free Composition *preservation, order-independence, ...*

Evaluation Function Specification *constraints, static properties ...*

Interpretations on Domain

Activity Meta-Model on Domain D $\equiv D$

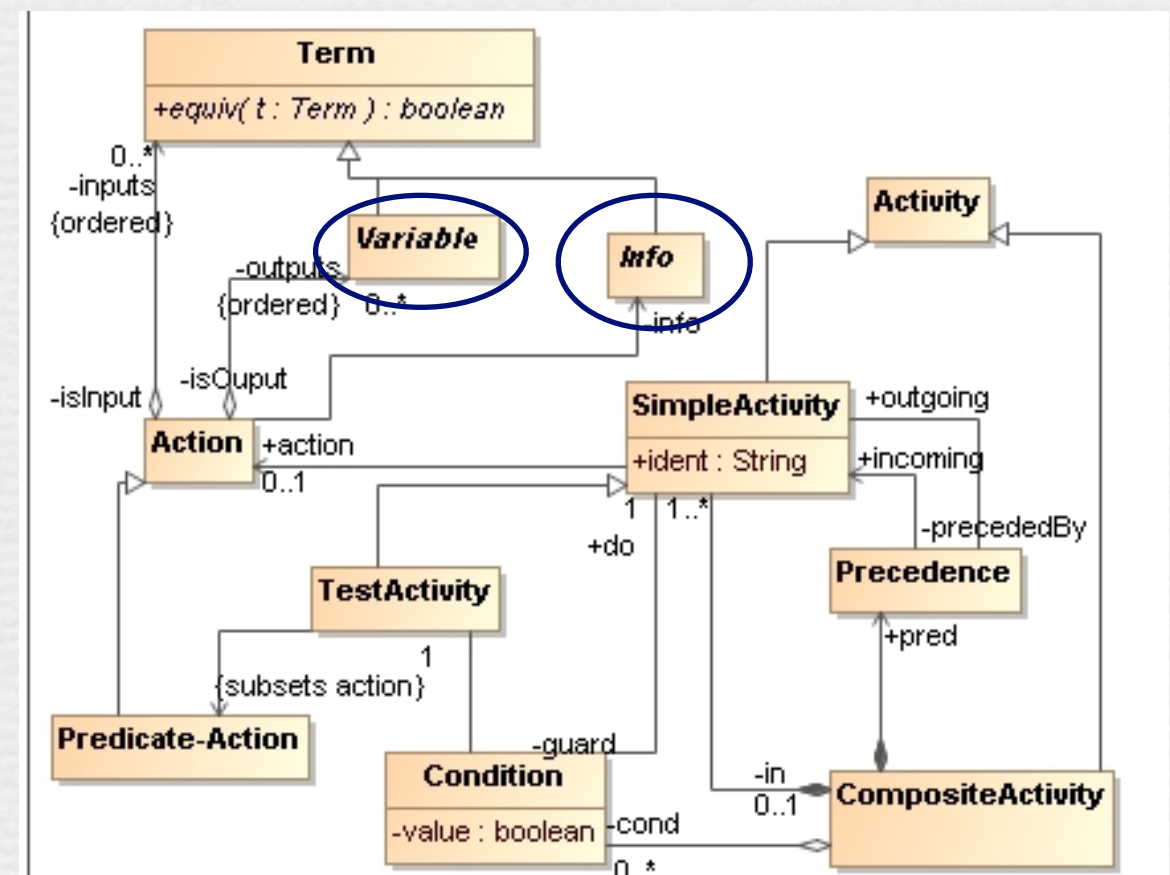
Evaluation Functions on Domain *constraint checking*

Activity Domain

❧ MM4CA Refinement

~ *Information and Variable Specialization*

~ Adding New Constraints



Activity Merging

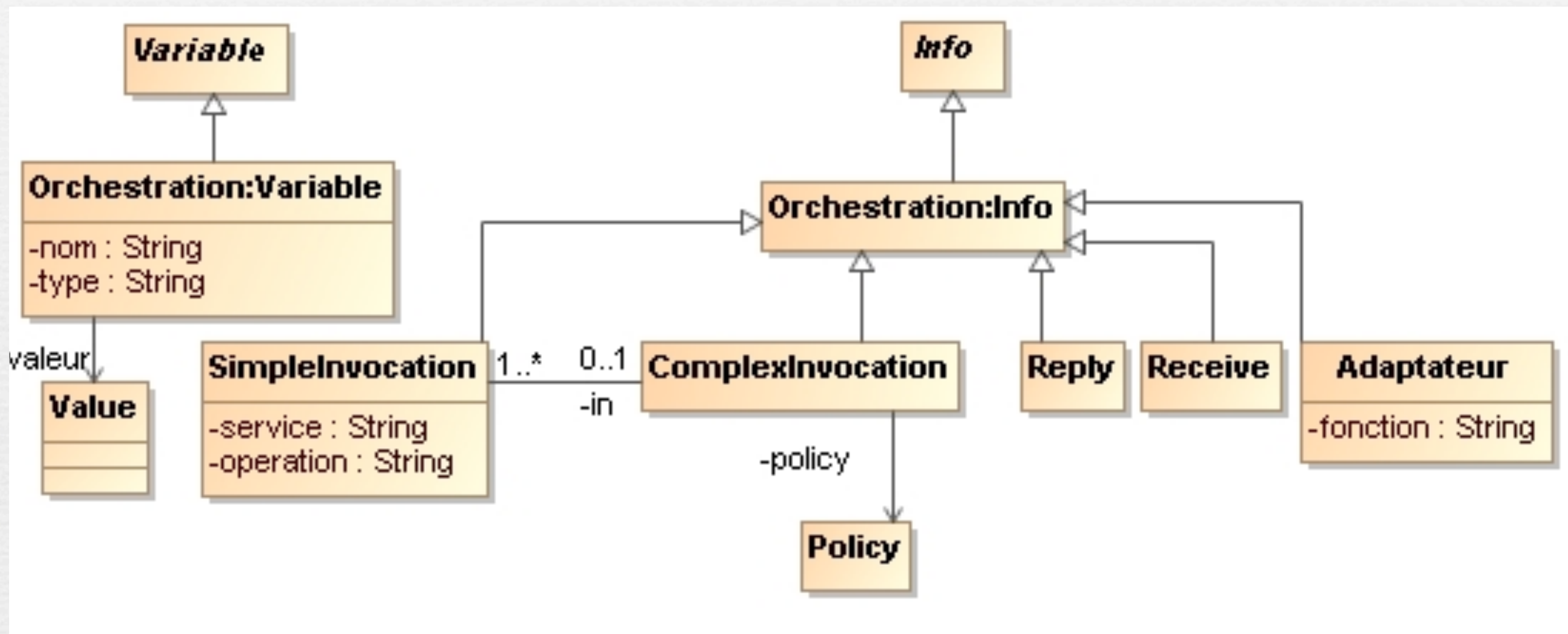
Merging Orchestrations is a crucial issue in the development process of service-based applications.
However merging orchestrations with overlaps is a manual and tedious process today
[Nemo-Glatard 07]

Evolution Management

Semi-Automatic Composition (Humatisation)

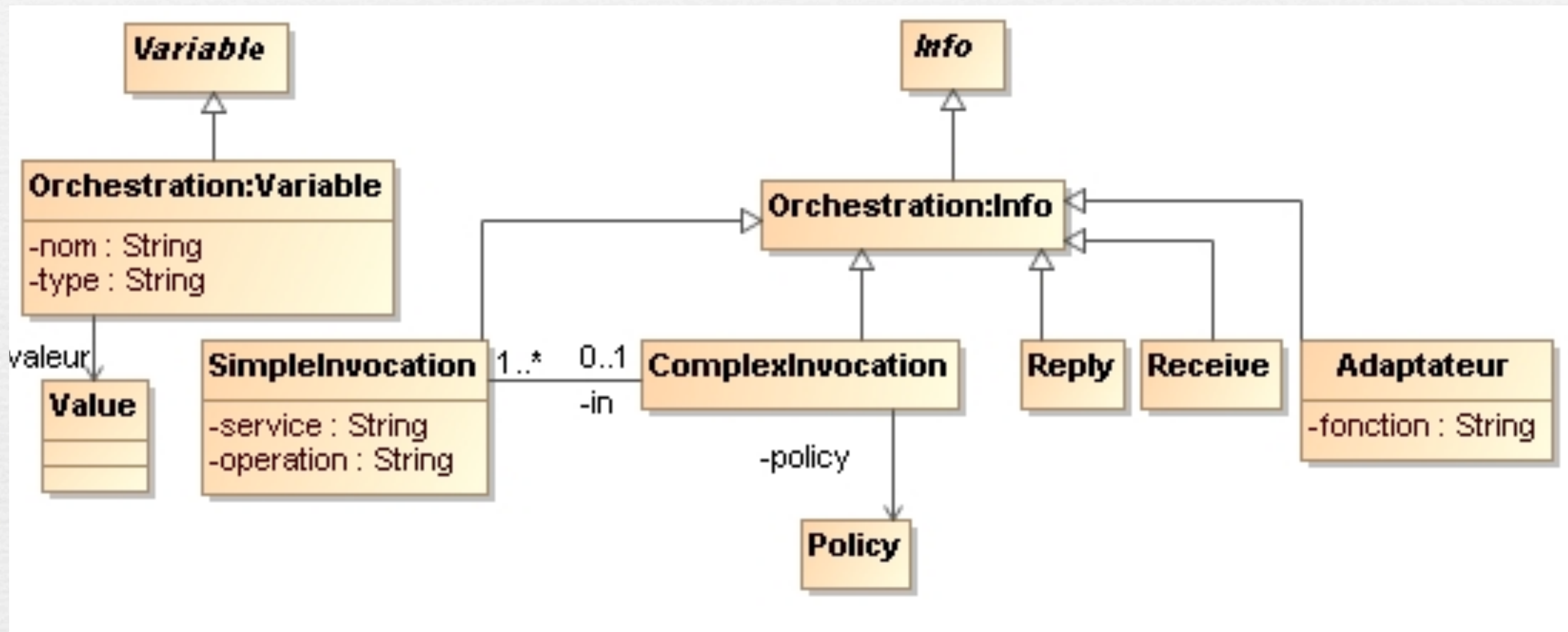
- ☒ Mutualization
- ☒ Quality of service
- ☒ Activity preservation
- ☒ Consistency

Orchestration Domain



- *Receive* activity is the root of the graph
- Only one *Receive* Activity and one *Reply* Activity
- Several *simple invocation* activities referring to a same operation refer to a unique *complex Invocation*

Orchestration Domain



Two simple invocations are **equivalent** if they refer to the same service and operation. The reference to a complex invocation isn't taken into account.



Abstraction & Domains

Abstraction as Free Interpretation

Domain-Free Activity Meta-Model $\equiv, \subseteq, \text{exclusion}$

Domain-Free Composition *preservation, order-independence, ...*

Evaluation Function Specification *constraints, static properties ...*

Interpretations on Domain

Activity Meta-Model on Domain D $\equiv D$

Evaluation Functions on Domain *constraint checking*

Domain-Free Composition

$\oplus (g_1, \dots, g_n)$

- ❧ Merging Pair Detection :
 - ~ $\text{detectMergePairs}_D((g_1, \dots, g_n)) \rightarrow \{(a_1, a_2), (a_3, a_4), \dots\}$
- ❧ Computing transformations on every merging sets
 - ~ $\text{mergeSimpleActivities}_D(\{a_1 \dots a_k\}, P, \{a_1 \dots a_n\}) \rightarrow$
 $(\text{NewActivities}, \text{NewR}, \text{Transformations})$
- ❧ Applying transformations
 - ~ $\Sigma_D(\{T_1 \dots T_n\}) \rightarrow T, T(\{a_1 \dots a_n\}, R) \rightarrow (\{a'_1 \dots a'_p\}, R')$
- ❧ Partial Normalization
- ❧ Normalization on Domain



Domain-Free Composition $\oplus (g_1, \dots, g_n) :$ Evaluation Functions

- ❧ Merging Pair Detection
 - ~ $\text{detectMergePairs}_D((g_1, \dots, g_n)) \rightarrow \{(a_1, a_2), (a_3, a_4), \dots\}$
- ❧ Computing transformations on every merging set
 - ~ **$\text{mergeSimpleActivities}_D(\{a_1 \dots a_k\}, P, \{a_1 \dots a_n\}) \rightarrow (\text{NewActivities}, \text{NewR}, \text{Transformations})$**
- ❧ Applying transformations
 - ~ $\Sigma_D(\{T_1 \dots T_n\}) \rightarrow T, T(\{a_1 \dots a_n\}, R) \rightarrow (\{a'_1 \dots a'_p\}, R')$
- ❧ Partially Normalizing
- ❧ Normalizing on Domain

Evaluation Function Specification

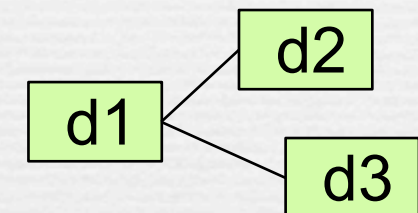
Determining Transformations

s1

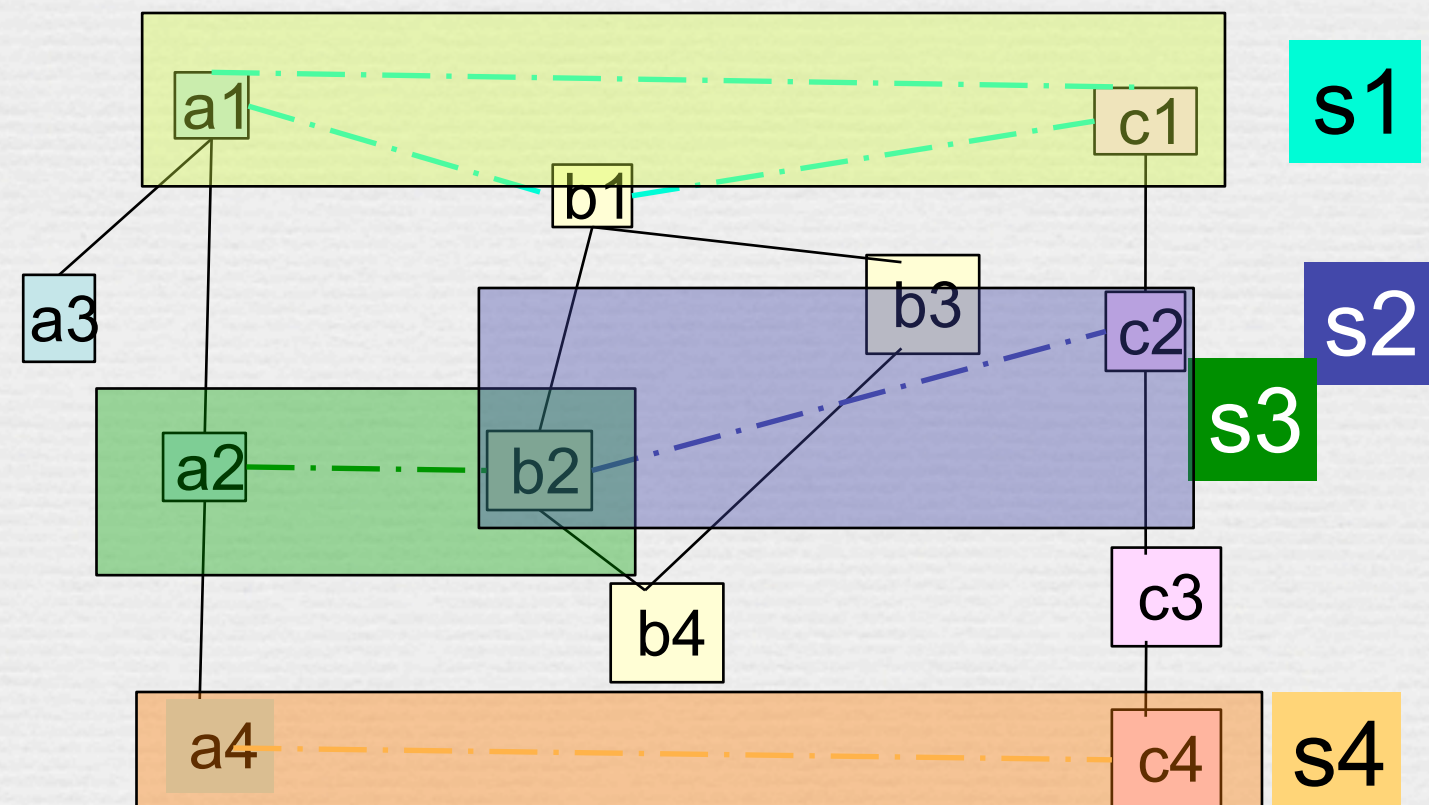
Transformations to apply

$\{t11, t12, t13...\}$

New activities and Relations

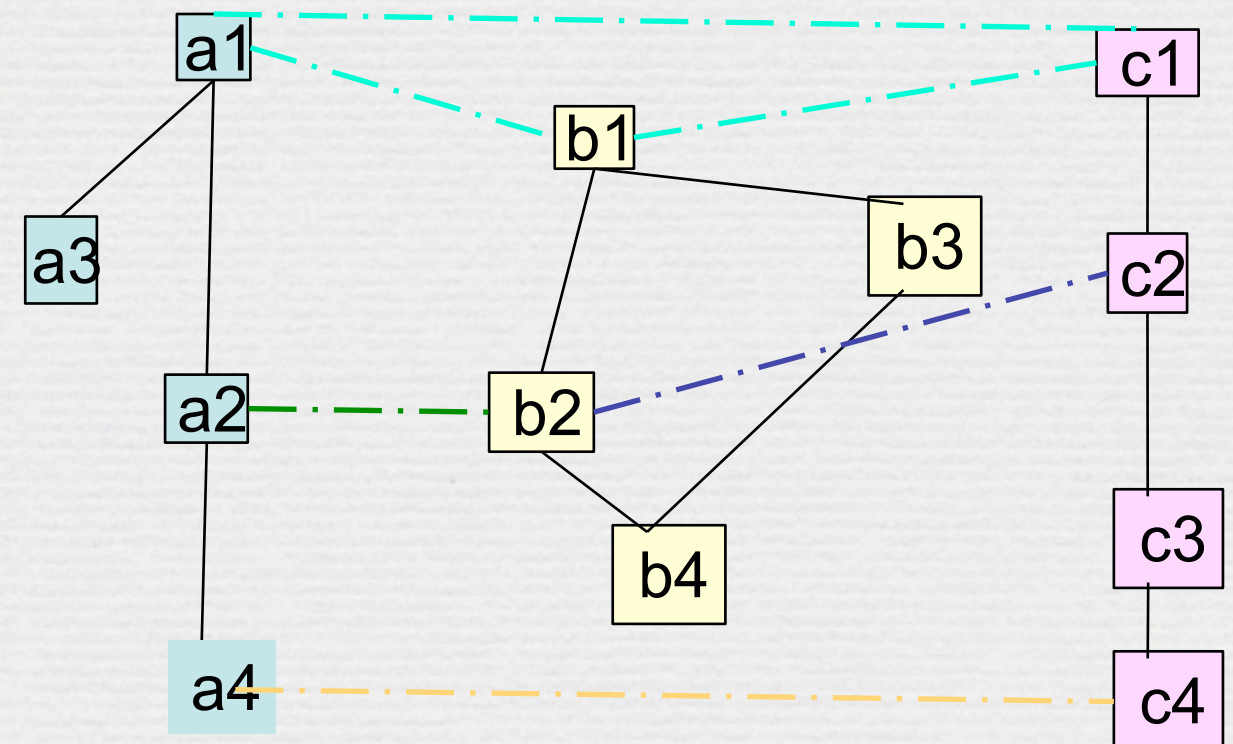
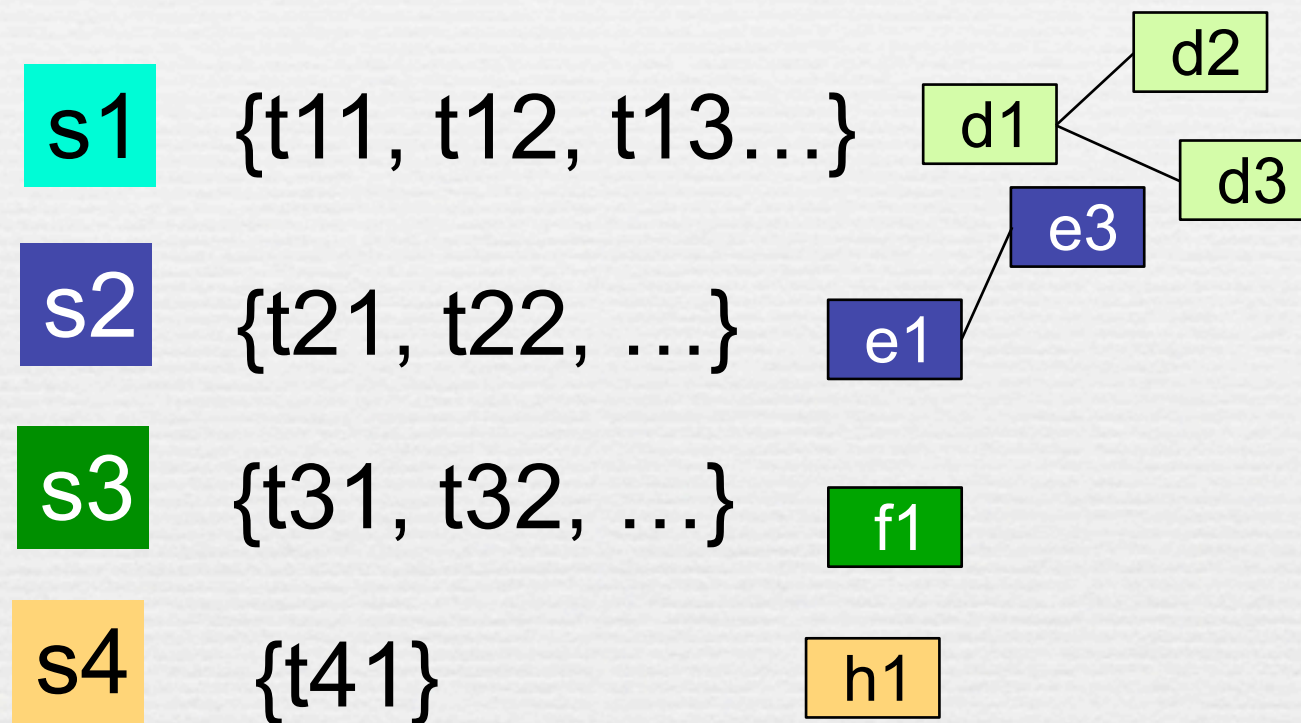


Merging Conflict Detection



Evaluation Function Specification

Determining Transformations



Determining Transformations : Properties

- ❧ Transformations must not introduce local circuit and inconsistency
- ❧ *Activity Order Independent Composition*
 - ⇒ Determining Transformations is not activity order dependent.
- ❧ *Property Preservation* ($P = \text{Activity}|\text{Relation}$)
 - ⇒ No remove of P without introducing a new equivalent P'
- ❧ Associativity

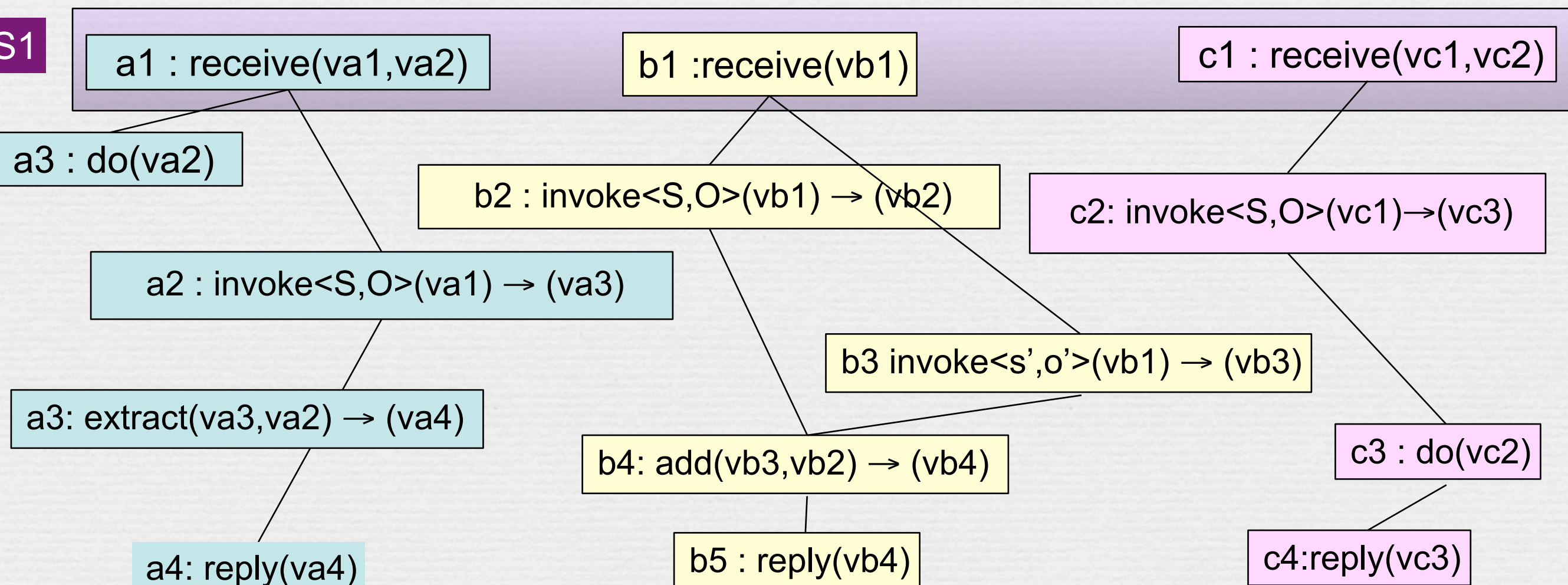
Evaluation Function : Determining Transformations In Orchestration Domain

New activity : $d1 : \text{receive}(v1, vc2)$

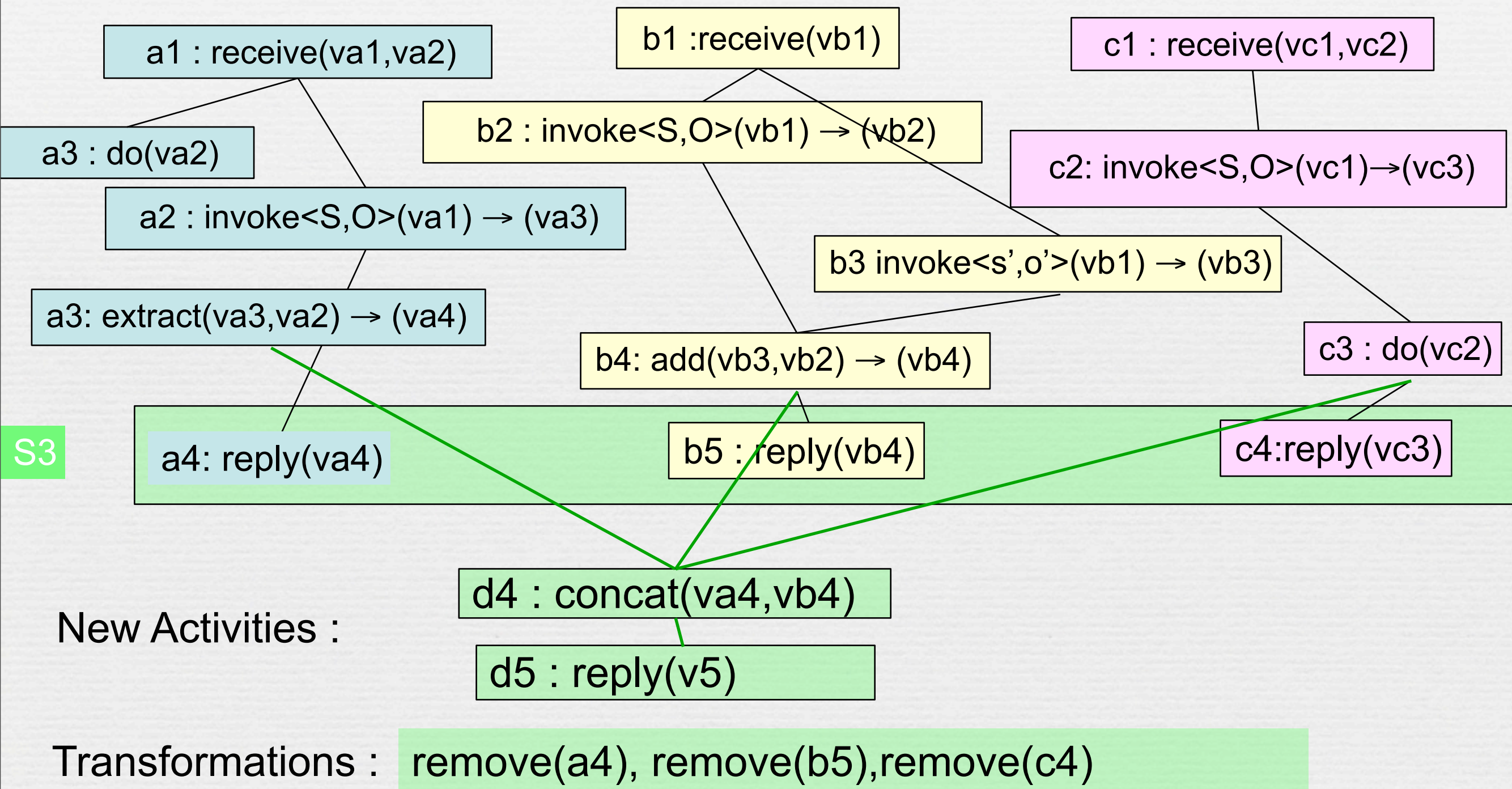
Transformations : $va1 \leftrightarrow vb1 \leftrightarrow vc1 \Rightarrow v1$

$\text{replace}(a1, d1), \text{replace}(b1, d1), \text{replace}(c1, d1)$

S1



Evaluation Function : Determining Transformations In Orchestration Domain



Determining Transformations Properties

❧ Transformations must not introduce local circuit and inconsistency

❧ *Activity Order Independent Composition*

$D_{\text{orchestrations}}$: Yes

⇒ Determining Transformations is not activity order dependent.



Property Preservation ($P = \text{Activity} | \text{Relation}$)

$D_{\text{orchestrations}}$: Yes^b

⇒ No remove of P without introducing a new equivalent P'



Associativity

$D_{\text{orchestrations}}$:

activities : No, *receive* can be modified.

relations : Yes

$D_{\text{orchestrations}}$: No, user can add precedences.

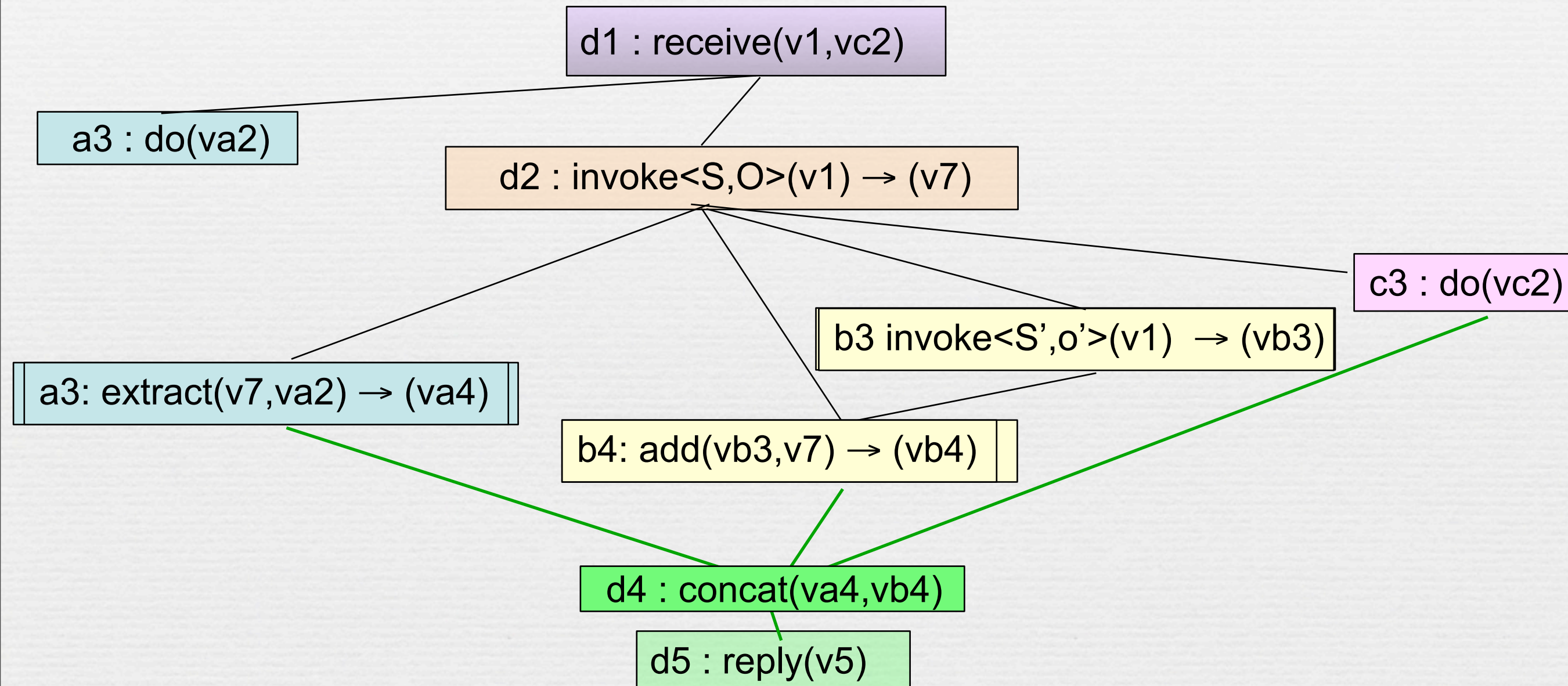
Composition $\oplus (g_1, \dots, g_n) :$ Principles

- ❧ Merging Pair Detection
 - ~ $\text{detectMergePairs}_D((g_1, \dots, g_n)) \rightarrow \{(a_1, a_2), (a_3, a_4), \dots\}$
- ❧ Computing transformations every each merging set
 - ~ $\text{mergeSimpleActivities}_D(\{a_1 \dots a_k\}, P, \{a_1 \dots a_n\}) \rightarrow$
 $(\text{NewActivities}, \text{NewR}, \text{Transformations})$
- ❧ Applying transformations : **activity order dependent ?**
 - ~ $\Sigma D(\{T_1 \dots T_n\}) \rightarrow T, T(\{a_1 \dots a_n\}, R) \rightarrow (\{a'_1 \dots a'_p\}, R')$
- ~ Partially Normalizing : **=> acircuitic Consistent Graph (Closure)**

Number of Merging Sets Decreasing

- ❧ Normalizing on Domain

Result Of $\oplus(g_1, g_2, g_3)$



From Model Composition To Platforms

- ❧ At Model Level : Concepts and Properties
- ❧ According to objectives, composition implementation
 - ~ at model level,
 - ~ or at platform level

Software Activity Composition

- PhD Thesis in progress

- Perspectives

Control Flow

Dealing With Data Set

.. workflow languages lack the necessary modularization techniques and data flow capabilities to express processes in a way that facilitates their design, evolution and reuse
[Joncheere & All 2008]

- ❧ Definition of Orchestration Evolutions including
 - ~ Data Set management
 - ~ Error management
- ❧ **Evolution Composition** enforcing order and condition preservations.
- ❧ Applications : SEDUITE, Scientific Workflows

~ Faros 06-09

~ Sébastien Mosser, end 2010? : Evolution d'orchestrations

Ensuring Safe Pattern Composition

Advanced techniques in SOC... helps to develop more efficient applications... However conflicts may appear when several concerns need to be composed for the same application...

[Pawlak & All 2005]

Application building based on components assembly and patterns

- A pattern implies a set of constraints : How to enforce them?
 - A pattern P consists of a “conforming” transformation (t) which defines an implicit constraint $(c) : t(a) \Rightarrow c(t(a))$
 - Patterns Composition implies at least : **idempotence** $t(t(a))=t(a)$

• Région DCNS 06-09

• Clémentine Nemo, begin 2010? : Construction et Validation d'assemblages de Composants

Another Experiment

Weaving Multiple Aspects In Sequence Diagram

We have proposed a technique to statically weave behavioral aspects into sequence diagrams.
[Klein & All 2007]

- ❧ Merging pairs detection should be extended to two activity sets.
- ❧ Could we use transformation composition in order to determine the best solution?

Quality Of Services & Evolutions

- How to help users to define accurate workflows and to adapt them in a safe way according to changes at design time and runtime ?

~ Salty?/Maat-G

Building Evolutive Complex Systems

Logic &
Objects

th: Ducasse 97

P: ARCAD 01-03

P: Eclipse 03

th: Rapicault 02

P: ASPECT 02-04

P: MS Rotor 03-05

th: Nano 04



RSTI 04

P: Faros 06-09

Meta-Composition :
Model directed Composition

Book : IDM 06

Custom solution

th: Mosser 2010?

RSTI 07, 08; SCC 07,
ICEIS 07

IAWTIC 08,
ICIW 09,
ECSA'08

Evolution Management

th: Nemo 2010?

C: DCNS 06-09

QoS

th: 2013?

In Short

- ❧ PhD Thesis : 3 PhD, 2 in progress
- ❧ Publications (4 last years) : 1 book, 1 international journal(JOT), 6 national Journals, 3 book chapters, 2 proceedings(LMO, IDM), 6 international conferences (SCC, ICIW, ECSA, IAWTIC, ICEIS, IADIS) , Workshops (EduSymp,SPLAT...), ...
- ❧ APP Deposit : Noah
- ❧ Research Projects (4 last years) : Faros, MobiVIP, MS Rotor, IBM, AS IDM,...
- ❧ Responsibilities : Action IDM, LMO Steering committee
- ❧ Program Committees (4 last years): MDD4RES, Chairwoman: LMO08, IDM09, Co-Chair Numero Special TSI sur l'IDM

With

Michel

Anne-Marie

Claudine

Johan

Yves

Jean-Marc

Olivier

Rose

Gilles

Sébastien

Claude

Laurence

Jean-Yves

Jean-Paul

Viviane

Stéphane L.

Daniel

Audrey

Paul

Philou

Laurent

David

Clémentine

Sabine M.

Stéphane D.

Pascal

Christian

Philippe

Diane

Dominique, Héloïse, Jérémy

...

Sabine B.

Thanks

Les Fleurs du mal, IV, *Correspondances*

La Nature est un temple où de vivants piliers
Laissent parfois sortir de confuses paroles ;
L'homme y passe à travers des forêts de symboles
Qui l'observent avec des regards familiers.

Comme de longs échos qui de loin se confondent
Dans une ténébreuse et profonde unité,
Vaste comme la nuit et comme la clarté,
Les parfums, les couleurs et les sons se répondent.

Il est des parfums frais comme des chairs d'enfants,
Doux comme les hautbois, verts comme les prairies,
Et d'autres, corrompus, riches et triomphants,

Ayant l'expansion des choses infinies,
Comme l'ambre, le musc, le benjoin et l'encens
Qui chantent les transports de l'esprit et des sens.