A formal denotation of complex softwares

Pascale LE GALL

Joint work with Marc Aiguier

IBISC FRE 2873 - Université d'Évry Val d'Essonne
Programme d'Epigenomique - Genopole
Evry France

23th March, 2007
IFIP Working Group 1.3

Context : the GENNETEC project 2006-2009

GENetic NETworks : Emergence and Complexity
(european STREP project FP6)

General objective :

"Develop scalable computational modelling and inference tools and scalable simulation techniques for complex systems"

Our particular sub-objective (WP1) :

"Develop a theoretical framework for modelling complex systems and for analysis of their emergent properties, inspired by the biological case study"

Targeted application domain

Genetic Regulatory Network (GRN)

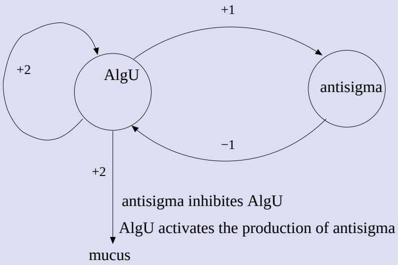
- qualitative description with the discrete asynchroneous modelisation of R. Thomas
- Unkown parameters
  $\Longrightarrow$ set of models
- Behaviours expressed as temporal properties

Complex systems ?

GRN are systems open to their environment :
they represent a biological function under study which can be embedded in a larger network

Example

Mucus production in P Aeruginosa



Mucus production occurs when the discrete value of AlgU is greater than the threshold 2
($\Longrightarrow$ diseased lung)

Other natural application domains in the field of software engineering (SE)

Open complex systems are common in SE

- oriented-object design

  (active objects with an execution model involving true concurrency)

- service-oriented design (services are also called features)

  Due to some conflicts (or interactions) between two features, the integration of a new feature on an existing system can modify the expected properties of the underlying system

What is complexity ? An informal starting point

Initial assumption

A complex system is more than the set of its components

- Systems depend on the way components interact, i.e. on the connectors (glue) used to link subsystems together
- Adding a component can modify properties inherited from a given underlying subsystem

Informal definition

A system is said to be complex when systems can inherit from its components some properties which cannot be anticipated from the knowlegde issued from the components.

Our aim

to propose a formal denotation of complex systems provided with a characterisation of emerging properties

Which formal elements to consider ?

- institutions to abstractly and generically denote
  - signatures (interfaces),
  - formulas (properties),
  - models (systems),
  - and satisfaction (verification of properties by systems)
- a language of system specifications
  - expressed in the institutional framework
  - allowing us to manipulate properties associated to specifications
- an institution-independant denotation for connectors building systems from subsystems

Plan

- Language of specifications in a institutional framework
- Definition of specification connectors, classified as
  - modular for composite systems without emerging properties
  - complex for composite systems with emerging properties
- Classification of emerging properties as
  - non conformant properties
  - or “true” emerging properties

An abstraction of specification formalisms

Definition

An institution $\mathcal{I} = (Sig, Sen, Mod, \models)$ consists of

- a category Sig of signatures,
- a functor $Sen : Sig \to Set$ giving for each signature Σ a set, element of sentences,
- a contravariant functor $Mod : Sig^{op} \to Cat$ giving for each signature Σ a category of Σ-models
- a $|Sig|$-indexed family of satisfaction relations $\models_\Sigma \subseteq |Mod(\Sigma)| \times Sen(\Sigma)$

such that the satisfaction condition holds :
$\forall \sigma : \Sigma \to \Sigma', \forall \mathcal{M}' \in |Mod(\Sigma')|, \forall \varphi \in Sen(\Sigma),$

$$\mathcal{M}' \models_{\Sigma'} Sen(\sigma)(\varphi) \Leftrightarrow Mod(\sigma)(\mathcal{M}') \models_\Sigma \varphi$$

Classical examples of institutions

Many variations combining propositional logic, first order logic, (typed or not) equational logic, Horn Clause

- Propositional Logic (PL)
- Many-sorted First Order Logic with equality (FOL)
- Horn Clause Logic (HCL)
- Equational Logic (EQL)
- Conditional equational logic (CEL)
- Rewriting Logic (RWL)

Modal First Order Logic (MFOL)

- Signatures (Σ, A) are composed of a First Order Logic with equality (FOL) signature $\Sigma = (S, F, P)$ and of a set A of actions
- (Σ, A)-formulae are built over :
  - FOL formulae over Σ
  - and modalities in $\{\Box_a | a \in A\}$
- A (Σ, A)-model (W, R), called a Kripke frame, consists of
  - a family $W = (W^i)_{i \in I}$ of FOL Σ-models s.t. $W^i_s = W^j_s$ $(i, j \in I, s \in S)$
  - and "accessibility" relations $\{R_a \subseteq I \times I\}_{a \in A}$.
- For a signature morphism $\sigma : (\Sigma, A) \to (\Sigma', A')$ and a (Σ', A')-model (W', R'), $Mod(\sigma)((W', R'))$ is the (Σ, A)-model (W, R) defined by $W = Mod(\sigma)(W')$ and $\{R_a = R'_{\sigma(a)}\}_{a \in A}$.
- $(W, R) \models_{(\Sigma, A)} \varphi$, if for every $i \in I$, we have $(W, R) \models^i_\Sigma \varphi$ s.t.
  - atoms, Boolean connectives and quantifiers are handled as in FOL,
  - $(W, R) \models^i_\Sigma \Box_a \varphi$ when $(W, R) \models^j_\Sigma \varphi$ for every $j \in I$ s.t. $i \; R_a \; j$.

Specifications in institutions

Specifications

A *specification language* $\mathcal{SL}$ over an institution $\mathcal{I} = (Sig, Sen, Mod, \models)$ is a pair $(Spec, \_^{\bullet})$ where :

- $Spec : Sig^{op} \to Set$ is a functor, and
- $\_^{\bullet} = (\_^{\bullet}_{\Sigma})_{\Sigma \in Sig}$ is a *Sig*-indexed familly of mappings

$$\_^{\bullet}_{\Sigma} : Spec(\Sigma) \to \mathcal{P}(Sen(\Sigma))$$

Category of specifications

The category *SPEC* of specifications over $\mathcal{I}$ is s.t. :

- objects are objects of $Spec(\Sigma)$ for every signature $\Sigma \in Sig$
- morphisms are every arrow σ from Sp to Sp' s.t.
  - $\sigma : Sig(Sp) \to Sig(Sp')$ is a signature morphism
  - and $Sen(\sigma)(Sp^{\bullet}) \subseteq Sp'^{\bullet}$.

Notation : $Sig(Sp) = \Sigma \Longleftrightarrow Sp \in Spec(\Sigma)$.

Illustration (1) : specifications

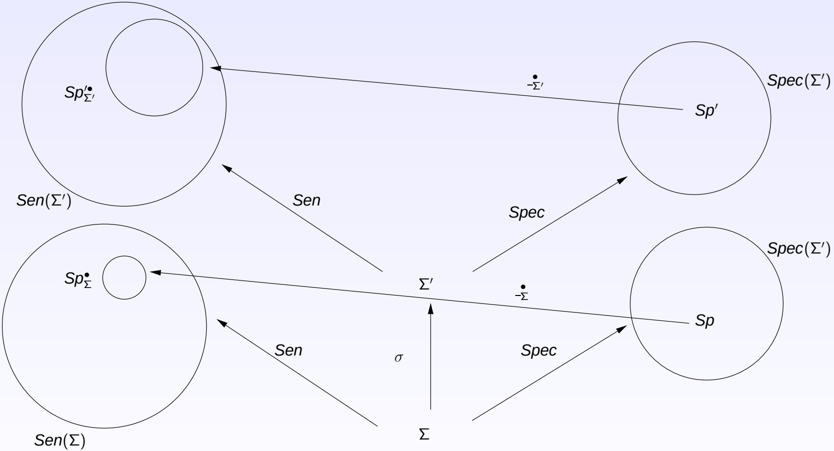
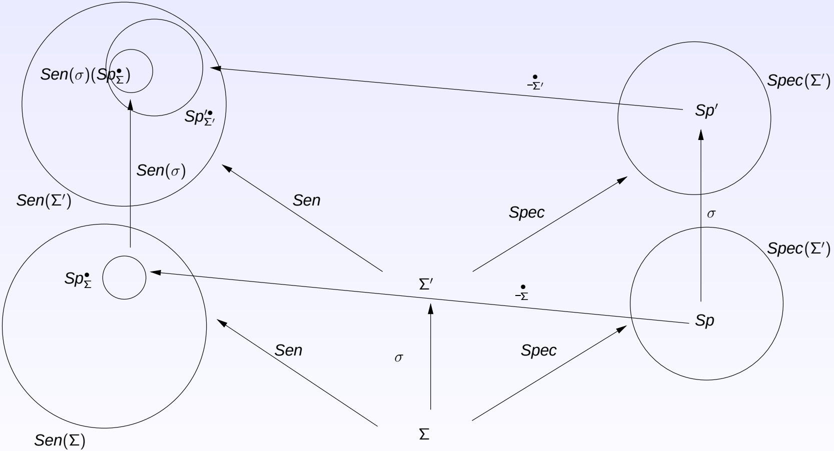


Illustration (2) : category of specifications



Models of specifications

Specifications Sp are already defined by their signatures $Sig(Sp)$, their properties $Sp^{\bullet}_{Sig(Sp)}$. They are also defined by their models :

Specification models

Let Sp be a specification in $Spec(\Sigma)$.

$Mod(Sp)$ is the full subcategory of $Mod(Sig(Sp))$ whose objects, called models of Sp, are $Sig(Sp)$-models $\mathcal{M}$ s.t. :

$$\forall \varphi \in Sp^{\bullet}_{Sig(Sp)}, \mathcal{M} \models_{Sig(Sp)} \varphi$$

Property

Let $\sigma : Sp \rightarrow Sp'$ be a specification morphism.

$Mod(\sigma) : Mod(Sig(Sp')) \rightarrow Mod(Sig(Sp))$ can be restricted :

$$Mod(\sigma) : Mod(Sp') \rightarrow Mod(Sp)$$

Illustration (1) : models of a specifications

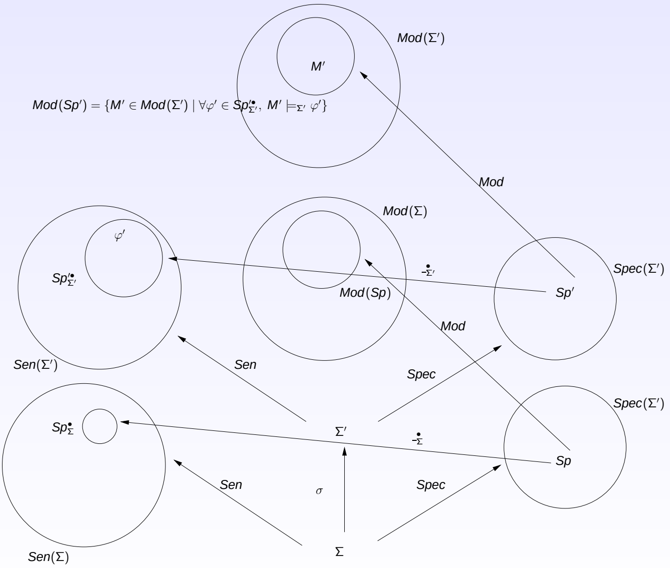
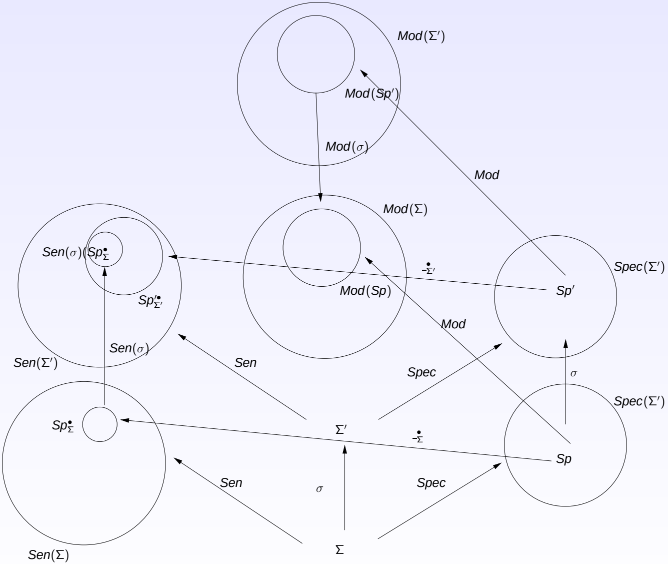


Illustration (2) : models of a specifications



Specifications as logical theories in an institution $\mathcal{I} = (Sig, Sen, Mod, \models)$

A Σ-theory is a set of Σ-sentences T s.t $T = T^\bullet$ where :

- $T^\bullet = \{\varphi \mid \forall \mathcal{M} \in Mod(T), \mathcal{M} \models_\Sigma \varphi\}$
- and $Mod(T) = \{\mathcal{M} \in Mod(\Sigma) \mid \forall \varphi \in T, \mathcal{M} \models_\Sigma \varphi\}$.

$Spec : Sig^{op} \to Set$ associates

- to each $\Sigma \in Sig$ the set of all T of $Sen(\Sigma)$ s.t. $T = T^\bullet$,
- and to each $\sigma : \Sigma \to \Sigma'$, the application matching to each T' of $Sen(\Sigma')$ with the subset $T = \{\varphi \mid Sen(\sigma)(\varphi) \in T'\}$.

Remark :

A Σ-theory T is often described by a finite set of axioms $Ax \subseteq Sen(\Sigma)$, s.t. $Ax^\bullet = T$
Specifications (Σ, Ax) then verify : $(\Sigma, Ax)^\bullet_\Sigma = Ax^\bullet$.

Specifications as transition systems for MFOL

System transitions $(Q, \mathbb{T})$ are defined by :

- Q is the set of states, and
- $\mathbb{T} \subseteq Q \times A \times Sen(\Sigma) \times Q$.

For $\sigma : (\Sigma, A) \to (\Sigma', A')$ and $\mathcal{S}' = (Q', \mathbb{T}')$ over (Σ', A'), $Spec(\sigma)(\mathcal{S}')$ is $\mathcal{S} = (Q, \mathbb{T})$ over (Σ, A) s.t.

- $Q = Q'$
- $\mathbb{T} = \{(q, a, \varphi, q') | (q, \sigma(a), \sigma(\varphi), q') \in \mathbb{T}'\}$ is a set of transitions.

$Mod(\mathcal{S})$: Models of $\mathcal{S} = (Q, \mathbb{T})$ are (Σ, A)-model (W, R) where W is a Q-indexed family of FOL Σ-models and $R = \{R_a \subseteq Q \times Q\}_{a \in A}$ s.t. :

$$q \; R_a \; q' \iff \exists (q, a, \varphi, q') \in \mathbb{T}, W^q \models_\Sigma \varphi$$

$\mathcal{S}^\bullet_{(\Sigma,A)} = \{\varphi \in Sen((\Sigma, A)) | \forall (W, R) \in Mod(\mathcal{S}), (W, R) \models \varphi\}$.

Remark

Sometimes, an inital state q_0 is identified among all states in Q.

Connectors for building systems from subsystems

Connectors defined by means of "colimit"

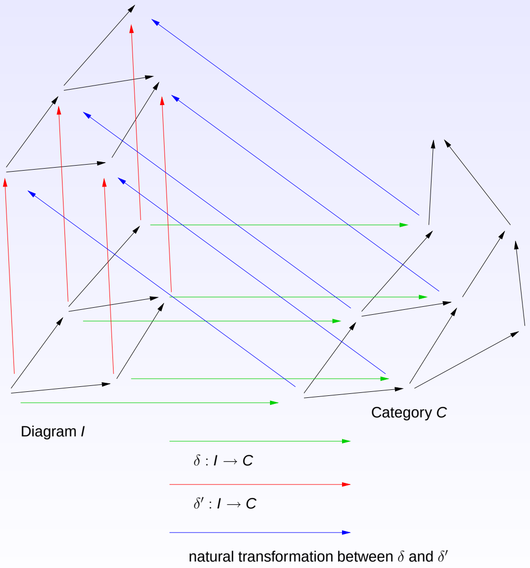
- Intuitively, a colimit captures all minimal information of objects involved in the colimit construction
- Connectors in CommUnity [Fiadeiro and all] for describing architectural description of software systems
- See [Ehresmann and Vanbremersh, Brown, Paton and Porter] for modelisation of (biological) complex systems

diagram category

Let I and C be two categories.
Note $\Delta_{(I,C)}$ the category of diagrams in C with shape I defined by

- objects are functors $\delta : I \to C$,
- morphisms are natural transformations between functors $\delta, \delta' : I \to C$.

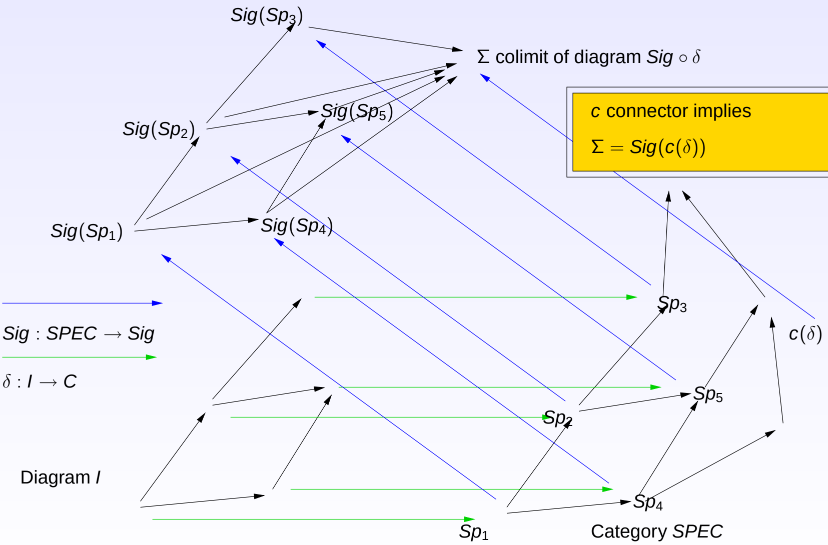
Illustration : diagram category



Let $\mathcal{SL} = (Spec, \_^{\bullet})$ be a specification language over an institution $\mathcal{I}$. An architectural connector $c : |\Delta_{(I,SPEC)}| \to |SPEC|$ is a partial mapping s.t. for each $\delta \in \Delta_{(I,SPEC)}$ for which $c(\delta)$ is defined,

- $Sig(c(\delta))$ is the signature, colimit of the diagram $Sig \circ \delta$.
- δ is equipped with a cocone $p : Sig \circ \delta \to (Sig(c(\delta))$

Illustration : architectural connector



Architectural connector : first examples for logical theories

Enrichment

Consider a shape I composed of two nodes i and j and one arrow $a : i \rightarrow j$.
The connector *Enrich* is defined for any $\delta : I \rightarrow SPEC$ where $\delta(i)$ is a Σ-theory T and $\delta(j)$ a Σ'-theory T' s. t. $Sen(\delta(a))(T) \subseteq T'$.
We define $Enrich(\delta) = T'$ (with Σ' as corresponding colimit signature).

Union

Consider a shape I composed of three nodes i, j, and k and two arrows $a_1 : i \rightarrow j$ and $a_2 : i \rightarrow k$.

The connector *Union* is defined for any $\delta : I \rightarrow SPEC$ where $\delta(i)$ is a Σ_0-theory T_0, $\delta(j)$ a Σ_1-theory T_1 and $\delta(k)$ a Σ_2-theory T_2 and s. t. $Sen(\delta(a_1))(T_0) \subseteq T_1$ and $Sen(\delta(a_2))(T_0) \subseteq T_2$.

It yields the Σ-theory $T = (Sen(p_1)(T_1) \cup Sen(p_2)(T_2))^\bullet$ together with the cocone $p : Sig \circ \delta \rightarrow \Sigma$, pushout of $Sig(\delta(a_1))$ and $Sig(\delta(a_2))$

Architectural connector : synchroneous product of transition systems

Two transition systems can be combined by synchronizing transitions sharing actions.

Let I be a shape composed of three nodes i, j and k and two arrows $a_1 : i \to j$ and $a_2 : i \to k$.

With $\delta(j) = (Q_j, \mathbb{T}_j)$ and $\delta(k) = (Q_k, \mathbb{T}_k)$, $Sync(\delta)$ is the transition system $(Q, \mathbb{T})$ over $(\Sigma_j + \Sigma_k, A_j \cup A_k)$ s. t.

- $Q = Q_j \times Q_k$
- if $a \in A_j \cap A_k$, $(q_j, a, \varphi_j, q_j') \in \mathbb{T}_j$ and $(q_k, a, \varphi_k, q_k') \in \mathbb{T}_k$ then $((q_j, q_k), a, \varphi_j \wedge \varphi_k, (q_j', q_k')) \in \mathbb{T}$
- if $a \in A_j \setminus A_k$ and $(q_j, a, \varphi_j, q_j') \in \mathbb{T}_j$ then for every $q_k \in Q_k$, $((q_j, q_k), a, \varphi_j, (q_j', q_k)) \in \mathbb{T}$
- if $a \in A_k \setminus A_j$ and $(q_k, a, \varphi_k, q_k') \in \mathbb{T}_k$ then for every $q_j \in Q_j$, $((q_j, q_k), a, \varphi_k, (q_j, q_k')) \in \mathbb{T}$

Complex and modular connectors

A connector will be considered as complex when :

non-conformance properties the resulting system gives more or less behaviors on a component with respect to what is expressed in the component specification.

emerging properties any behavior bringing into play many components cannot be deduced from a complete knowledge of these components.

Otherwise, the connector will be said modular.

Complex and modular connectors

Let $c : |\Delta_{I,SPEC}| \to |SPEC|$ be a connector.
Let δ be a diagram of $\Delta_{I,SPEC}$ s. t. $c(\delta)$ is defined, p denoting the corresponding colimit over signatures.
c is said modular for δ iff :

1. $\forall i \in I, \forall \varphi \in Sen(Sig(\delta(i)))$,
$$\delta(i) \models_{Sig(\delta(i))} \varphi \Longleftrightarrow c(\delta) \models_{Sig(c(\delta))} Sen(p_i)(\varphi)$$
2. $\forall \varphi \in c(\delta)^{\bullet}_{Sig(c(\delta))} \setminus (\bigcup_{i \in I} Sen(p_i)(Sig(\delta(i))))$,
$$\bigcup_{i \in I} Sen(p_i)(\delta(i))^{\bullet}_{Sig(\delta(i))} \models_{Sig(c(\delta))} \varphi$$

c is said complex for δ otherwise.

Feature oriented systems

Features are new capabilities incorporated in systems, possibly by modifying existing behaviors of other features present in the system.

Feature interactions are typical examples of emerging properties

Feature specifications

A feature specification $\mathcal{F}$ is a triple (Sp, Inv, Sp') where :

- Sp and Sp' are specifications of $SPEC$
- $Inv \subseteq Sen(Sig(Sp))$
- $\sigma : Sig(Sp) \to Sig(Sp')$
- $Sen(\sigma)(Inv) \subseteq Sen(\sigma)(Sp^\bullet) \cap Sp'^\bullet$.

Sp is called the required specification of $\mathcal{F}$.
Elements in Inv are called invariants.
Sp' represents properties specific to the feature under specification

Feature integration using the *Integrate* connector

Let I be a shape composed of three nodes i, j, and k and two arrows $a_1 : i \to j$ and $a_2 : i \to k$.

The connector *Integrate* is defined for $\delta : I \to SPEC$ satisfying

- $\delta(i) = (Sp_\emptyset, \emptyset, Sp'_i)$, $\delta(j) = (Sp_\emptyset, \emptyset, Sp'_j)$ and $\delta(k) = (Sp_k, Inv, Sp'_k)$,
- $\delta(a_1) = (Id_{Sp_\emptyset}, \rho'_j : Sp'_i \to Sp'_j)$ and $\delta(a_2) = (Sp_\emptyset \hookrightarrow Sp_k, \rho'_k : Sp'_i \to Sp'_k)$, and
- $Sp_k \hookrightarrow Sp'_j$

and yields $Integrate(\delta) = (Sp_\emptyset, \emptyset, (Sp'_j \setminus Sp_k) + Sp'_k)$ together with the cocone $p : \delta \to \Sigma'_j + \Sigma'_k$ pushout of $Sig(\delta(a_1))$ and $Sig(\delta(a_2))$.

Remarks

1. *Integrate* is defined for δ where $\delta(j)$ is the system specification on which the feature $\delta(k)$ is plugged on.
2. Specification inclusion and specification difference have to be defined first
3. In previous works, we have exhibited non-conformant properties and “true” emerging properties for such kinds of specifications

Ongoing research (1)

Which architectural connectors for Genetic Regulatory Networks ?

- to redefine qualitative description of GRN in a institutional framework
- to identify adequate connectors to build systems from subsystems corresponding to biological functions
- Our aim : to be able to propose to biologists (a family of) connector(s) linking sub GRN to design a larger GRN, ensuring that a global formula is satisfied by the whole system

  (such a global formula should represent a biological knowledge which is considered as reliable by experts)

Ongoing research (2)

Dealing with non-conformance and emerging properties through refinement steps

Our aim : to study emerging properties at the right level of abstraction and to give necessary or/and sufficient conditions for the preservation of emerging properties with respect to

- vertical composition
- horizontal composition