

Facing Uncertainty in Web Service Compositions

Professor Germán H. Alférez, Ph.D.

School of Engineering and Technology,
Universidad de Montemorelos, Mexico



UNIVERSIDAD
POLITECNICA
DE VALENCIA



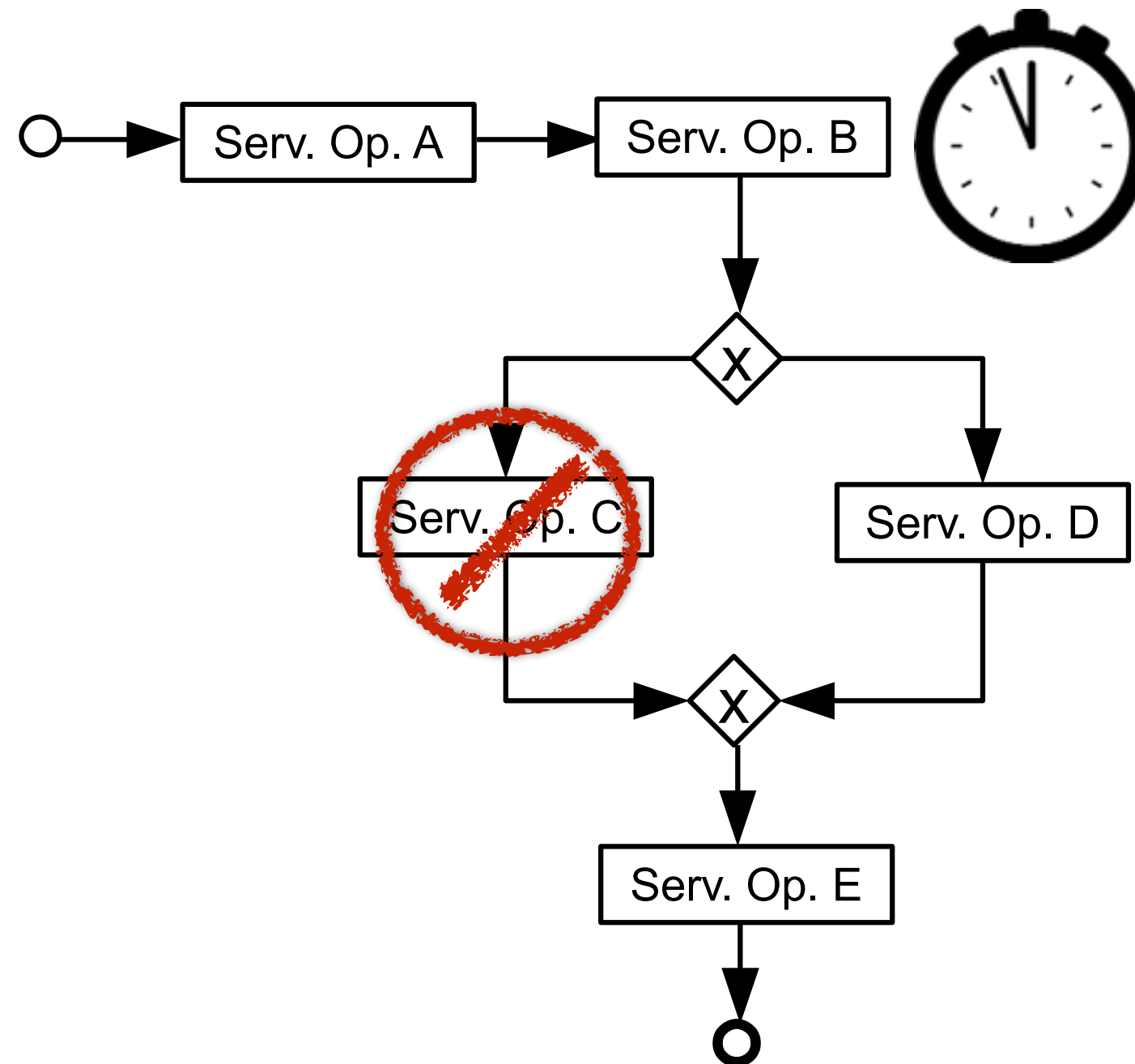
Software also runs in complex & heterogeneous computing infrastructures.

Translate the ideas of **adaptation** in the natural world to software.



Adaptability is emerging as a **necessary capability** of **highly-dynamic systems** (Hong et al., 2009).

Systems based on **Web service compositions**



Dynamic adaptations: the service composition **self-adjusts at runtime** to do the following:

- Keep the **quality** of the service composition
- Offer **extra functionality** depending on the context
- **Protect** the system
- Make the system **more usable**



Related work on **dynamic adaptation** of service compositions has traditionally tended to focus on:

**1. Variability
constructs at the
language level**

2. Brokers

I. Variability constructs at the language level

It can **hinder reasoning** about adaptations with **complex** and **error-prone scripts** (Fleurey and Solberg, 2009)

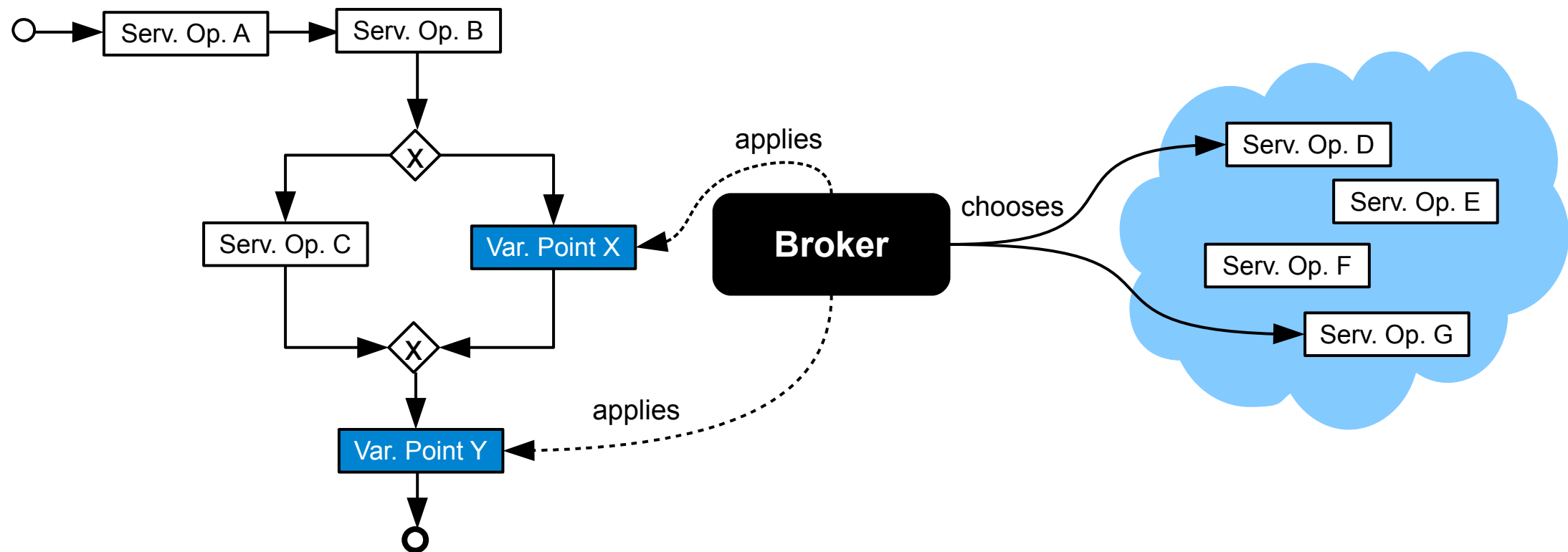
```
<vxbpel:VariationPoint name="VPService1">
  <vxbpel:Variants>
    <vxbpel:Variant name="Service1A">
      <vxbpel:VPBpelCode>
        <invoke ... partnerLink="service1a"/>
      </vxbpel:VPBpelCode>
    </vxbpel:Variant>
    <vxbpel:Variant name="Service1B">
      <vxbpel:VPBpelCode>
        <invoke ... partnerLink="service1b"/>
      </vxbpel:VPBpelCode>
    </vxbpel:Variant>
  </vxbpel:Variants>
</vxbpel:VariationPoint>
```

Need to manage adjustments at a **higher level of abstraction.**

Koning et al., 2009

2. Brokers

Most research works **lack support** for analyzing the inherent **variability** of dynamic adaptation at **design time**



Need to manage **variability** at **design time** and at **runtime**.

Tendency: dynamic adaptation of service compositions in the **closed-world**. However, **the world is increasingly open!**

Closed World

Stable contexts (the context changes *slowly*)

Anticipated changes (*foreseen* context events)

Open World

Dynamic contexts (focused on *ubiquitous* computing infrastructures. The context changes *rapidly*)

Unanticipated changes (*unknown* context events)

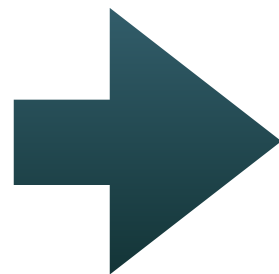
Need to manage dynamic adjustments in the unpredictable **open world**.

Contributions



Need to manage adjustments at a **higher level of abstraction.**

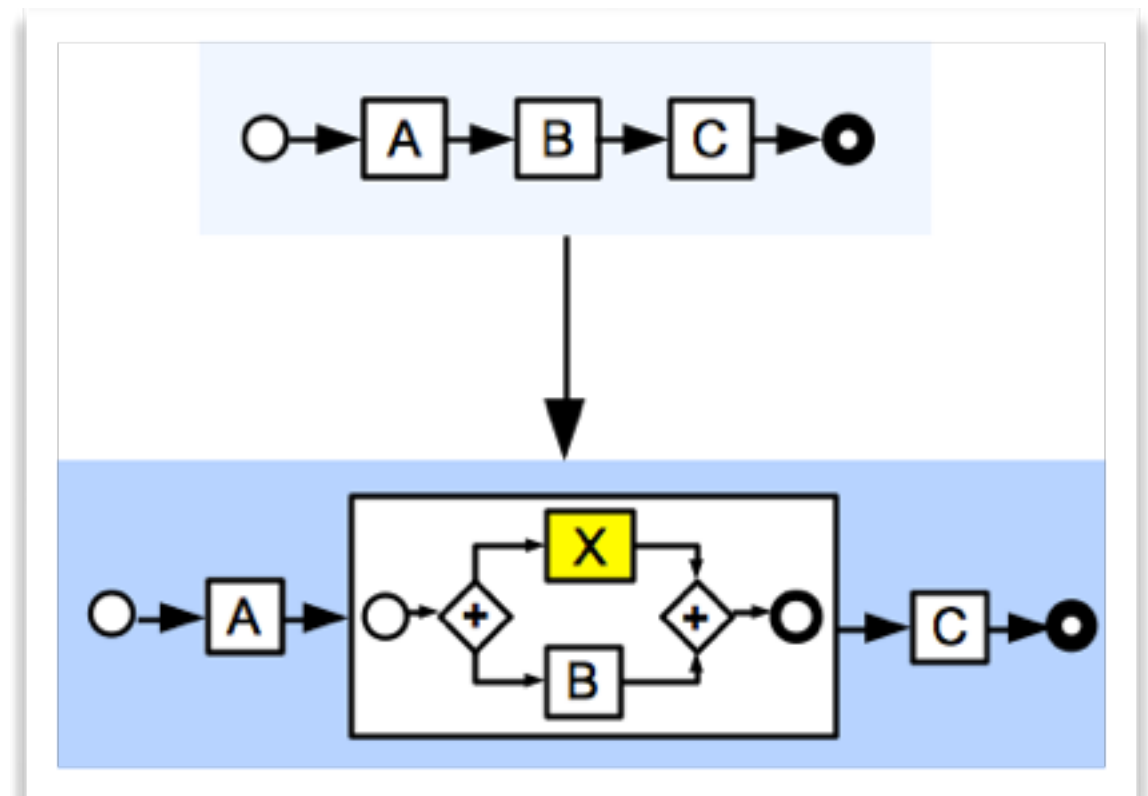
Need to manage dynamic adjustments in the unpredictable **open world.**



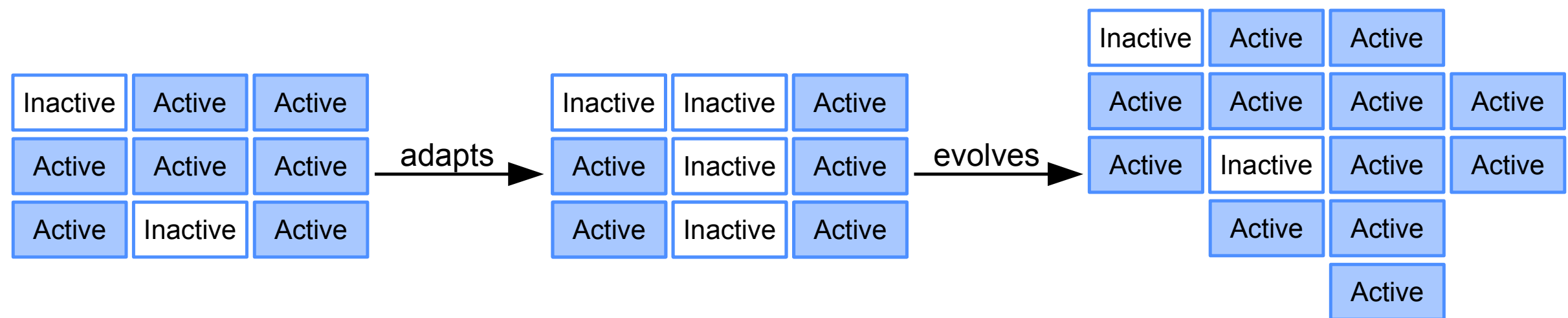
An approach to manage **some situations of uncertainty in the open world** by **dynamically evolving service compositions through models at runtime.**

An approach to manage **some situations of uncertainty in the open world** by dynamically evolving service compositions through **models at runtime**.

Dynamic evolution: “The process of moving the service composition to a new version (which cannot be supported by predefined dynamic adaptations) in order to manage **unknown context events at runtime**.” (Alférez and Pelechano, **MODELS** 2012)



An approach to manage **some situations of uncertainty in the open world** by dynamically evolving service compositions through **models at runtime**.



Dynamic adaptations are carried out to make **punctual** **Dynamic evolutions** imply a **gradual structural or changes** in the service composition with “**known**” adaptation **architectural growth into a better state** in order to face policies. Dynamic adaptations **face particular “known” events in uncertainty** in the open world (Alférez and Pelechano, **MODELS the closed world** (Alférez and Pelechano, **SPLC** 2011; Alférez et al., **JSS Elsevier** 2014). **ICWS** 2013).

An approach to manage **some situations** of uncertainty **in the open world** by **dynamically evolving service compositions** through **models at runtime**.

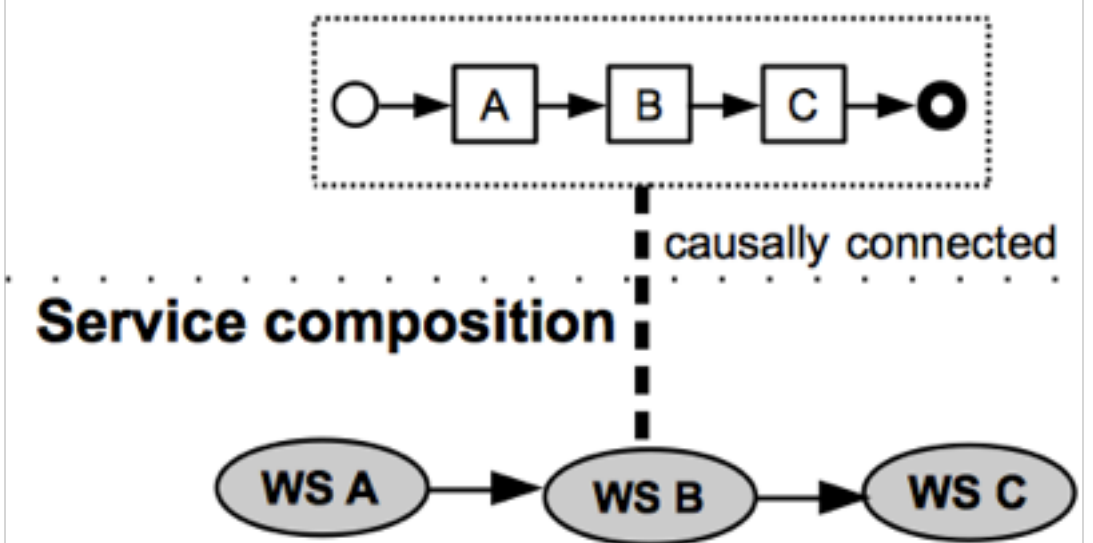
In the open world, **uncertainty** is caused by how the service composition should deal with **unknown context events**.



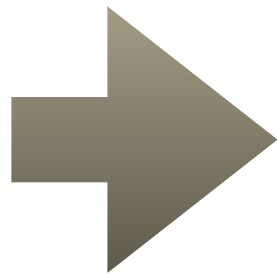
An approach to manage **some situations of uncertainty in the open world** by **dynamically evolving service compositions** through **models at runtime**.

Models at runtime: “Causally connected self-representations of the associated system that emphasize the structure, behavior, or goals of the system from a problem space perspective” (Blair, 2009).

Model at runtime



Need to manage
variability at **design time**
and at **runtime**.



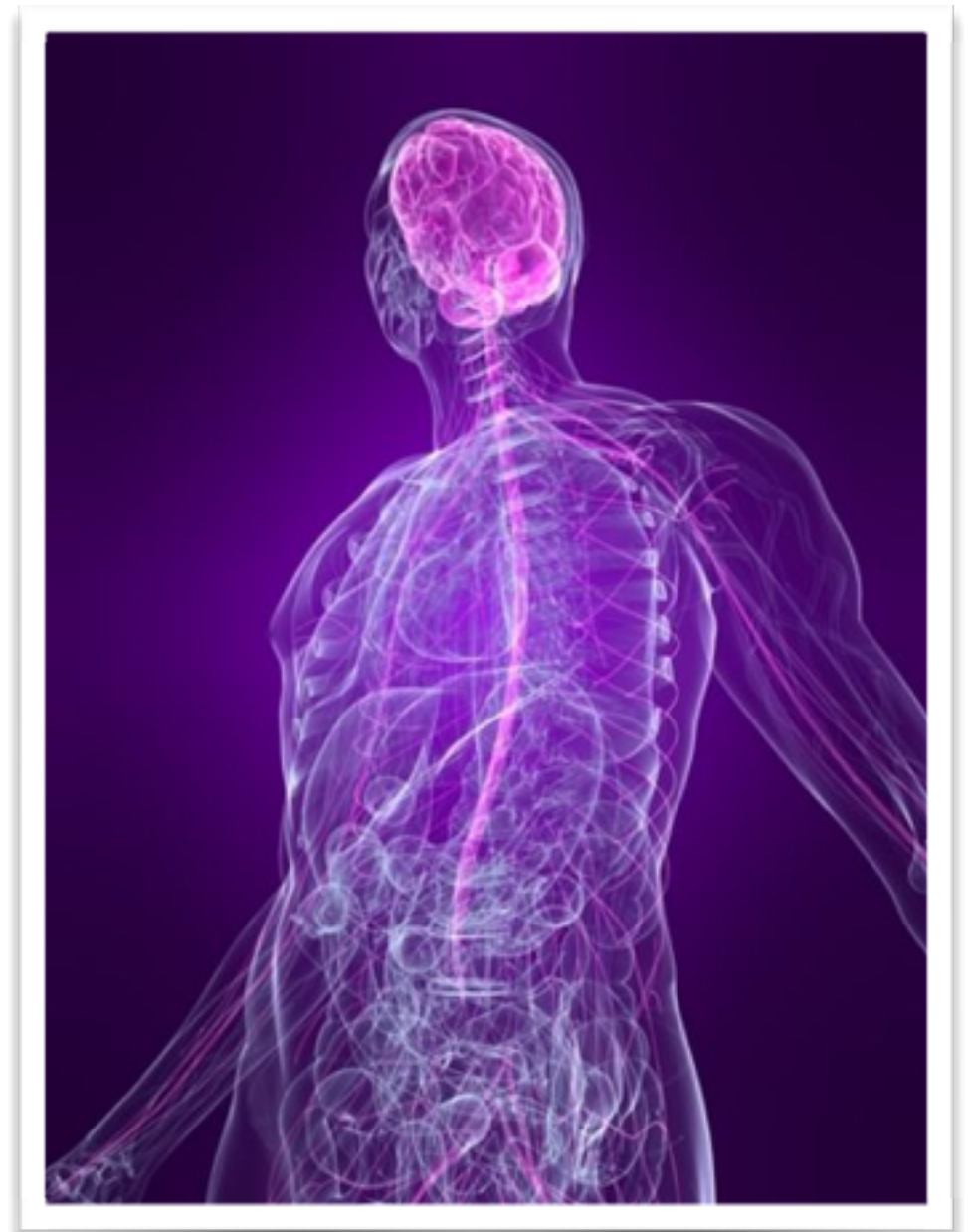
A tool-supported
software engineering
approach for the
development of
context-aware service
compositions from
design time to
execution.

A **tool-supported software engineering approach** for the **development** of context-aware service compositions from **design time** to **execution**.

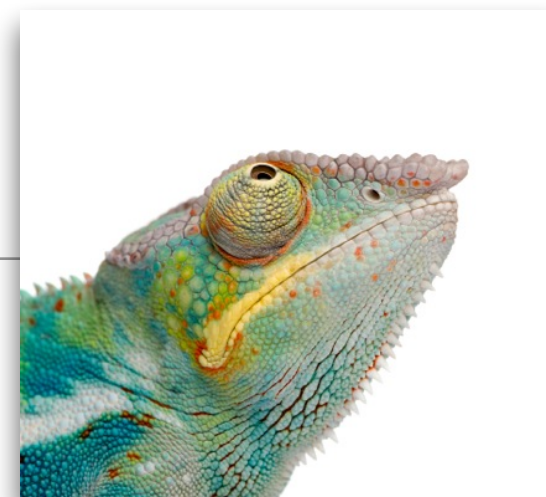
Design Time: Model-Driven Engineering

Runtime: Autonomic Computing

- AC is an initiative proposed by IBM.
- **Goal:** to develop computer systems with **self-management** capabilities.



Solution



Modeling to Face Uncertainty in the Open World

How to **preserve** expected **requirements** when the service composition faces **unknown context events** in the **open world**?

Tactics are **abstract last-resort surviving actions** to **preserve** the **requirements** that can be **negatively impacted** by **unknown context events** (Alférez and Pelechano, **MODELS** 2012).

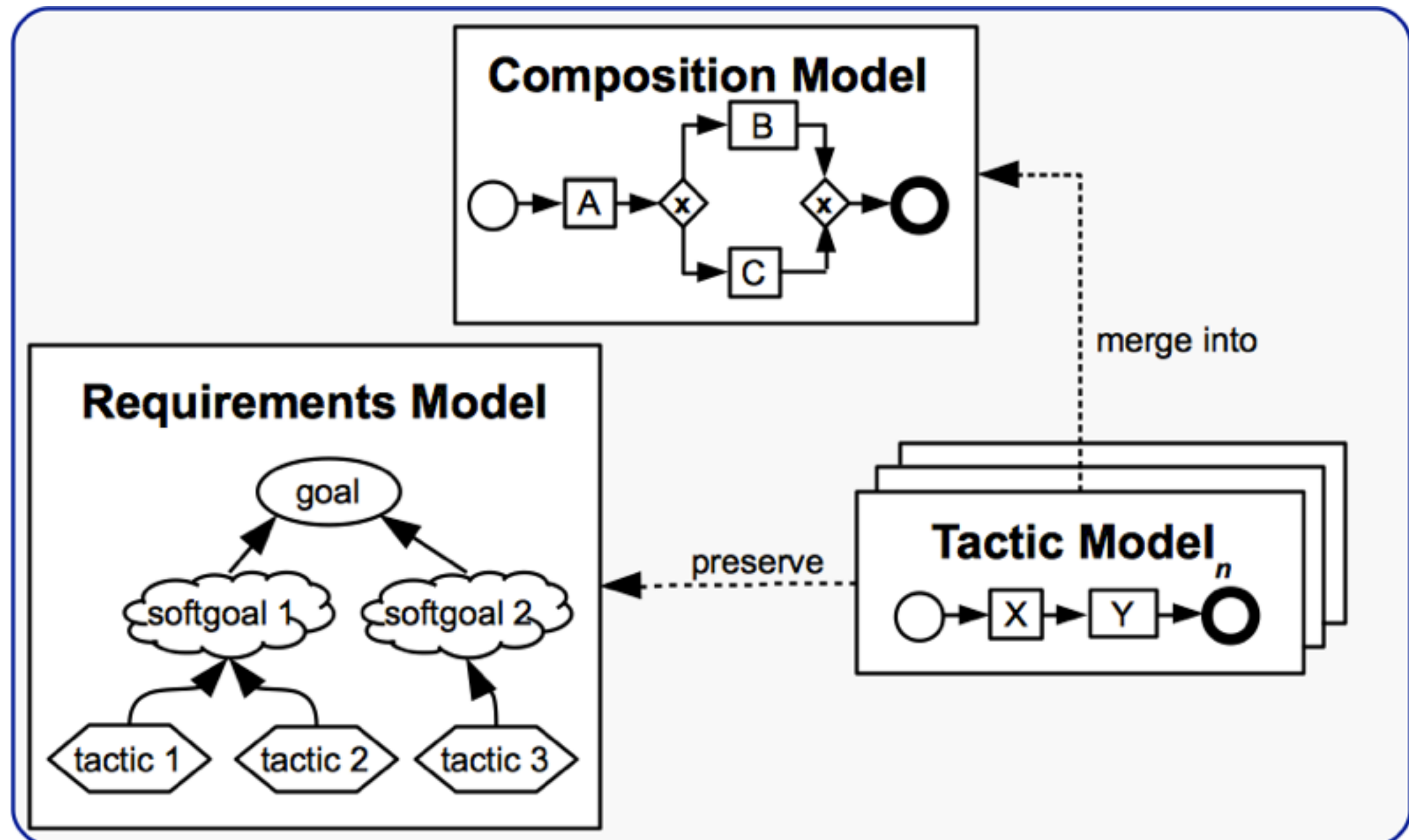
Tactics try to **reduce** the impact of unknown context events in the open world.



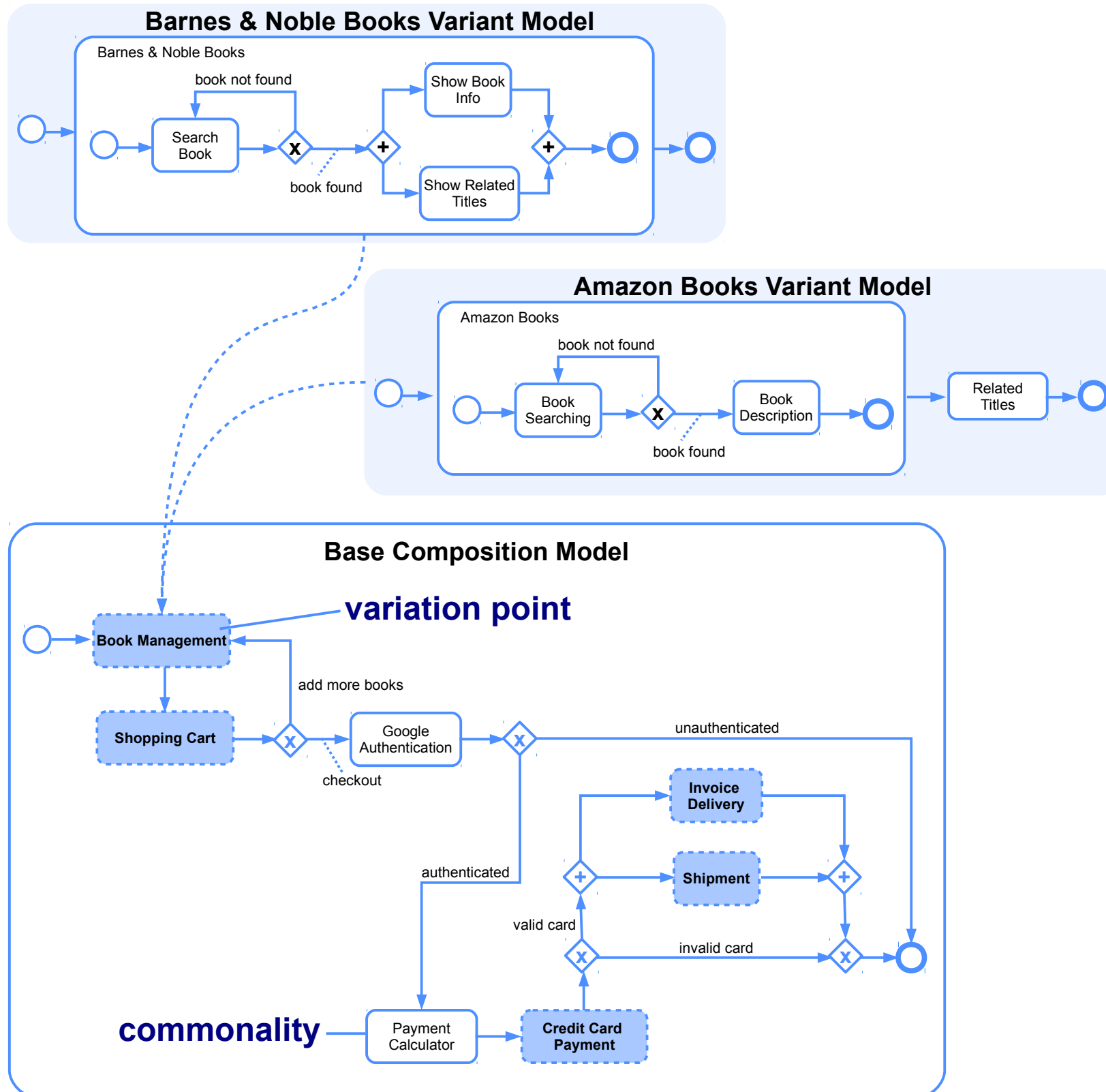
Known Unknowns

- **Goal:** To Win.
- **Unknown or unforeseen events:** Surprise assaults.
- **What to do?** Choose among a set of tactics to reach the goal - to scape vs. to do a frontal attack.
 - **Tactics** are **known beforehand**, but soldiers **do not know** to which specific arising **unknown context events** they will be applied.

Pieces of knowledge during execution to achieve dynamic evolution of service compositions.



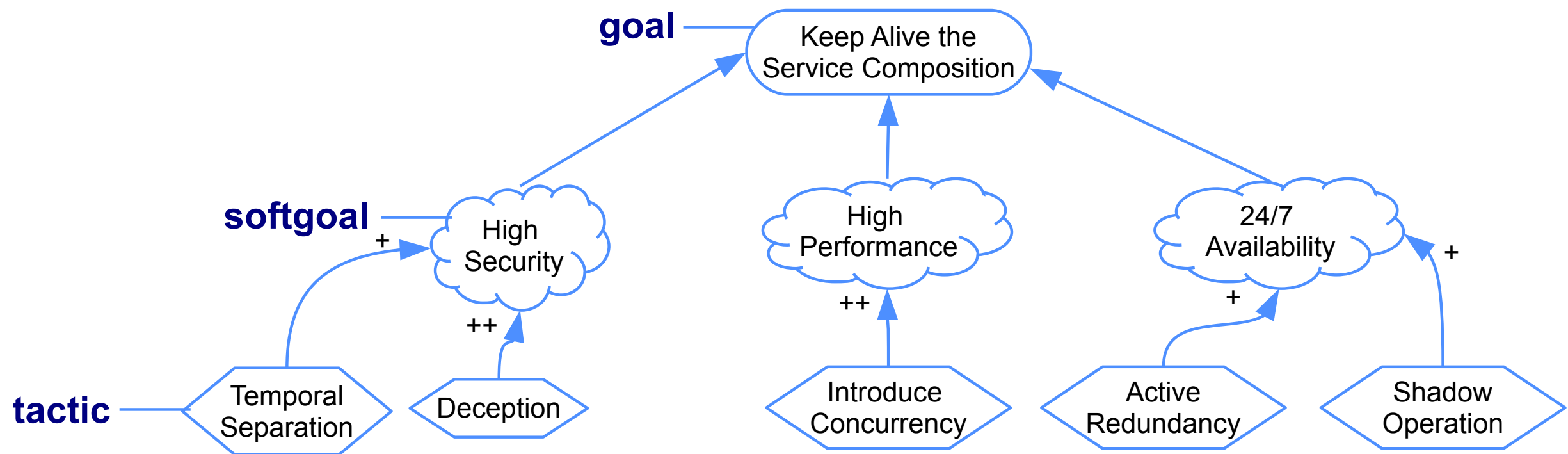
Composition Model



Extended
Business Process
Model and Notation
(BPMN)

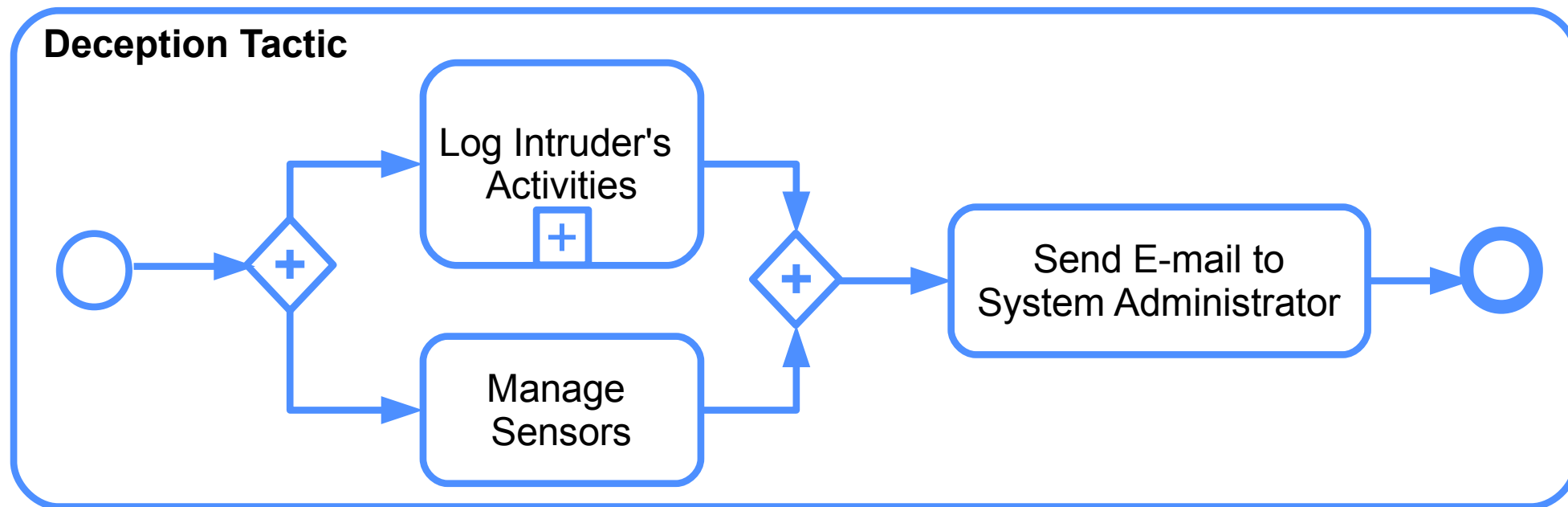
Solution - Modeling to Face Uncertainty in the Open World

Requirements Model



Goal Model (Liu and Yu, 2004; Yu, 2009)

Tactic Models



Tactic models express the tactical functionality to be triggered on the service composition to preserve requirements.

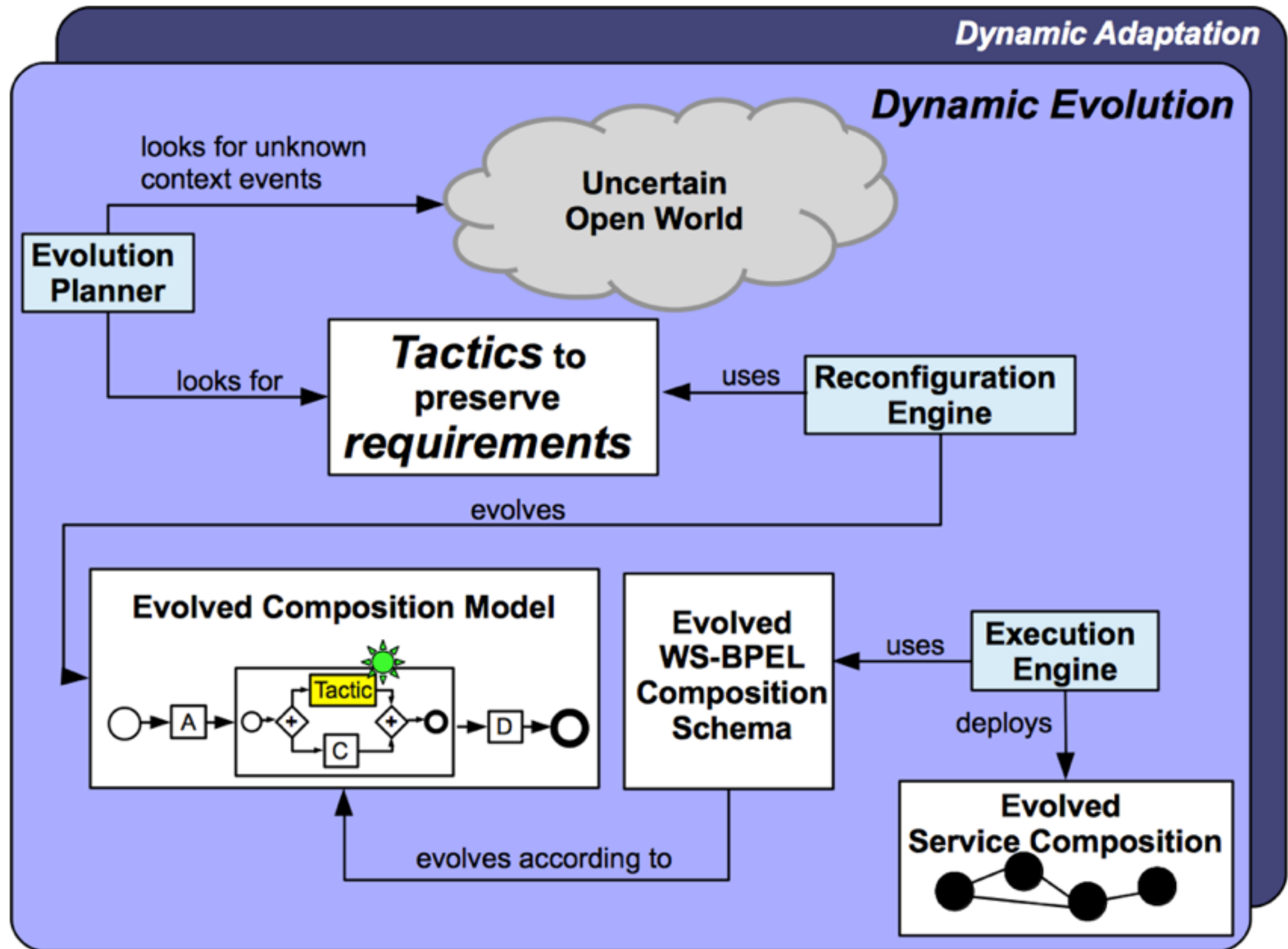
- They are **causally connected** to Web services that implement the tactical functionality.
- They are **merged** into the composition model at runtime.

How to find the **requirements** that can be affected by **unknown context events**?

Knowledge base implemented as a rules file.

```
1 @prefix j.0: http://my.ontology#
2 [underAttack: (?s rdf:type j.0:WebService)
3   (?s j.0:executionTime ?c)
4   greaterThan(?c,5000)
5   ->
6   (?s rdf:type j.0:UnderAttack)
7 ]
8
9 [affectedHighSecurityRequirement: (?s rdf:type j.0:UnderAttack)
10  ->
11  (?s rdf:type j.0:AffectedHighSecurityRequirement)
12 ]
```

Achieving Dynamic Evolution through Models at Runtime



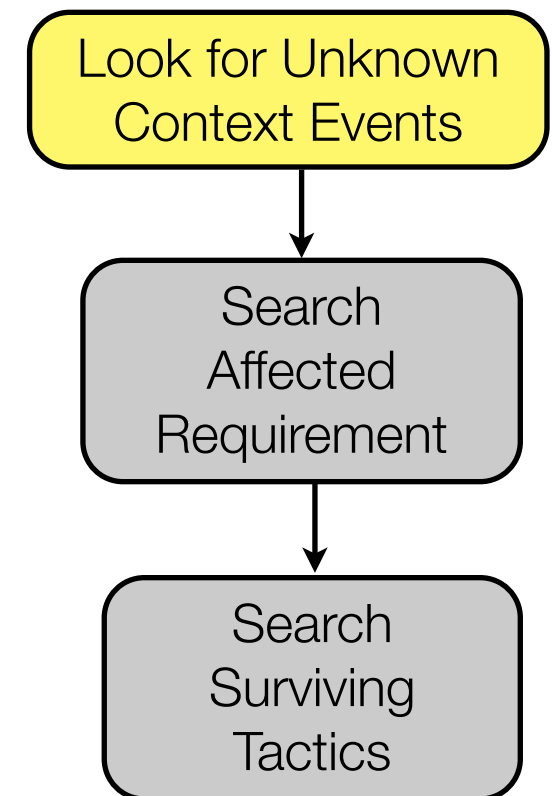
Solution - Achieving Dynamic Evolution through Models at Runtime

Evolution Planner

1) Look for Unknown Context Events from the Collected Information:

Periodically checks an updated **ontology** (Alférez and Pelechano, **SPLC** 2011; Alférez et al., **JSS Elsevier** 2014).

- An observed context event is considered as **unknown** when there are not predefined **context conditions** to deal with it. E.g. *UPSShipping*, *HasResponseTime*, $> 2,000\text{ ms}$.

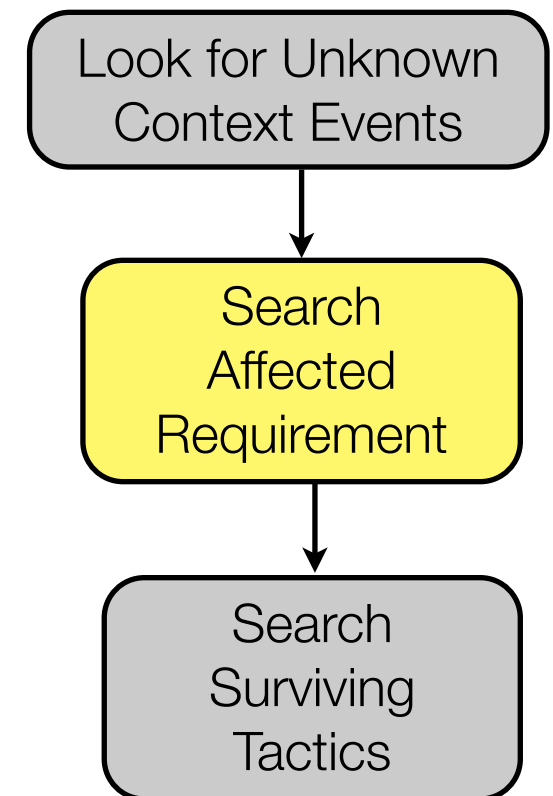


Evolution Planner

2) Search Affected Requirement(s):

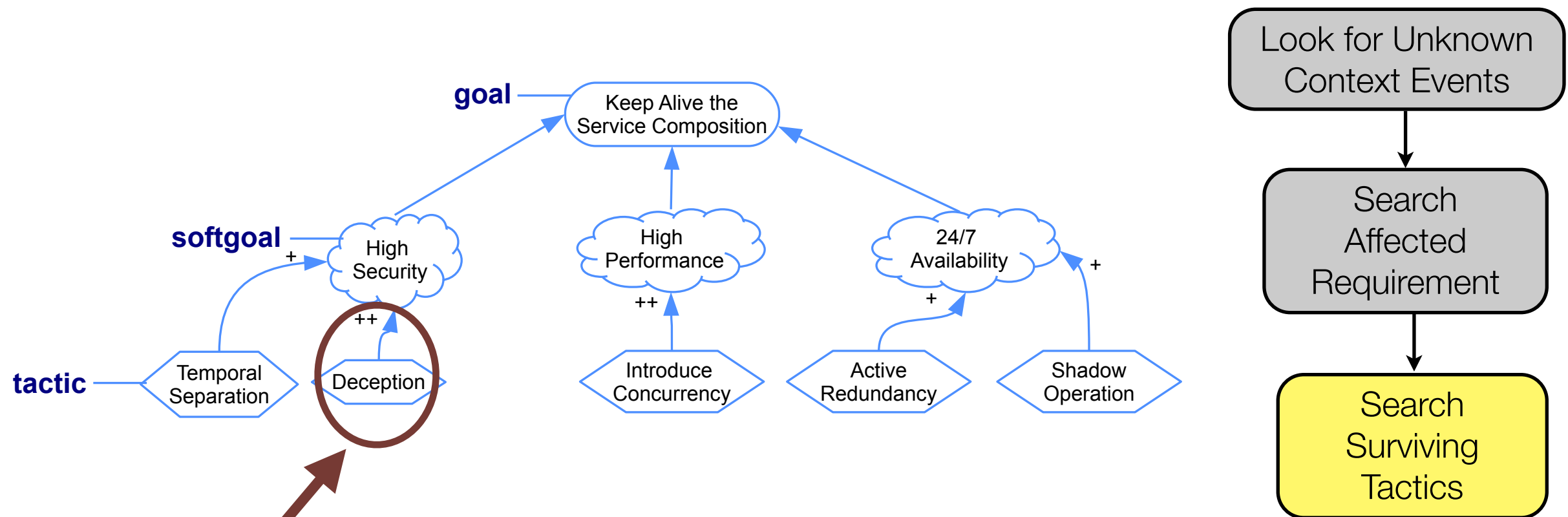
Forward chaining. This method evaluates arising context facts (i.e., context events) against general rule premises in the knowledge base. **New context events can trigger new inferences!**

Alférez and Pelechano, **MODELS** 2012;
Alférez and Pelechano, **ICWS** 2013.



Evolution Planner

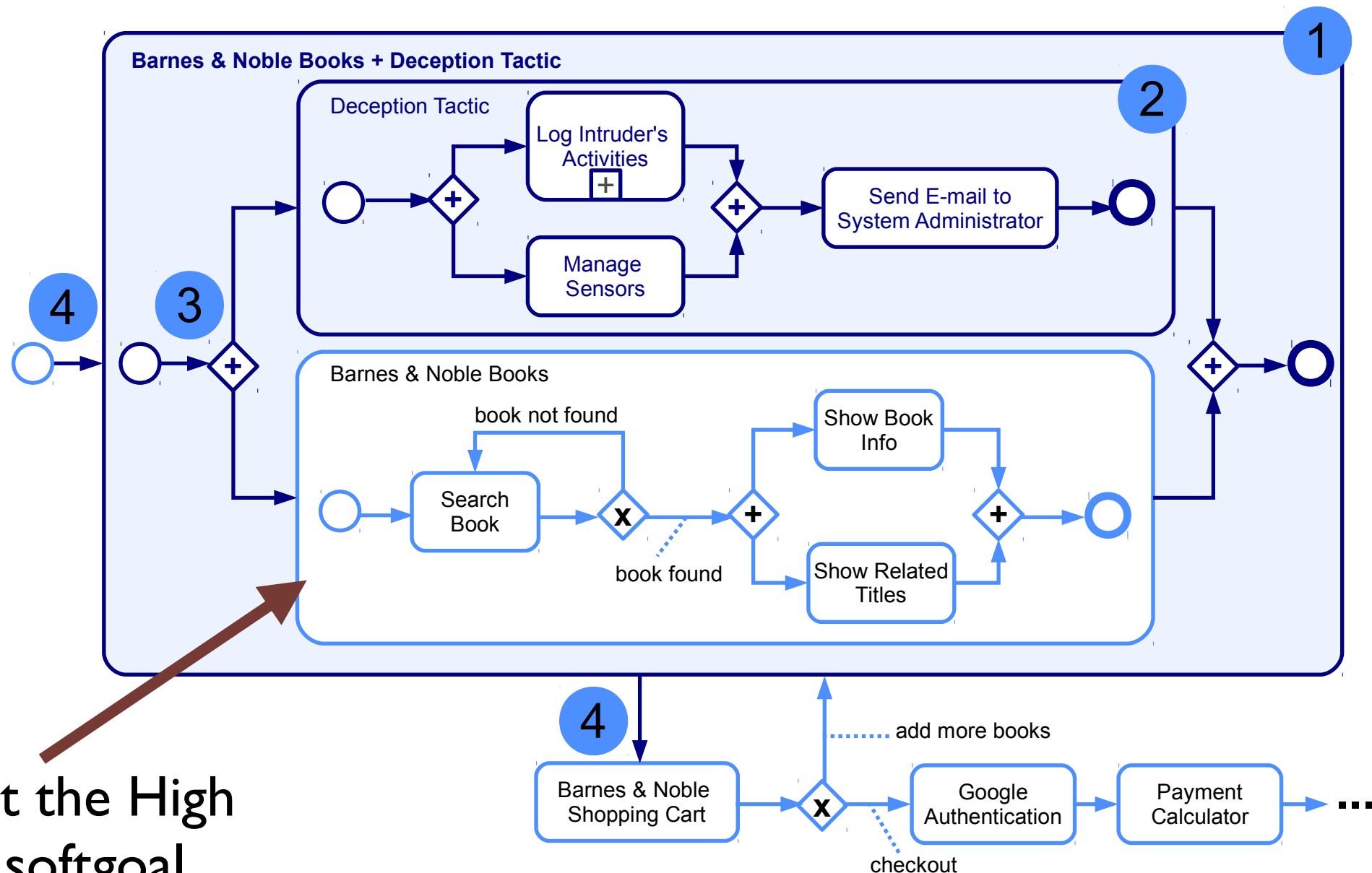
3) Search Surviving Tactics:



E.g. The Evolution Planner has inferred that “*The Barnes & Noble Books service operation can affect the High Security softgoal*”

Reconfiguration Engine

1) Merge a Tactic Model into the Composition Model:

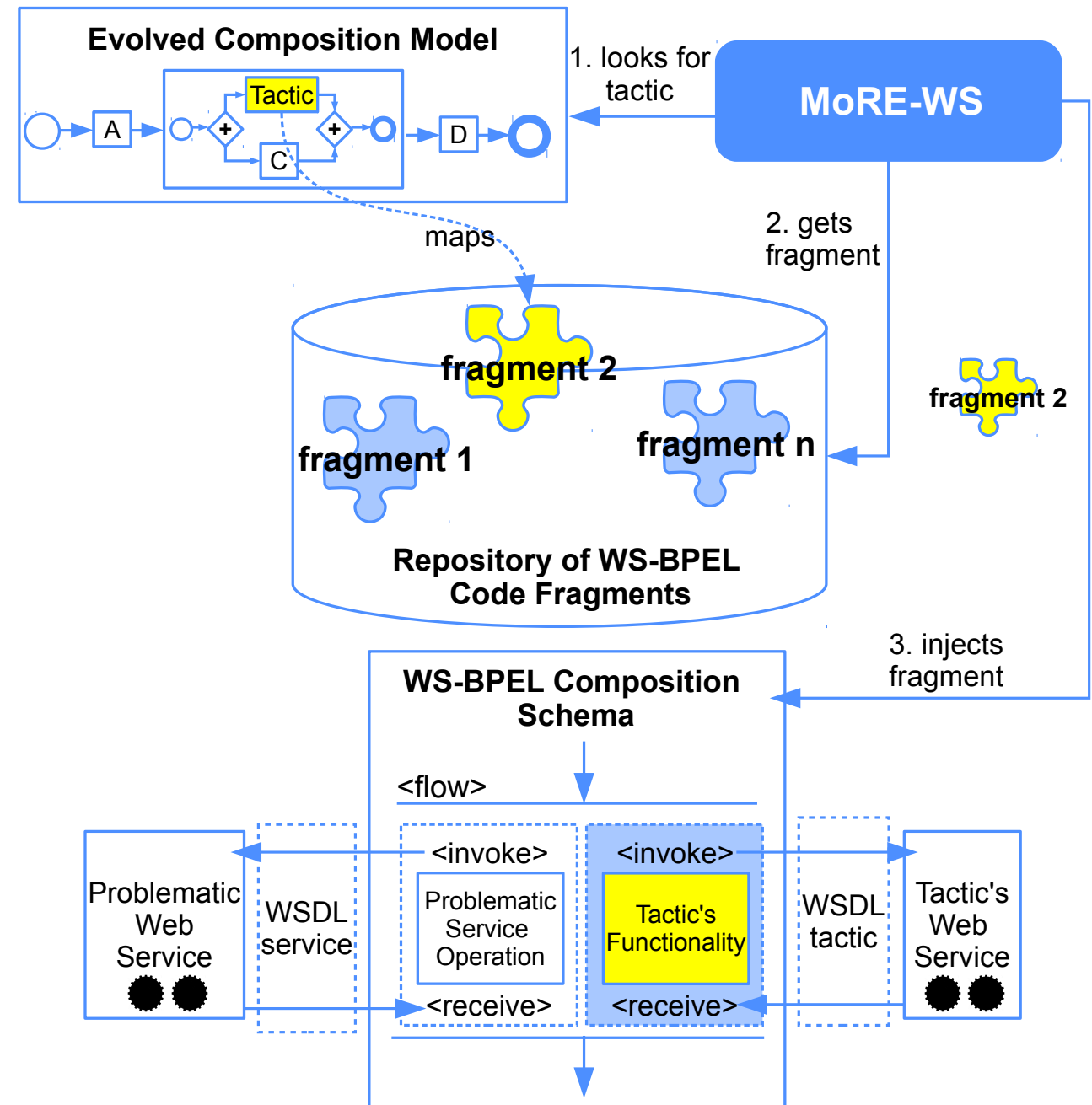


It can affect the High Security softgoal

Solution - Achieving Dynamic Evolution through Models at Runtime

Reconfiguration Engine

2) Evolve the WS-BPEL Composition Schema:



Solution - Achieving Dynamic Evolution through Models at Runtime

Prototype

The image displays a screenshot of the Eclipse IDE interface, showcasing a prototype for a dynamic evolution system. The interface is divided into several key components:

- Generic Editor - compositionModel.xml**: This window shows a hierarchical tree of the composition model. The "Model" section is expanded, revealing sub-processes like "Sub Process BarnesAndNobleBooksVar" and "Sub Process Deception". The "Deception" sub-process is further detailed, showing activities such as "Activity startEventForTactic", "Activity endEventForTactic", "Activity startGateway", "Activity endGateway", and a "Sequence Edge seqFromStartEventToStartGateway".
- Graphical Model Editor and Visualizer**: This window provides a visual representation of the BPMN diagram. It shows a flow starting from a start event, branching into two parallel paths (one labeled "Deception Tactic" and the other "Barnes & Noble Books"), which then merge and lead to a "Barnes & Noble Shopping Cart" activity.
- Models at Runtime**: This label points to the "models" folder in the Package Explorer on the left, which contains various model files like "activeRedundancyTacticModel.xml", "bpmn.ecore", "compositionModel.xml", "deceptionTacticModel.xml", "introduceConcurrencyTacticModel", "openome_model.ecore", "shadowOperationTacticModel.xml", and "temporalSeparationTacticModel.xml".
- Evolved Composition Model (EMF Form Editor)**: This label points to the "Properties" and "Console" tabs at the bottom. The "Properties" tab shows the "Ncname" and "Ordered Messages" for the selected model. The "Console" tab displays the execution log, including messages like "Loading the Deception BPMN model for merging..." and "Creating the parallel relationship between the problematic functionality and the tactic...".
- Dynamic Evolution Console**: This label points to the "Console" tab, which shows the detailed execution log of the system, including messages like "Inserting... bpmn.impl.ActivityImpl@58933ff7 (id: 2nkVFNX3EeK_J2kDGu28A) (documentation: null, name: LogIntrudersActivities, ncname: null, o...)" and "Inserting... bpmn.impl.ActivityImpl@333dd47e (id: d2aWhnX3EeK_J2kDGu28A) (documentation: null, name: ManageSensors, ncname: null, o...)".

<http://www.harveyalferez.com/dynamicevolutionservcomp/>

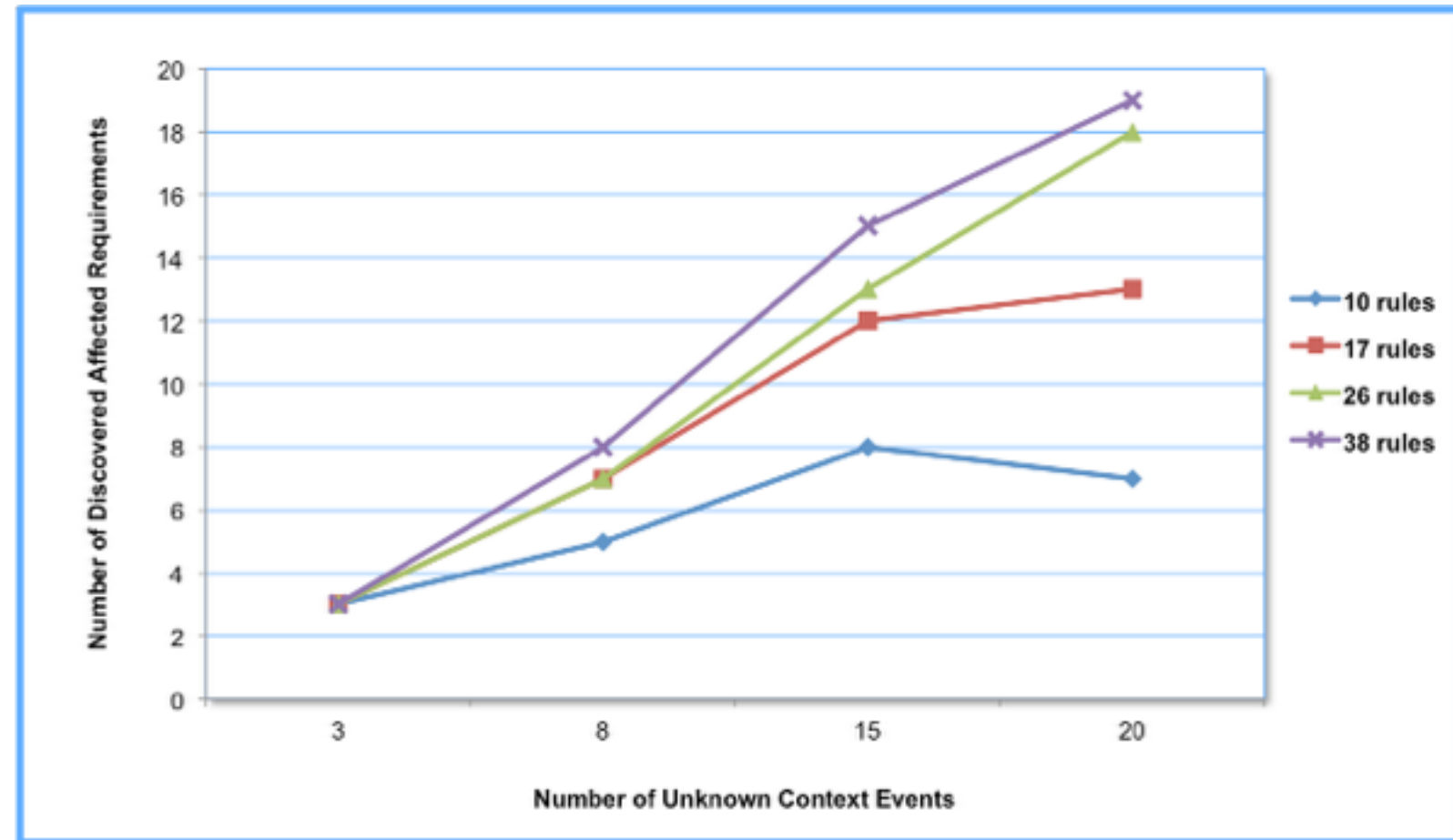
Preliminary Results



Inferences Accuracy

- We purposely injected **a set of context events that were not predefined at design time.**
- Our approach found the affected requirements in **83.9%** cases.

Reduced Uncertainty

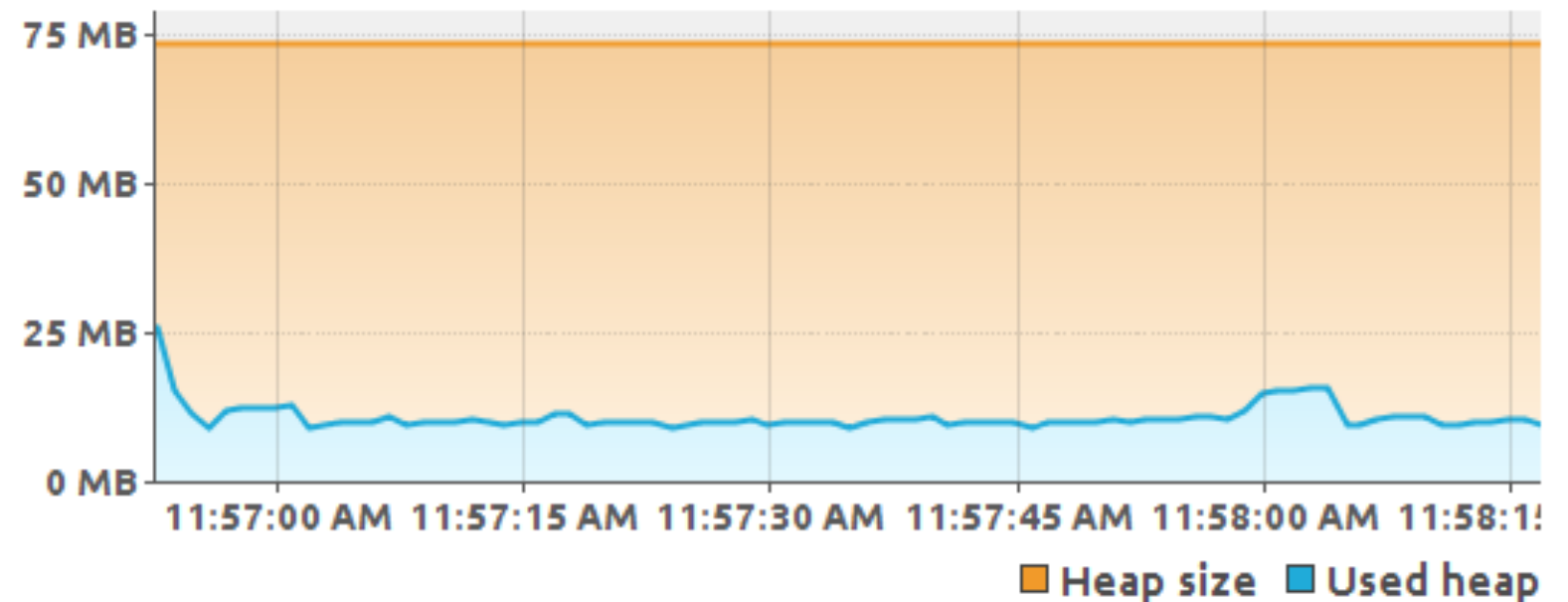
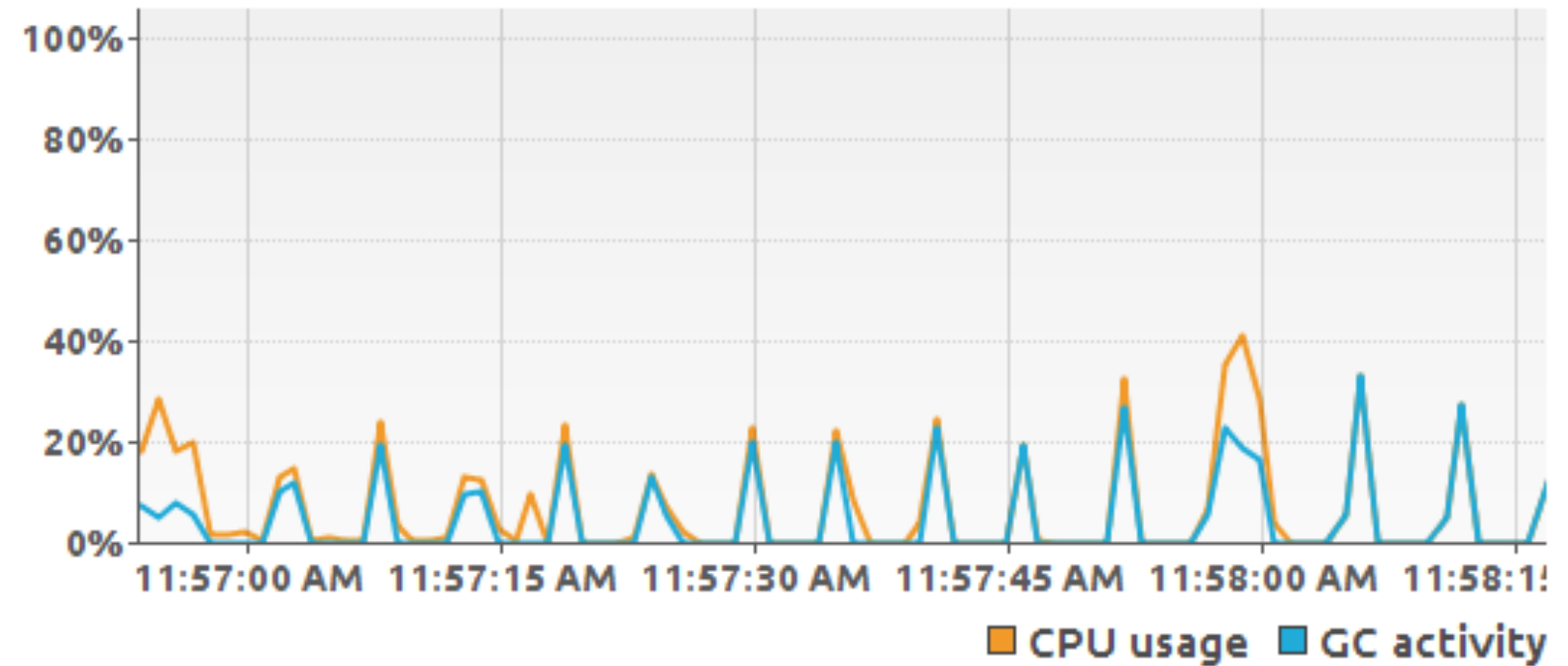


The number of discovered requirements that can be negatively affected is directly proportional to the number of rules.

Dynamic Evolution Efficiency

Measures during **a dynamic evolution.**

Efficient Dynamic Evolution



Validation

Conclusions and Future Work



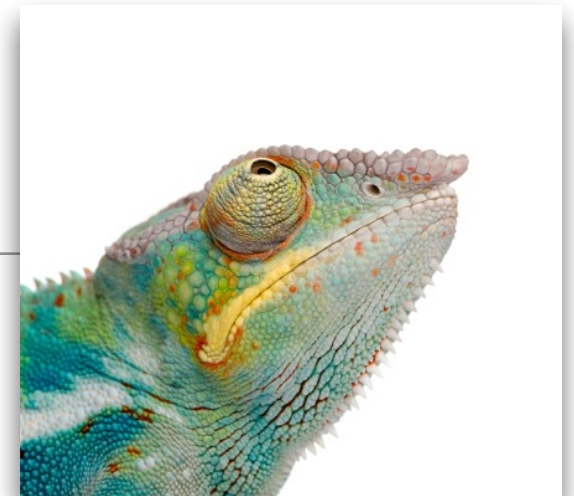
- A **tool-supported approach** that leverages **models at runtime** to guide the **dynamic evolution** of **context-aware service compositions** in the **open world**.
- It covers **design time** and **runtime**.
- It can be used to **manage uncertainty** produced by **unknown context events**.

- The use of **models at runtime** has the following **benefits**:
 - The modeling effort made at design time also provides a rich semantic base for autonomic behavior during execution.
 - They provide up-to-date information to drive subsequent evolutions.
 - Technological bridges are avoided.

- Use **Constraint Programming** to **verify** the evolved models and check that generated configurations respect the constraints imposed by the models.
- Carry out **proactive** dynamic evolutions with **machine learning**.
- Apply the approach in other **domains**:
 - Robotics
 - Smart Cities
 - Wearables
 - Cloud Computing
 - The Internet of Things

Thanks!

www.harveyalferez.com
harveyalferez@um.edu.mx



UNIVERSIDAD
POLITECNICA
DE VALENCIA

