10 years of Model Federation with Openflexo : Challenges and Lessons Learned

Jean-Christophe Bach, Antoine Beugnard, Joël Champeau,
Fabien Dagnat, **Sylvain Guérin**, **Salvador Martínez**

MODELS'2024, Linz, September 26th

What is OpenFlexo?

- A *model federation* framework
- A scientific experimental platform
- A successful but defunct company
- Many projects, academia and industry collaborations.

- Complex systems : multiple **points of view**
- Complex systems : heterogeneity (tools, actors, formalism)

- Complex systems : multiple **points of view**
- Complex systems : heterogeneity (tools, actors, formalism)
- We were (and remain) modeling enthusiasts

- **C1: Preserving existing practices**

- Preserve data and life cycles
- Do not break what already works (existing tools and processes)
- Do not require to replace tools or re-train users to new tooling

- C1: Preserving existing practices
- **C2: Seeing everything as a model**

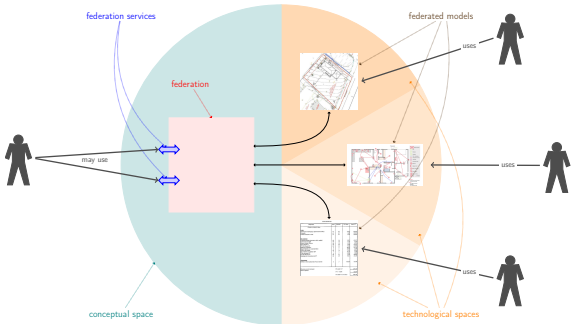
- Any source of information may be considered as a model
- The "most appropriate" formalism
- Requires a way to interpret (or reinterpret) data contextually

- C1: Preserving existing practices
- C2: Seeing everything as a model
- **C3: Conceptualizing and organizing**

- Find a way to reference elements source and define links
- Make possible for new concepts to emerge

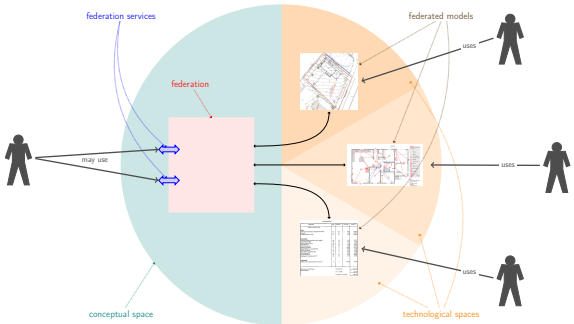
- C1: Preserving existing practices
- C2: Seeing everything as a model
- C3: Conceptualizing and organizing
- **C4: Mastering evolution**

- Detect (and eventually react to) changes
- Deal with/Tolerating inconsistencies



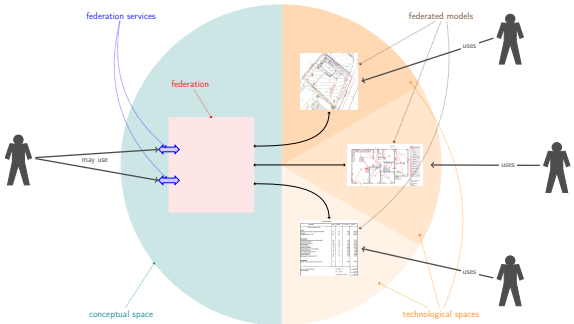
- Federated models
- Technological space

Federated models remain in their original **technological spaces**, users keep their practices



- Federated models
- Technological space
- Correspondence
- Conceptual space
- Federation

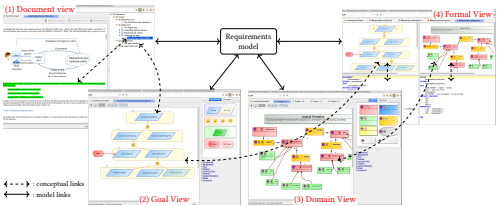
Their **correspondences** are reified within a model of the **conceptual space** called a **federation**



- Federated models
- Technological space
- Correspondence
- Conceptual space
- Federation

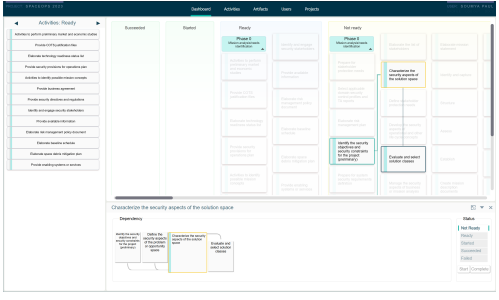
A **federation** is connected to a set of **federated models** and manages their **correspondences**. A **federation** can provide new services

- **UC1 : Formose**



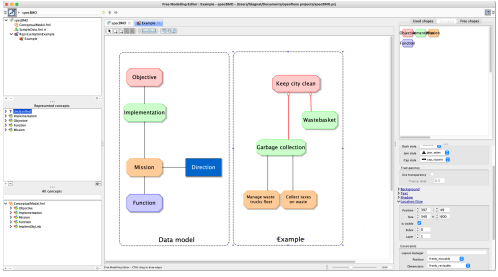
- A model-based requirement engineering tool
- Research prototype with academic and industrial partners

- UC1 : Formose
- **UC2 : SSE4Space**



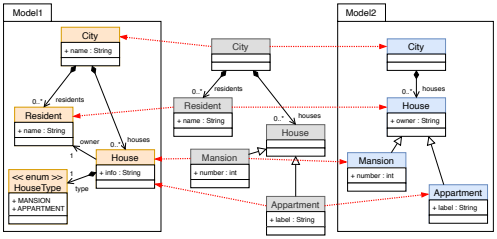
- Managing the security concern in satellite development
- An industrial project developed with Telindus for European Space Agency

- UC1 : Formose
- UC2 : SSE4Space
- **UC3 : Free Modeling Editor**



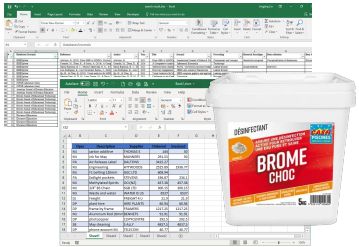
- Tool to co-construct a model and its metamodel
- Centered around an example diagram that may not conform to a metamodel

- UC1 : Formose
- UC2 : SSE4Space
- UC3 : Free Modeling Editor
- **UC4 : Model Mapping**



- Tool to build model mapping between various DSML
- Mapping, edition and consistency management

- UC1 : Formose
- UC2 : SSE4Space
- UC3 : Free Modeling Editor
- UC4 : Model Mapping
- **UC5 : PDF Labels**



- Tool to audit and update internal databases from information contained in PDF labels
- Federate informations using business-specific rules

- **L1: Separate business from technique**

- Focus to the essence of things and forget about technical details
- When modelling focus on business concepts, forget your own modeling habits
- A space for technical aspects, a space for domain-specific semantics

- L1: Separate business from technique
- **L2: Tool maintenance and dissemination**

- Flexibility comes at a price
- Keeping up to the technical stack is hard (e.g., Java)
- Dissemination is not easy

- L1: Separate business from technique
- L2: Tool maintenance and dissemination
- **L3: Need for many concrete syntax**

- The most appropriate langage for the right person at the right time :
  - Graphical form-based is ideal for small projects, textual is better for bigger and more complex projects
  - Diagramming is well suited at the early stages, textual is more appropriate for developers
  - ...

- L1: Separate business from technique
- L2: Tool maintenance and dissemination
- L3: Need for many concrete syntax
- **L4: Federation requires organization**

...as they grow rapidly !

- Designation problems are solved with technological adapters
- Federation models are models too and re-use model organization techniques

- L1: Separate business from technique
- L2: Tool maintenance and dissemination
- L3: Need for many concrete syntax
- L4: Federation requires organization
- **L5: Modeling requires flexibility**

- Free modeling help capture domains
- Interpretation (vs code generation) improves productivity in early stages

- L1: Separate business from technique
- L2: Tool maintenance and dissemination
- L3: Need for many concrete syntax
- L4: Federation requires organization
- L5: Modeling requires flexibility
- **L6: Modeling needs model capitalization**

- Re-use is key (especially for large scale modeling)
- Federation eases re-use


Openflexo

- Collaboration
- Free Modeling
- Maintainability
- Digital Twins

- Collaboration
- Free Modeling
- Maintainability
- Digital Twins

- Collaboration
- Free Modeling
- Maintainability
- Digital Twins

- Want to..
- Collaborate?
- Contribute?
- Contact us!

Openflexo

- Collaboration
- Free Modeling
- Maintainability
- Digital Twins

Lab-STICC

P4S

- Want to..
- Collaborate?
- Contribute?
- Contact us!