«Expressing KAOS Goal Refinement Patterns with Event-B»

Abderrahman Matoussi, Frédéric Gervais and Régine Laleau
LACL , Université Paris-Est
{abderrahman.matoussi,frederic.gervais,laleau}@univ-paris12.fr

- Rodin User and Developer Workshop-
University of Southampton

July 17th , 2009

Outline

- Motivation
- Overview of KAOS
- The proposed approach
- Summary and perspectives

Introduction

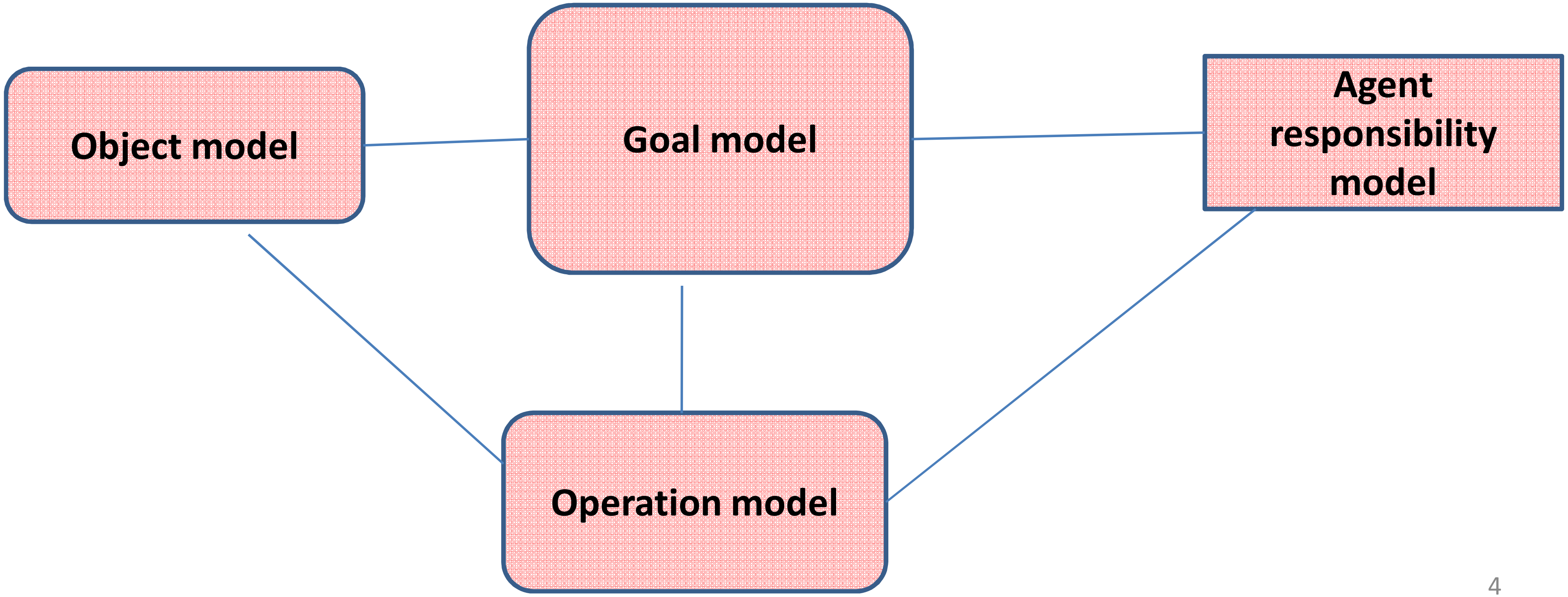
- A large gap between textual or semi-formal requirements and the initial formal specification.
- The validation of the initial formal specification is difficult .
  - ➢Inability to understand the formal model (for the customer).
  - ➢Link them with initial requirements (for designers).
- Up to now, the development process associated with formal methods begins only at the specification level.

Solution

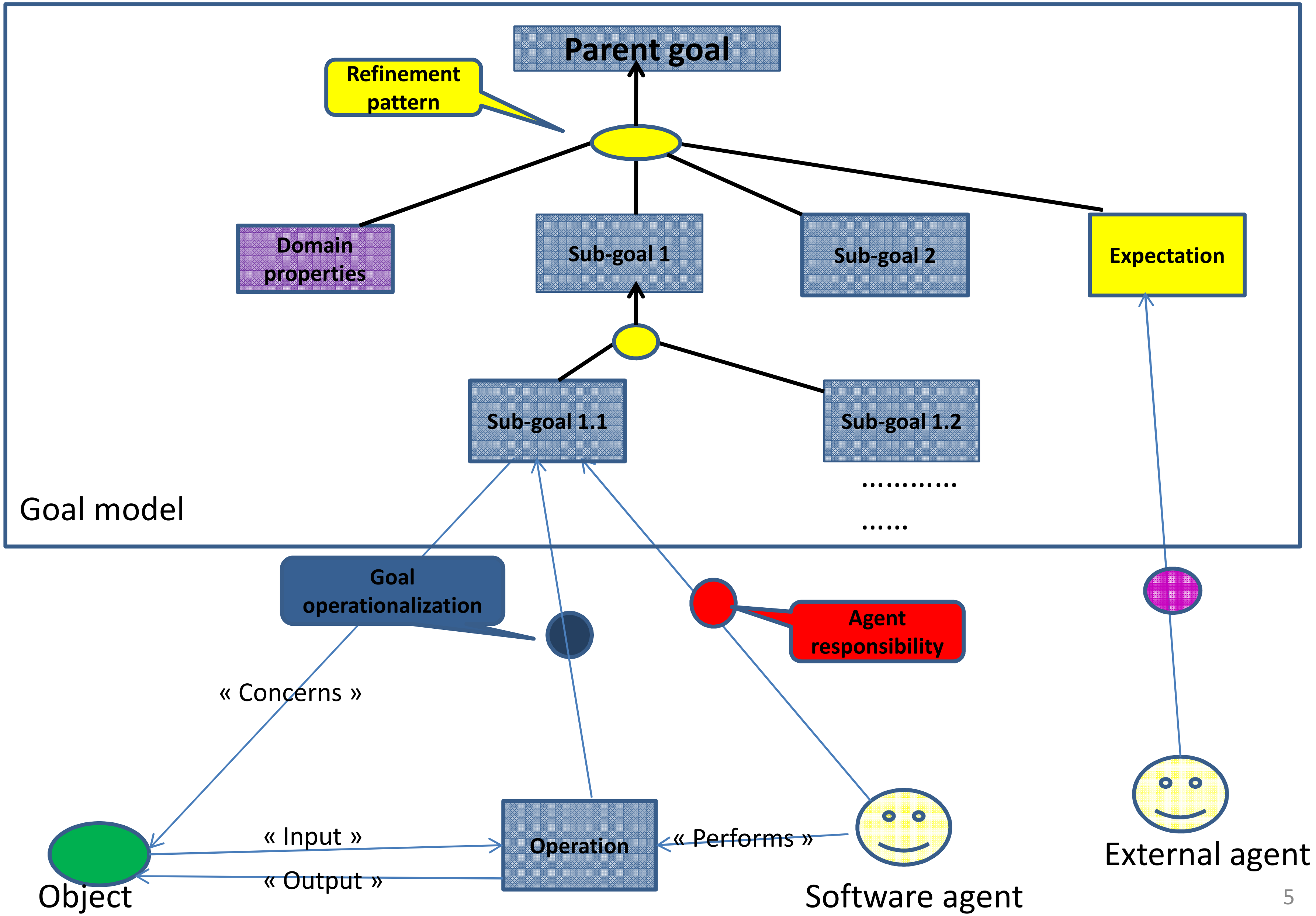
- Our objective is to include requirements analysis within this process.
  - ➢ Coupling KAOS and the Event-B method.

Overview of KAOS

- A methodology to implement goal-based reasoning.
- A goal defines an objective the system should meet, usually through the cooperation of multiple agents such as devices or humans.



Overview of KAOS



Bad news about KAOS

- The KAOS method stops at the requirements phase.
- It doesn't address the other software development levels.

➢ A serious shortcoming:

  - Obliging designers to use another method for developing their systems.
  - Difficulty to validate specifications with regard to requirements.

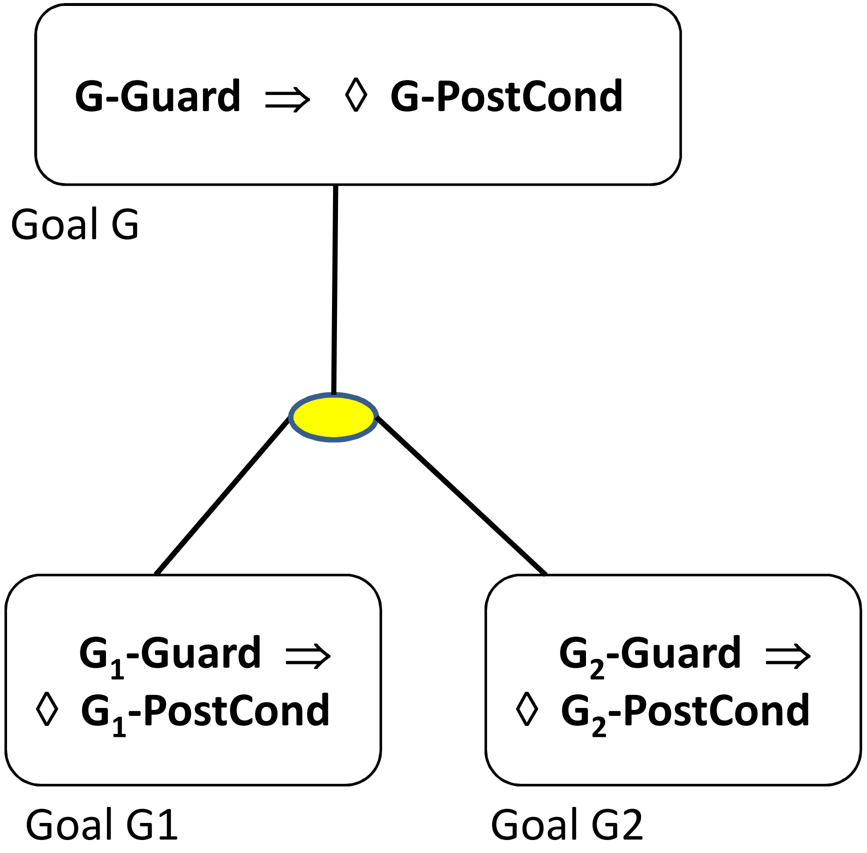
Good news about KAOS

- KAOS can be extended with an extra step of formality. (Contrary to other requirements methods such as i*).
- Similarity and complementarity between KAOS and Event-B.
  - Employing first-order predicate logic.
  - The notion of refinement (constructive approach).
  - The ability to model both the system and its environment (contrary to the classical B).

- ➢ Facilitating the correspondence between KAOS and Event-B.
- ➢ Filling the gap between requirements and the later phases of development.

An overview of the approach

"Staying at the same abstraction level"



G-Guard $\Rightarrow$ ◊ G-PostCond

Goal G

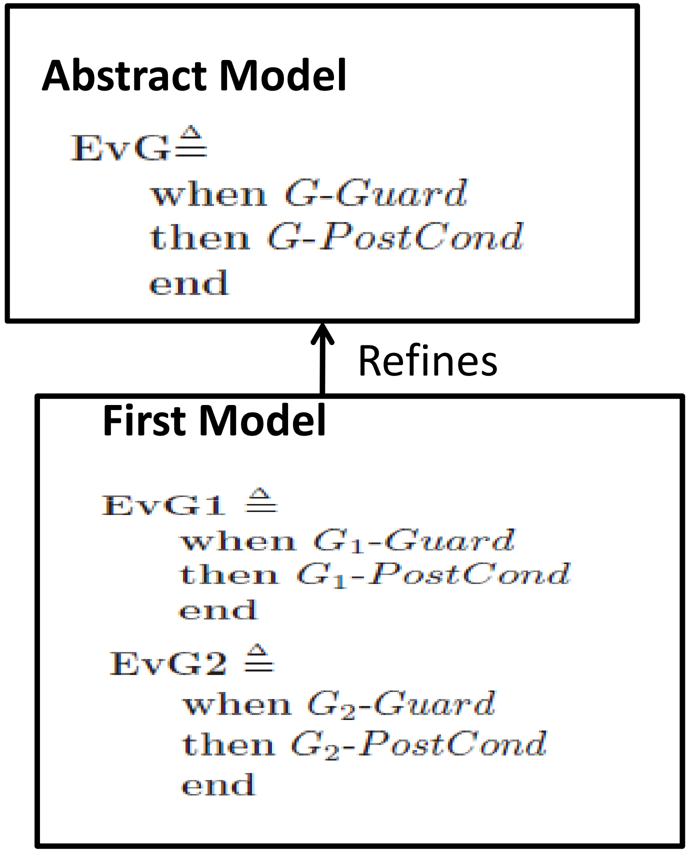
G_1-Guard $\Rightarrow$ ◊ G_1-PostCond

Goal G1

G_2-Guard $\Rightarrow$ ◊ G_2-PostCond

Goal G2

KAOS goal model



Abstract Model

```
EvG ≙
    when G-Guard
    then G-PostCond
    end
```

Refines

First Model

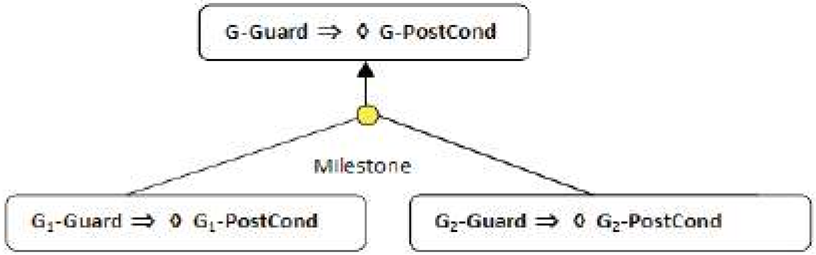
```
EvG1 ≙
    when G₁-Guard
    then G₁-PostCond
    end
EvG2 ≙
    when G₂-Guard
    then G₂-PostCond
    end
```

Event-B representation of KAOS goal model

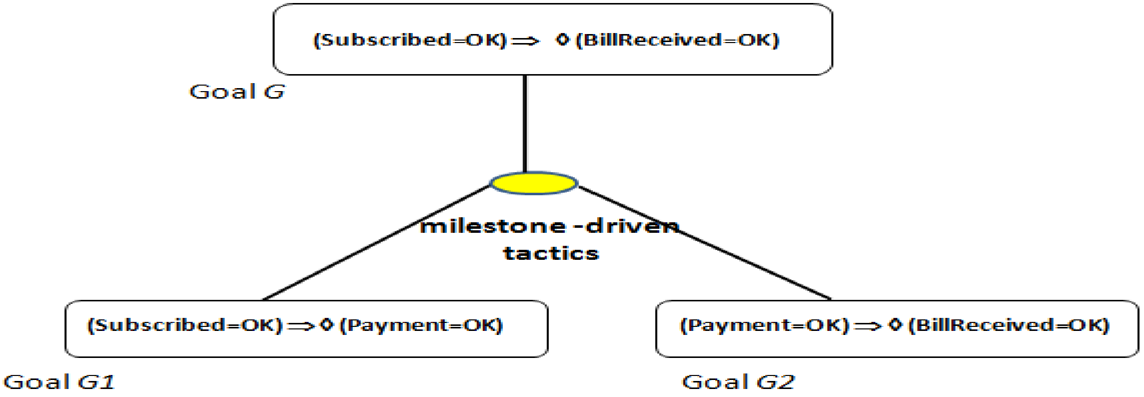
- ➢ The Event-B expression of the KAOS goal model allows us to give it a precise semantics.

Expressing the milestone-driven goal refinement pattern in Event-B

Description of the KAOS pattern:


G-Guard ⇒ ◊ G-PostCond
Milestone
G_1-Guard ⇒ ◊ G_1-PostCond
G_2-Guard ⇒ ◊ G_2-PostCond

Example: Workshop Subscription


(Subscribed=OK) ⇒ ◊ (BillReceived=OK)
Goal *G*
milestone -driven tactics
(Subscribed=OK) ⇒ ◊ (Payment=OK)
Goal *G1*
(Payment=OK) ⇒ ◊ (BillReceived=OK)
Goal *G2*

Expressing the milestone-driven goal refinement pattern in Event-B

Formal semantics of the pattern: $(EvG1 \,;\, EvG2)$ Refines EvG

$$\begin{tikzcd}
S \arrow[rr, "EvG"] & & S \\
T \arrow[u, "r"'] \arrow[r, "EvG_1"'] & T \arrow[r, "EvG_2"'] & T \arrow[u, "r"]
\end{tikzcd}$$

Forward Simulation : $r^{-1}; EvG1; EvG2 \subseteq EvG; r^{-1}$

Expressing the milestone-driven goal refinement pattern in Event-B

Proof obligations identification:

$G_1\text{-}PostCond \Rightarrow G_2\text{-}Guard \qquad \textbf{(PO1)}$

”Ordering constraint”.

$$\boxed{\begin{array}{l} I(v) \\ J(v,w) \\ G_1\text{-}Guard(w) \\ G_1\text{-}PostCond(w,t) \\ \vdash \\ \exists w' \;.\; G_2\text{-}PostCond(t,w') \end{array}} \qquad \textbf{(PO2)}$$

”Feasibility refinement”.

$G_1\text{-}Guard \Rightarrow G\text{-}Guard \qquad \textbf{(PO3)}$

”Guard Strengthening”.

$G_2\text{-}PostCond \Rightarrow G\text{-}PostCond \qquad \textbf{(PO4)}$

”Correct Refinement”.

Summary

- Formal design patterns or proof-based design patterns is very useful and explores the fact that the Event-B method provides a framework for developing generic models of systems.
- Proving the requirements model and to establish formal links between this model and the specification of a system.
- Balancing the tradeoff between complexity of rigid formality (Event-B method) and expressiveness of semi-formal approaches (KAOS).

Perspectives

- Applying the approach on a number of case studies in order to support non-functional goals.
- Extending the approach on the other goals patterns.
- Defining a “formal mapping” between this Event-B representation and the initial formal specification.
- Developing a connector between KAOS toolset (not OBJECTIVER) and the RODIN open platform.

Thanks for your attention

Any question ?