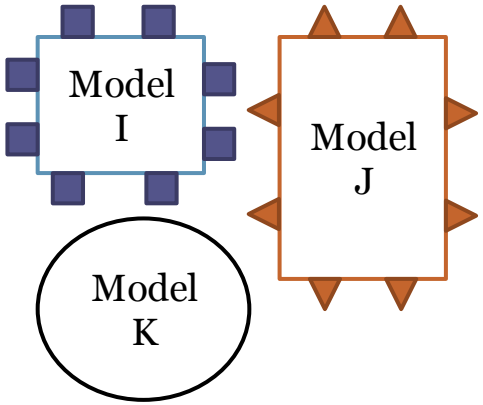
Multi-Facets Contract for Modeling and Verifying Heterogeneous Systems

Abdelkader Khouass[1;2] Christian Attiogbé[1] Mohamed Messabihi[2]

[1] University of Nantes, LS2N CNRS UMR 6004, France
christian.attiogbe@univ-nantes.fr

[2] University of Tlemcen, LRIT, Algeria
abderrahmaneabdelkader.khouass@univ-tlemcen.dz
mohamedelhabib.messabihi@univ-tlemcen.dz

The context



- **CBSE** and **the reuse of components**
- **Heterogeneous systems**
  (**Facets:** data, functionality, time, security, quality, etc.)
- Correctness of the heterogeneous systems: **modeling**

Some issues

- Components are from **different languages** and cover **different facets.**

  The composition and verification are not simple, need to be "normalized".

- **Global properties** are **heterogeneous**; need to be clearly **expressed**, **integrated** and **analyzed**.

  Need for expressive language.

- The composition of the components should preserve their local contracts.

  Respect for local requirements.

- Global properties require heterogeneous formal **analysis tools**, which generates complexity.

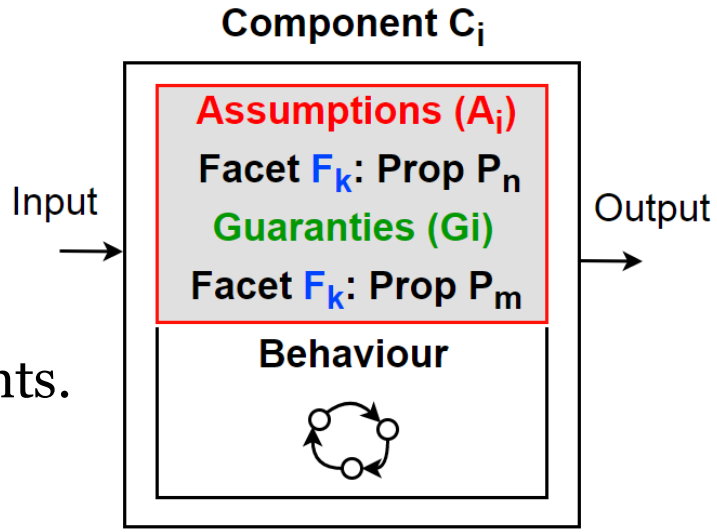
  The need of tools.

- **Focus**: top-down and practical **method**

The main concepts of our solution

Normalized components

- A normalized component is a component equipped with a **generalized contract**, acting as its interface with other components.

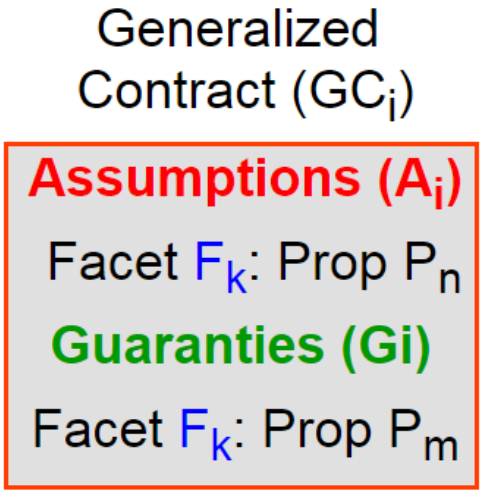


Language to express global properties

- We consider **PSL (Property Specification Language)** as an *expressive language* to express the generalized contracts.

Generalized contract

- An extension of an A/G contract.

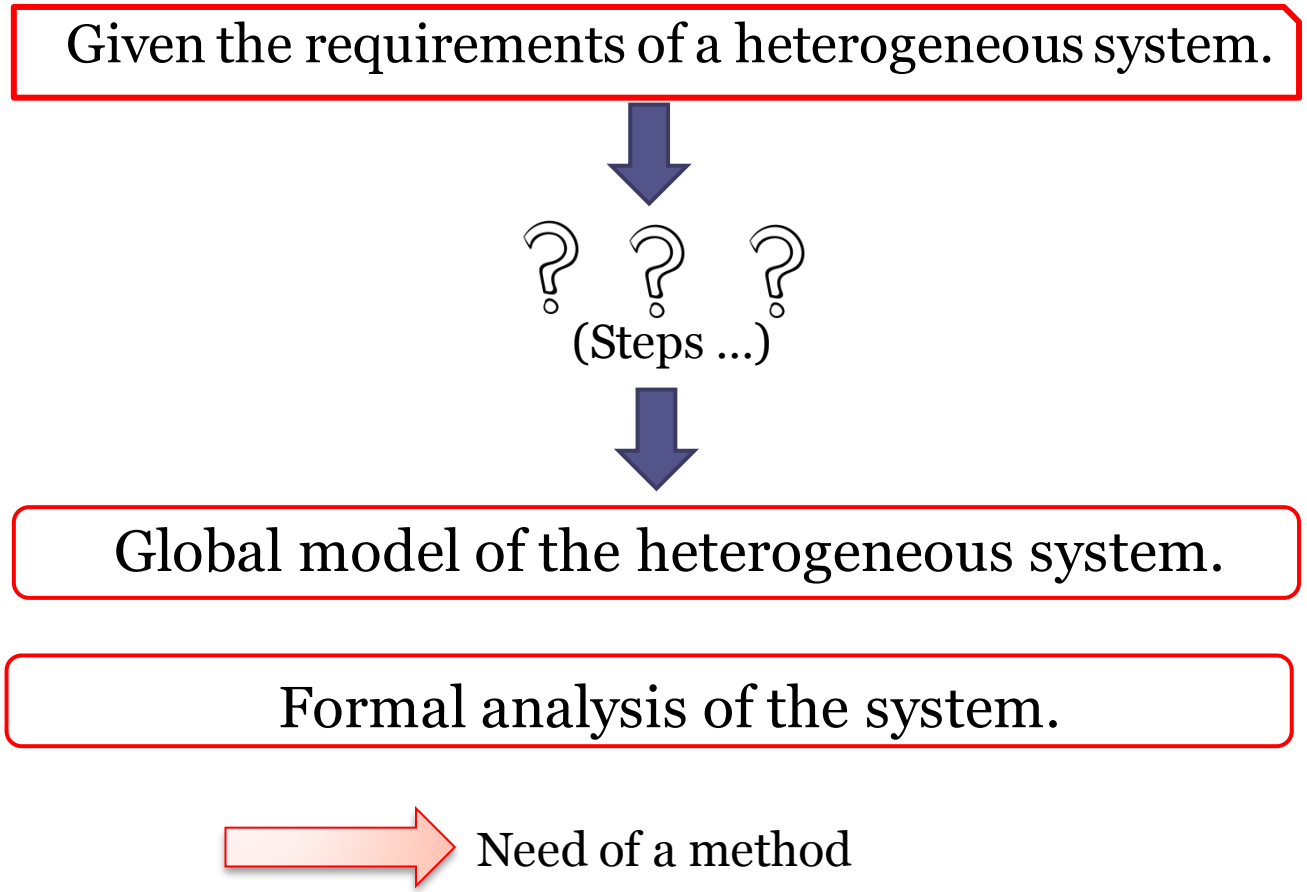


- Structured with its Assume and Guarantee parts.

- Structured according to different clearly identified **facets** (data, functionality, time, safety, quality, etc.) in its Assume or Guarantee.

- The **behaviour** is not included in the contract

Towards modeling and verification of heterogeneous systems



A method - heterogeneous system

- Composition of **normalized components** only Ci(AG,…), Cj(AG,…), …
- Decomposition of the properties with respect to the identified and agreed upon **facets** and distribution along the analysis of the assembled components.
- Reuse existing components or build needed ones.
- Manipulation of components through their **generalized contracts** (A/G).
- Weakening or strengthening of the local contracts according to the global level properties.
- Addition of a **priority** for each facet, in order to **simplify** the analysis of the global property.
- We target different analysis tools according to the facets and we have to ensure the global consistency.

Minarets method …

Structure of a heterogeneous system

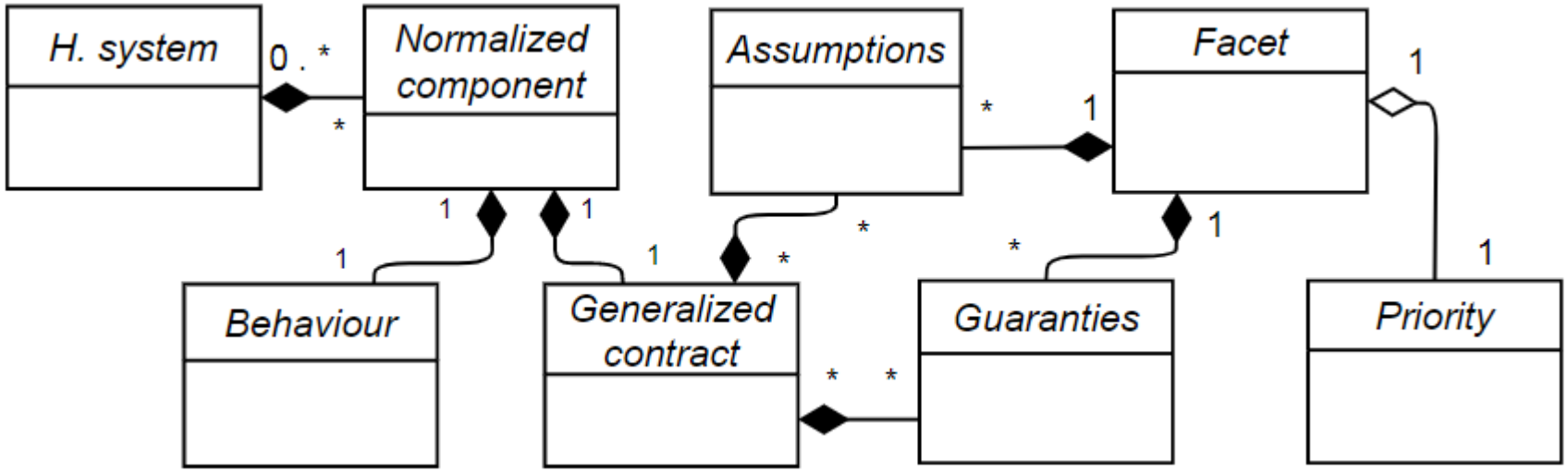


Fig. 1. Meta-model of a heterogeneous system with normalized components

Minarets Method

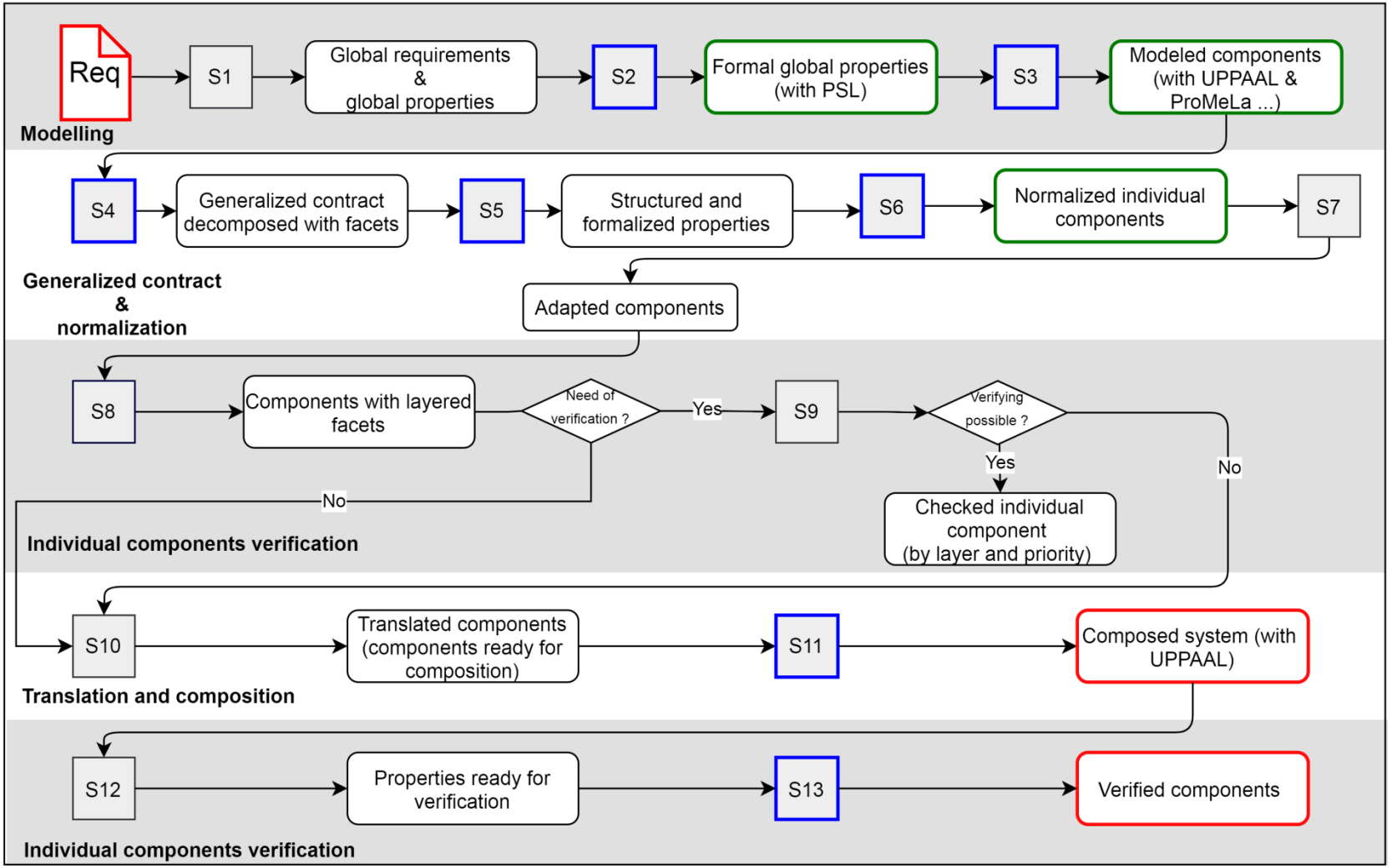


Fig. 2. The successive steps of our Minarets method

Case Study

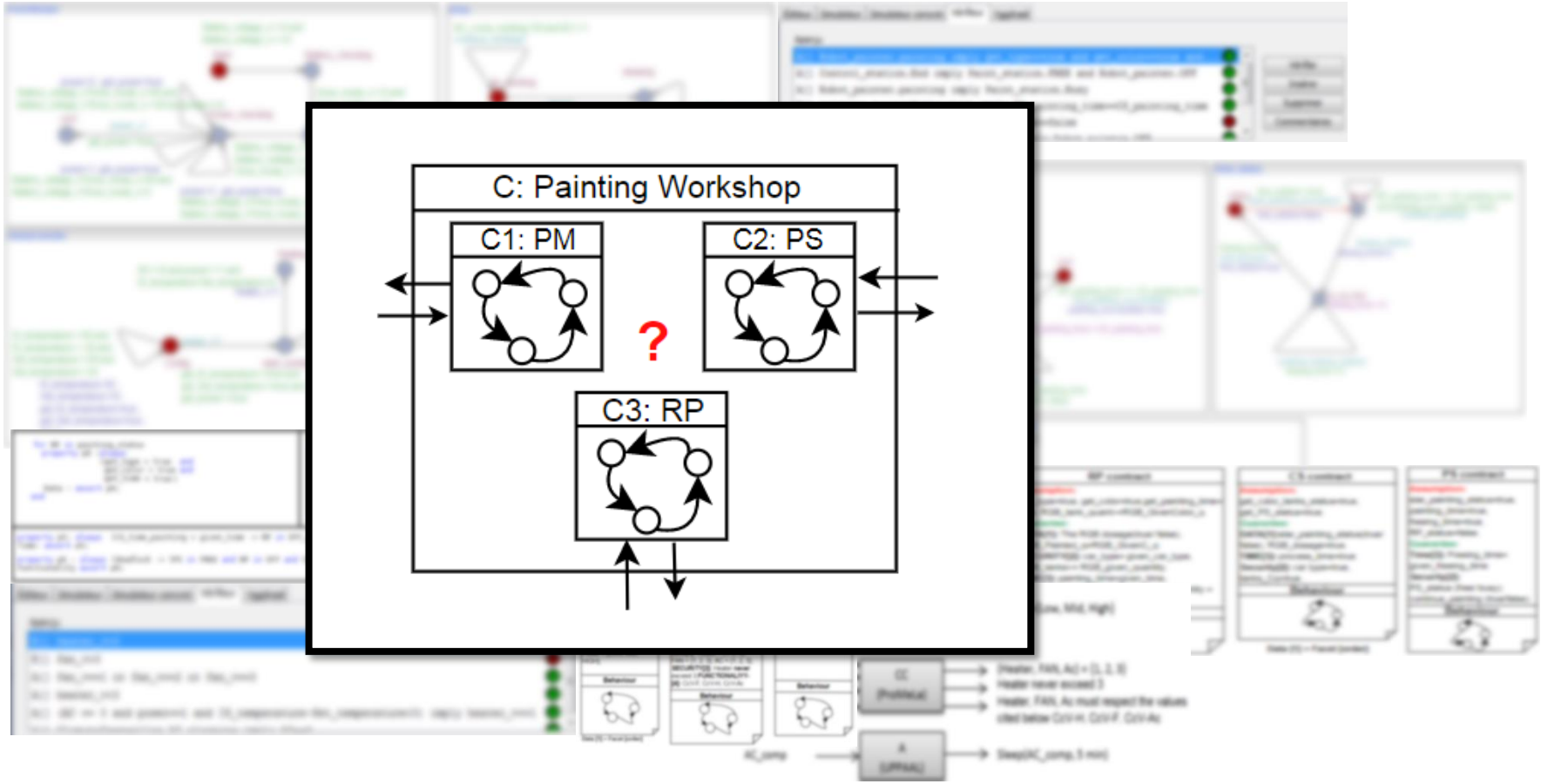


Fig. 3. Painting workshop

Step 3

- Modeling of components

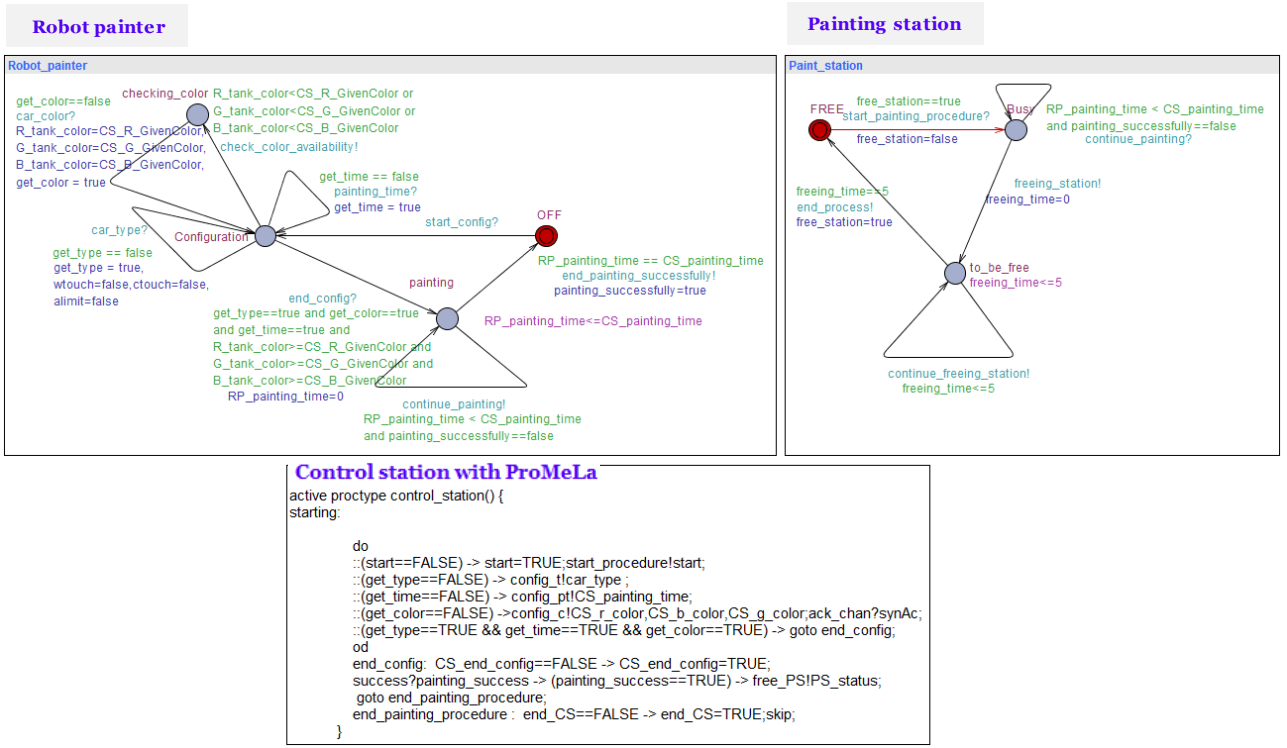


Fig. 4. Components modeled with UPPAAL and ProMeLa

Step 4

- Decomposition of the global properties with respect to the facets that we considered (Data, functionality, time, security);

DATA: The RGB dosage(true/ false), star_painting_status(true/ false).
FUNCTIONALITY:
RGB_painted_quantity = RGB_given_quantity.
TIME: painting_time=given_time, Freeing_time= given_freeing_time.
SECURITY: car_type= given_car_type, RGB_tanks>= RGB_given_quantity.

Step 5

- Structuration of the formalized properties with the PSL language

```
for RP in painting_status
  property p0 :always
              (get_type = true  and
               get_color = true and
               get_time = true )
   Data : assert p0;
end
```

```
for RP in painting_status
   property p1 :always
        (PS in Busy_status  and
         CS in end_configuration and
         R_tank_color >= CS_R_GivenColor and
         B_tank_color >= CS_B_GivenColor and
         G_tank_color >= CS_G_GivenColor)
   Security : assert p1;
end
```

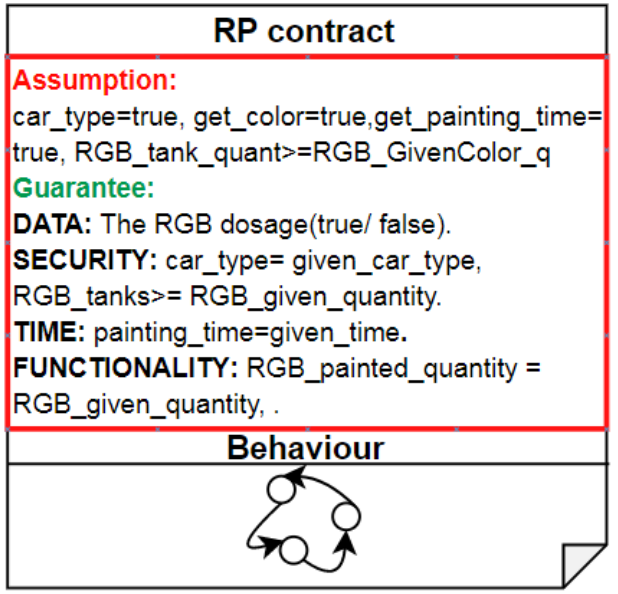
```
property p5: always  (CS_time_painting = given_time -> RP in OFF_status)
Time: assert p5;

property p6 : always (deadlock -> (PS in FREE and RP in OFF and CS in End ))
Functionality assert p6;
```

Fig. 5. Structured property with PSL

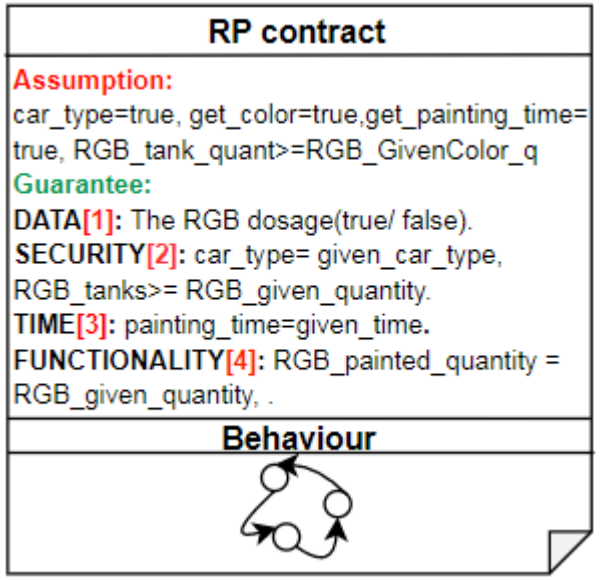
Step 6

- Normalization
- Integration of assumptions and guarantees



Step 8

- Attribution of a priority to each facet



Step 11

- Composition of the component behaviour (with UPPAAL).

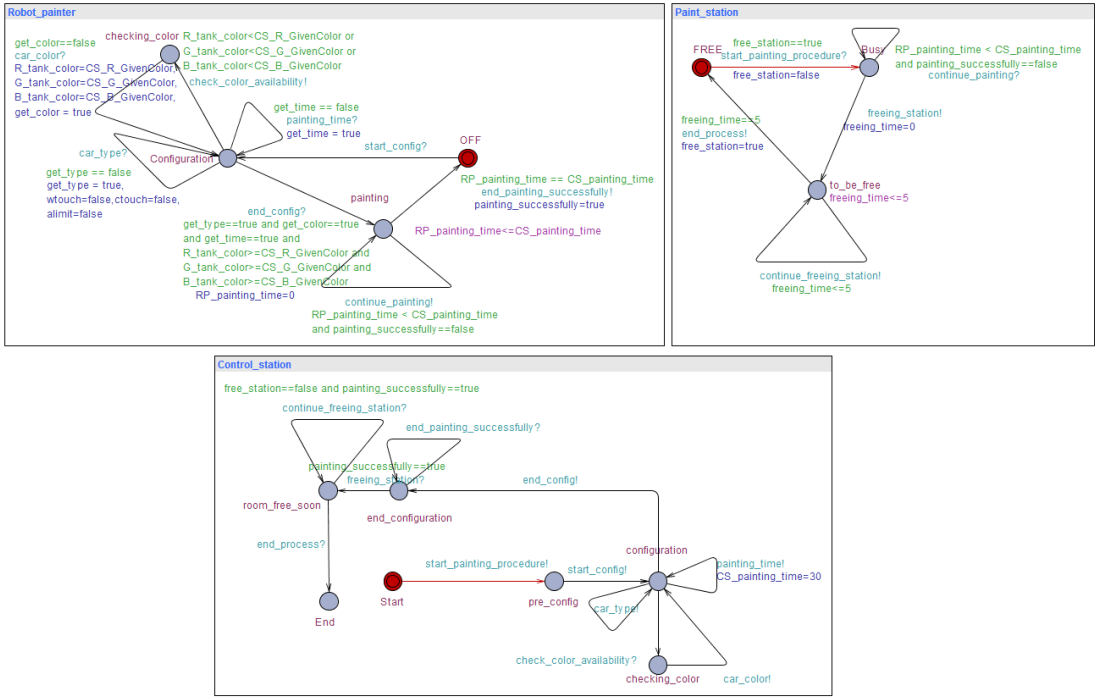


Fig. 6. The composed system after the component translation (in UPPAAL)

Assessment

- The proposed **Minarets** method solves a part of the faced issues
- More **tool** assistance is needed
- The experimentations give the opportunity to tune the method steps
- The impact of treated **facets** on interactions between various tools

Conclusion

- **Minarets** method for complex and heterogeneous systems **modeling** and **analysis**
- **Generalized contract** (the standard interfaces between components)
- Reducing the **difficulty of modeling and analysis** of heterogeneous systems composition.

Perspectives

- The study of various policies for the composition of the normalized components.
  The construction of the global property from the local properties.
- The study of the global consistency of the composed system.
- The distribution of the global property on the local components.
- Verification of the different facets written with PSL according to the verification tools.

Thank you for your attention . .