Centro de Investigación en Métodos de Producción de Software

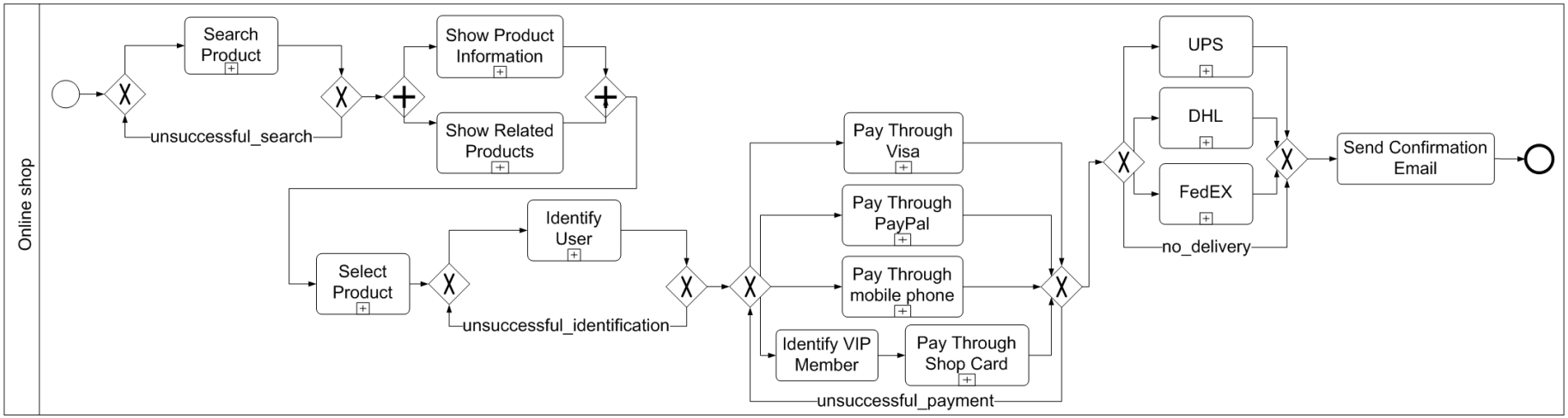
Research Center on Software Production Methods

Applying CVL to Business Process Variability Management

Clara Ayora, Germán H. Alférez*,
Victoria Torres, and Vicente Pelechano

Motivation

Business Process (BP): A set of related activities whose execution reaches a specific goal [Weske, 2007].



Motivation

- BP models often **vary** depending on the **application context** (i.e., the execution environment) [Hallerbach et al, 2008; Reinhartz-Berger, 2009]:
  - *Type of payment* or *delivery method*.
  - **Large collections** of related **process variants**.
    - **Process variants** pursue the same or similar business objective (maintenance of vehicles in a garage or the treatment of a patient).

Examples

- A repository for **vehicle repair and maintenance** comprising more than **900 process variants** that depend on country, garage, and vehicle differences [Hallerbach et al, 2010].

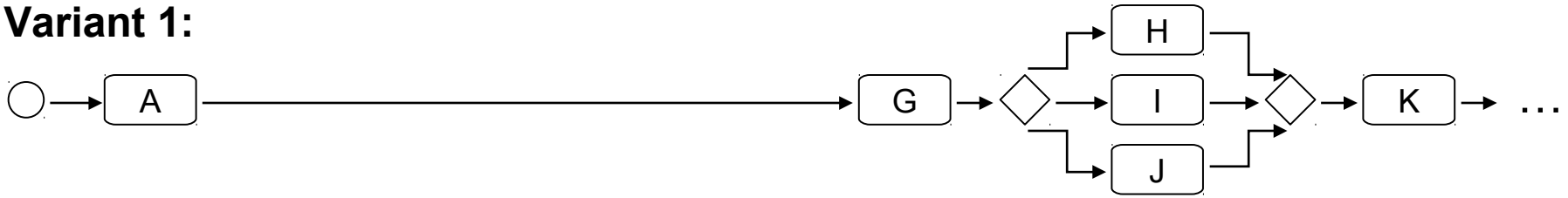
- **More than 90 process variants** for handling medical examinations [Li et al, 2011].

Motivation

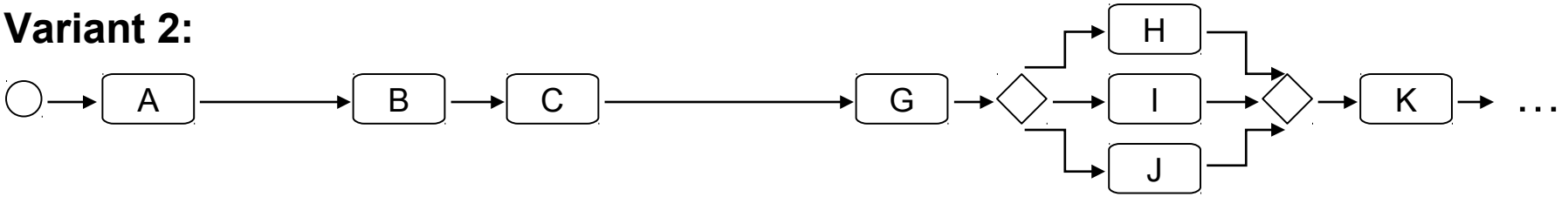
A lot of possibilities for BP variability.

Motivation

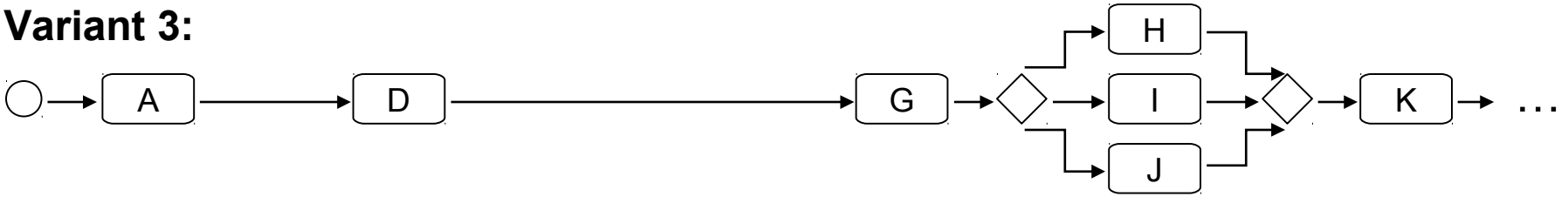
Variant 1:



Variant 2:

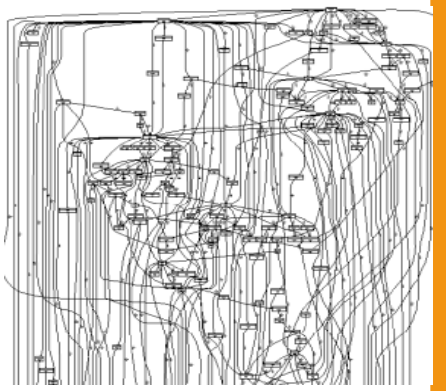


Variant 3:



Motivation

- Managing properly process variants constitutes a fundamental **challenge** to **reduce development and maintenance efforts** [Müller, 2006].
  - Managing process variants is **not a trivial task**.



- ***Design time:*** Current **support** to represent process variability is **limited**.
  - Process variants become error-prone and complex to build, manage, and understand.

- ***Runtime:*** Emerging necessity to adapt process variants at runtime that run in **changing contexts** [Hermosillo, 2010].
  - The dynamic adaptation of BPs at the **language level** is **complex** and time consuming, especially in large systems.

Contribution

- A proposal to support **process variants** at **design time** and **runtime**.

- At **design time:**
  - BP models represent, by means of **tasks** and **resources**, how **business goals** are achieved.
  - **Process variants** are considered as first-class concerns.
  - **Common Variability Language (CVL):** provides the mechanisms to represent variations in any Domain Specific Language (DSL).
    - DSLs do not have to be extended or overloaded with variability information [Fleurey, 2011].
    - CVL + Business Process Modeling Notation (BPMN).
      - Neither the current specification of BPMN supports process variability modeling
      - Nor current BPMN execution engines support variability.

Contribution

- At **runtime** we rely on **models at runtime** [Blair, 2009] to perform **dynamic adaptations** of process variants according to the current context.
  - **Models** are typically used to describe systems using concepts that **abstract** the system knowledge **over the underlying computing technologies**.
  - **Purpose:** to extend the use of models from design time to execution time.
    - The modeling effort made at design time is not only useful for producing the system.
    - Provide a richer semantic base for reasoning, monitoring, or adapting the system during execution [Cetina et al, 2009; Alférez et al, 2011].
    - Causally connected.

Contribution

- With **models at runtime**:
  - No cumbersome programming or error-prone code to carry out adaptations.
  - Models at runtime allow to reuse the knowledge created with CVL at design time to guide the adaptations during execution over the underlying technologies.

- Adaptations are automatically achieved by our **Model-based Reconfiguration Engine for Business Processes (MoRE-BP)**.
  - MoRE-BP is an extension of MoRE-WS [Alférez et al, 2011].
  - Autonomic Computing.

Contribution

Autonomic Computing

- Create software that self-manage in a bid to overcome the complexities to maintain systems effectively.

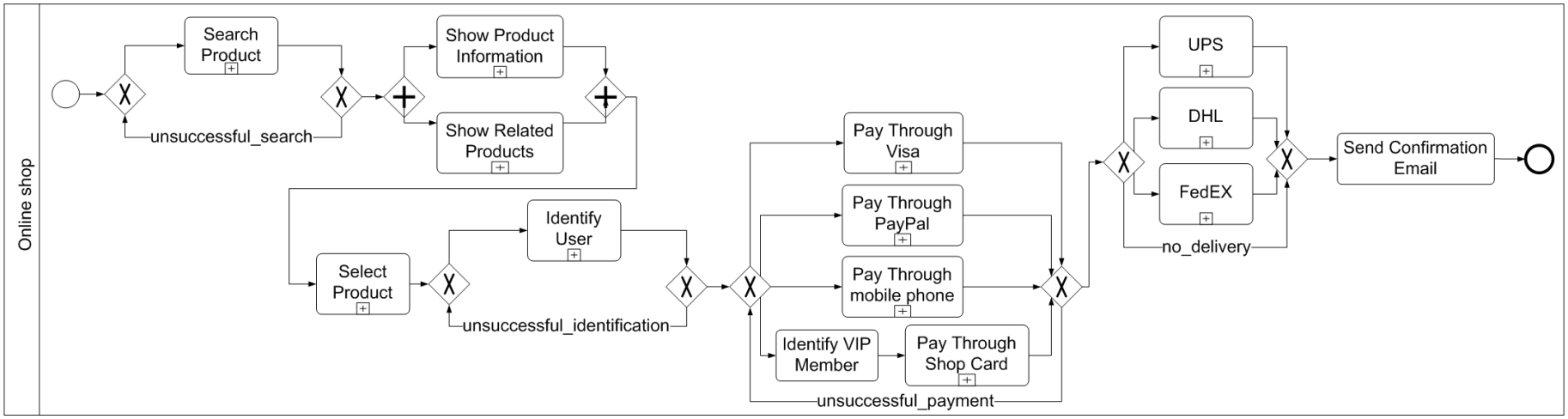
- Covers the broad spectrum of computing in domains as diverse as mobile devices [White et al, 2007] and home-automation [Cetina et al, 2009].

- Automating tasks such as **installation**, **healing**, and **updating**.

Case Study

- There are **16 process variants** depending on either the **type of payment** or the **shipping company**!!!

- All possible process variants can be **defined at design time** since the different alternatives (e.g. UPS and DHL) are known in advance.



Handling Variability in Business Processes

Variability management in BPs:

1) At **design time** when the process variants are modeled.

2) At **runtime** when the process variants are adapted in response to context changes.

Handling Variability in Business Processes

Process Variant Modeling

- The aim is to **properly model** the existing **process variants**.

- Current proposals model process variants by **extending the original DSLs** (e.g. BPMN, EPC, or UML Activity Diagrams).

  - PESOA [Puhlmann et al, 2006] and C-EPC [Rosemann et al, 2007] integrate all possible process variants in a single model resulting in:
    - **Large and difficult-to-understand models**.
    - Models are **overloaded** with **variability specifications**.

Process Variant Modeling

- On the contrary, **CVL** provides the mechanisms to **represent variations in a separate way**.

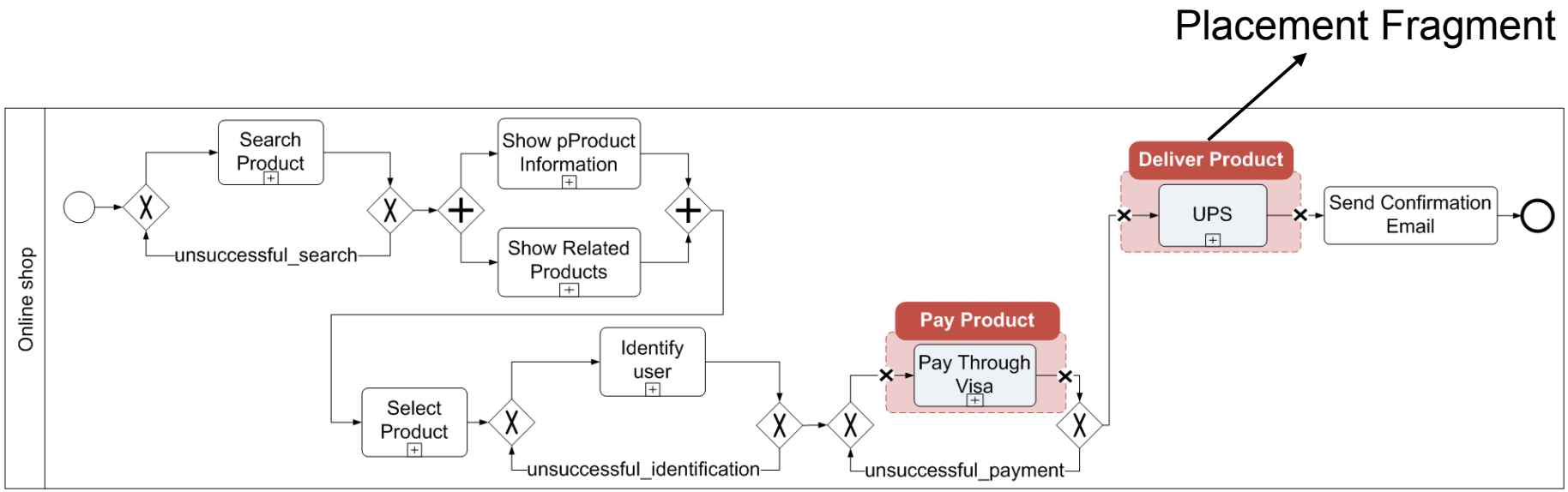
  - CVL alleviates the impact that variability issues have on the BP model, resulting in:
    - Better legibility
    - Understandability
    - Scalability

  - CVL is based on the **Base-Variation-Resolution (BVR)** approach that is supported by three models:
    - The base model
    - The variation model
    - The resolution model

Process Variant Modeling

Base Model

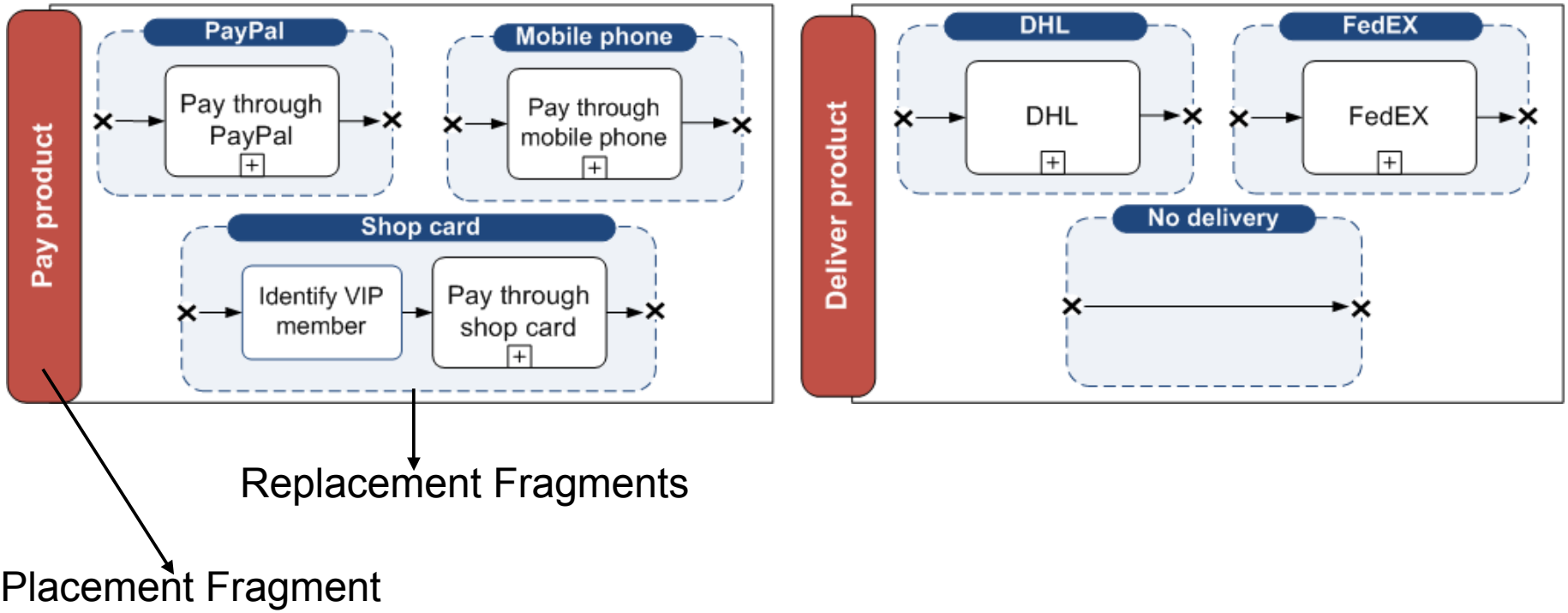
- Common parts to all process variants
- **Placements fragments** (i.e., variation points)



Process Variant Modeling

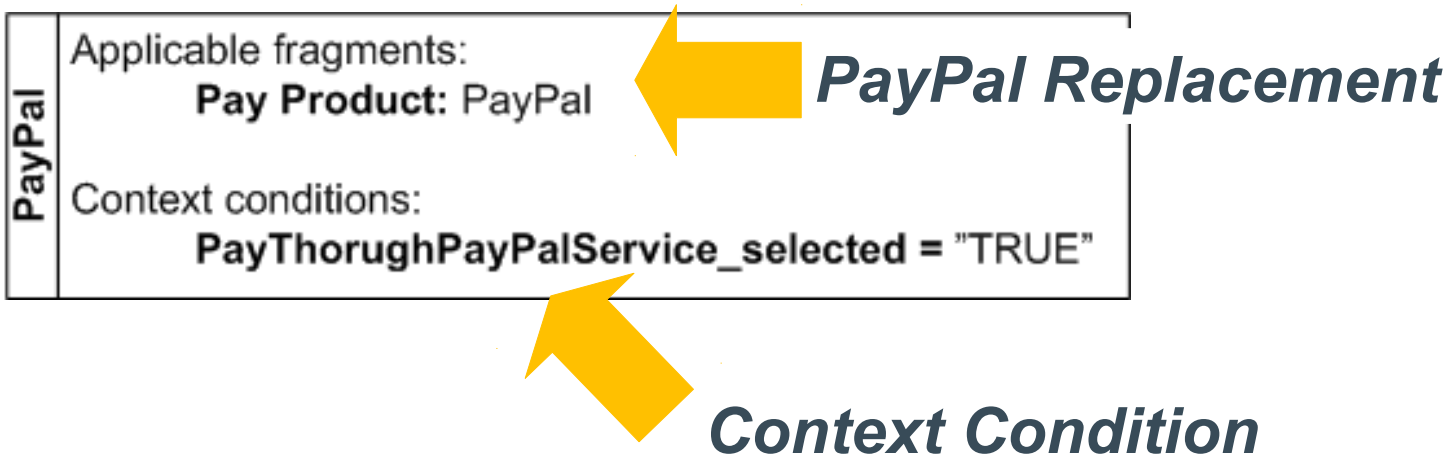
Variation Model

- **Replacement fragments** (i.e., alternatives)



Process Variant Modeling

- **Resolution Model**
  - Specifies the set of **context conditions** that determine the conditions under which the replacements can be instantiated.



Process Variant Modeling

- **Context Model**

➔ Supports the formal reasoning of the context information.
➔ Ontology-based provides a strong semantic vocabulary for the representation of the context knowledge and for describing specific situations in the context [Alférez et al, 2011].
➔ It enables the analysis of the domain knowledge using first-order logic.

Variability Management at Runtime

Existing proposals dealing with BP variability at runtime:

- SCENE [Colombo et al, 2006] **extends BPEL** with Event Condition Action (ECA) rules that define consequences for conditions to guide the execution of binding and rebinding self-reconfiguration operations.

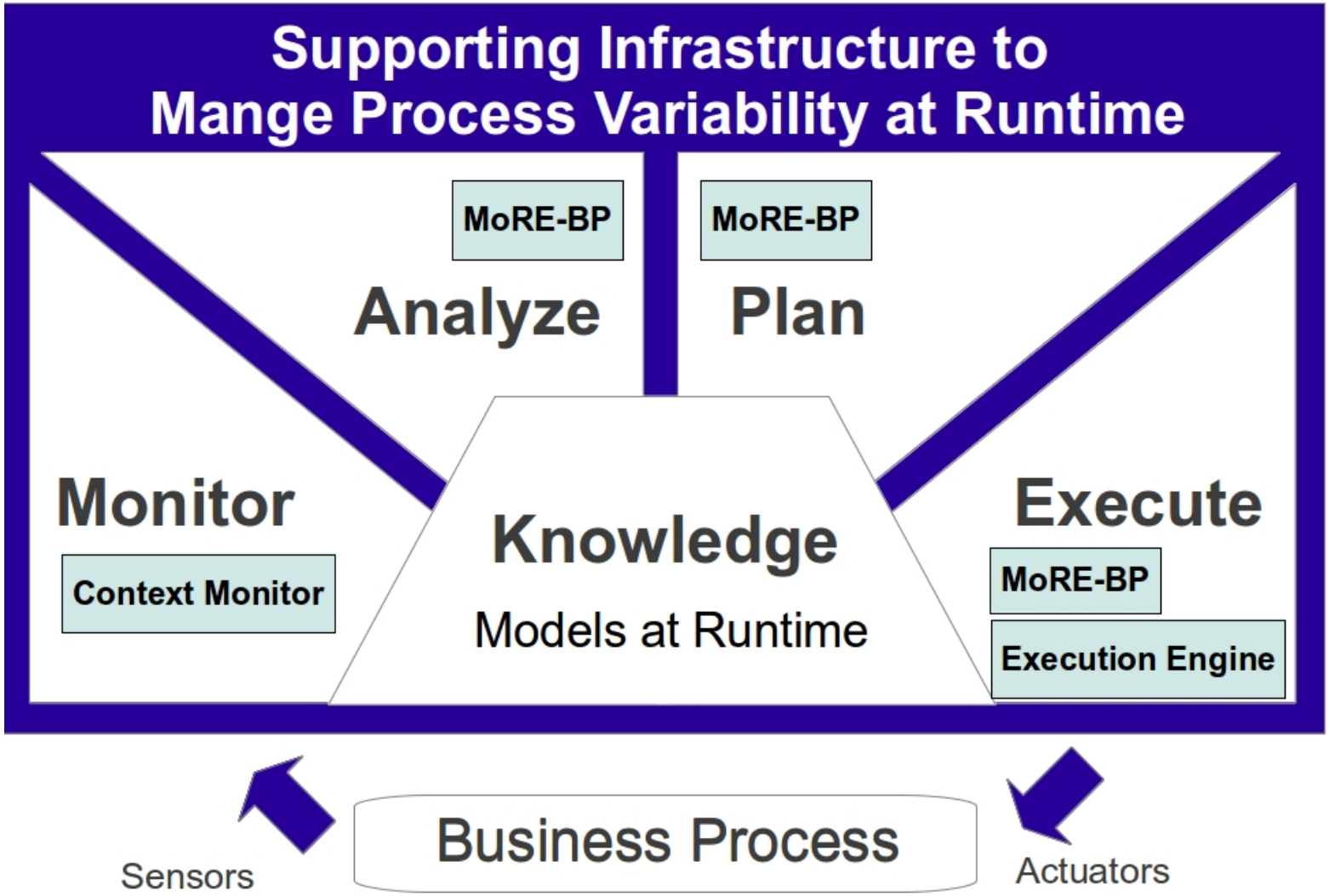
- VxBPEL [Koning et al, 2009] is an **adaptation of BPEL** that allows adapting a BP in a service-centric system.

- CEVICHE [Hermosillo et al, 2007] **extends BPEL** to directly include into the code the adaptation points and conditions that are required to create dynamic adaptable BPs.

Variability Management at Runtime

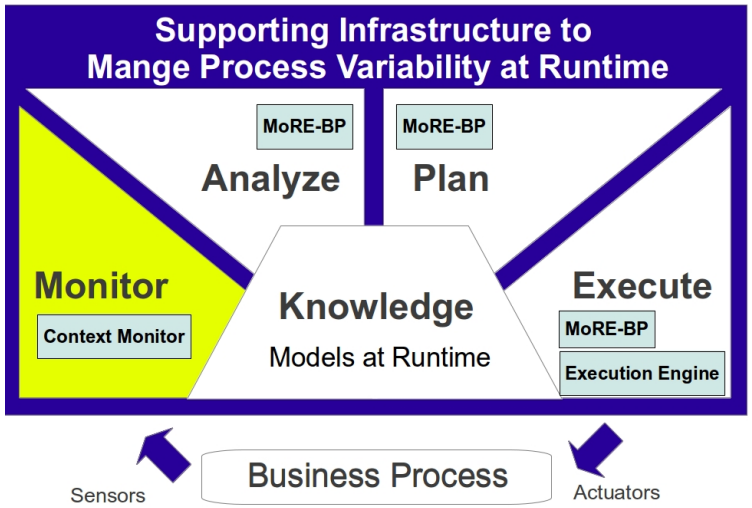
- **Dynamic adaptation** of **process variants in a higher level of abstraction**:
  - The **base model** is **dynamically adapted** according to the **variation model**.
  - **Adaptations** are supported by a **computing infrastructure** based on the components of the **MAPE-K loop** [IBM, 2003].

Variability Management at Runtime



Variability Management at Runtime

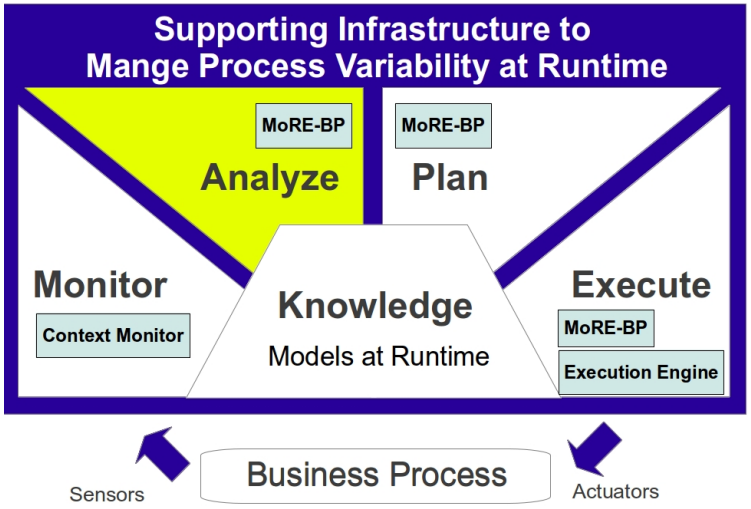
Monitor



- Collect information about the **context**.
  - This task is in charge of the **Context Monitor**.
  - The collected information is used to update a **stream database** that deals with continuous online data streams.

Variability Management at Runtime

Analyze



- The **stream database** that is updated with the measures taken from the context needs to be queried to determine if any adaptation has to be carried out.
  - MoRE-BP periodically queries the stream database to find new context information.
  - When a new context event is found, MoRE-BP inserts it into the **context model**.
  - Then, MoRE-BP **evaluates** the values of this model to find out if a **context condition** has been accomplished.

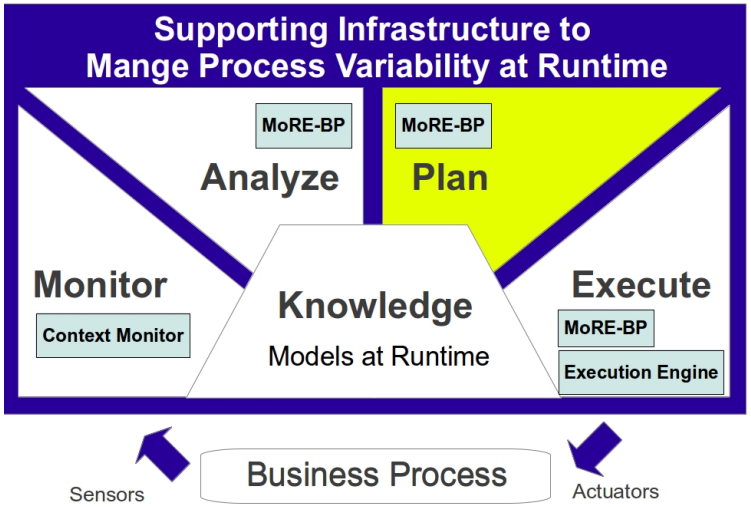
Variability Management at Runtime

Analyze

- For instance, if the user prefers to pay through **PayPal** accomplishes the **PayThroughPayPalServiceSelected** context condition
  - Triggers an adaptation in the **Pay Product placement**.

Variability Management at Runtime

Plan



- MoRE-BP generates an **Adaptation Plan**, which contains a list of actions **to adapt the base model**.
  - These actions are stated as CVL actions called **fragment substitutions**.
  - A **fragment substitution** replaces the **process fragment** included in any **placement** of the **base model** with any **replacement** of the **variation model**.

Variability Management at Runtime

Plan

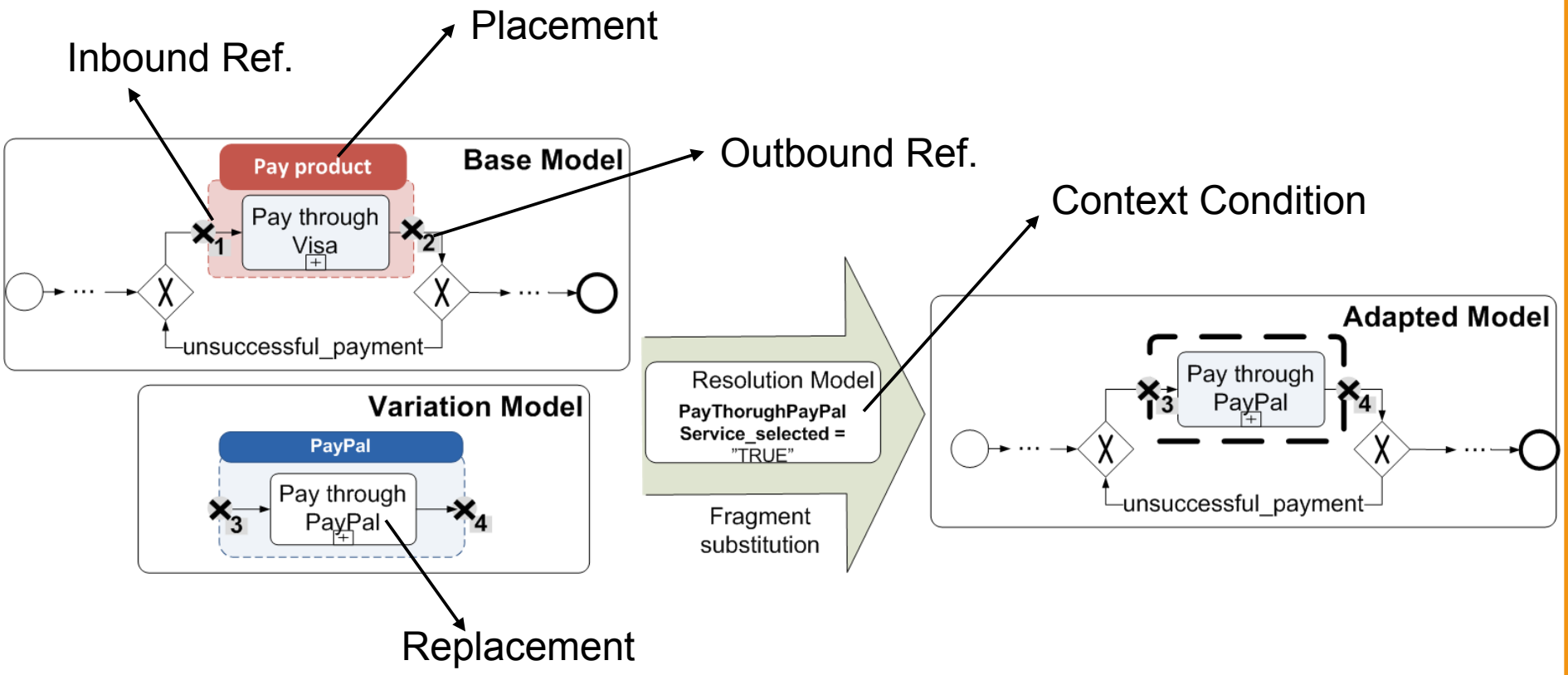
- A fragment substitution (FS) can be defined as follows:

FS = ((PlacementInboundReference, **FragmentToBeReplaced**, PlacementOutboundReference), (ReplacementInboundReference, **Replacement**, ReplacementOutboundReference))

FS = ((PayProductsInputSequenceEdge, **PayThrough Visa**, PayProductsOutputSequenceEdge), (PayThroughPayPalInputSequenceEdge, **PayThrough PayPal**, PayThroughPayPalOutputSequenceEdge))

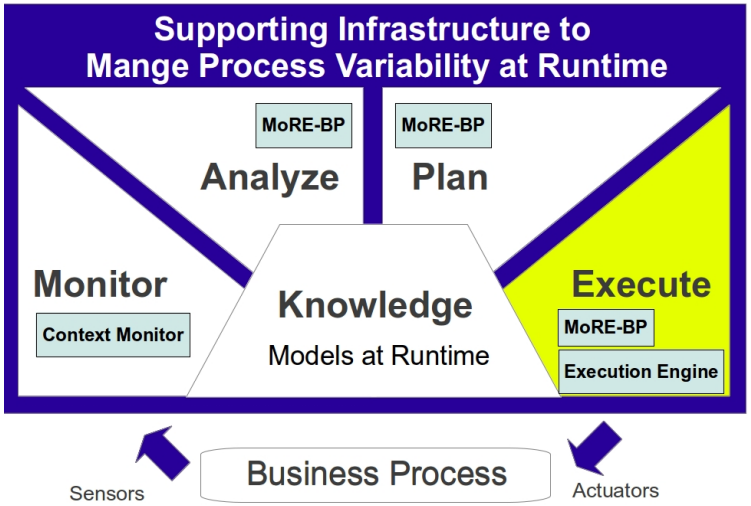
Variability Management at Runtime

Plan



Variability Management at Runtime

Execution



- **Transform** the **adapted base model** into executable **BPEL code** and **hot deploy** it in the **Execution Engine**.
  - MoRE-BP creates a **deployment directory:** the **deployment descriptor** and the **process schema** (i.e., the BPEL file).
    - A **new deployment directory** with an **increasing version number** is deployed with every adaptation.

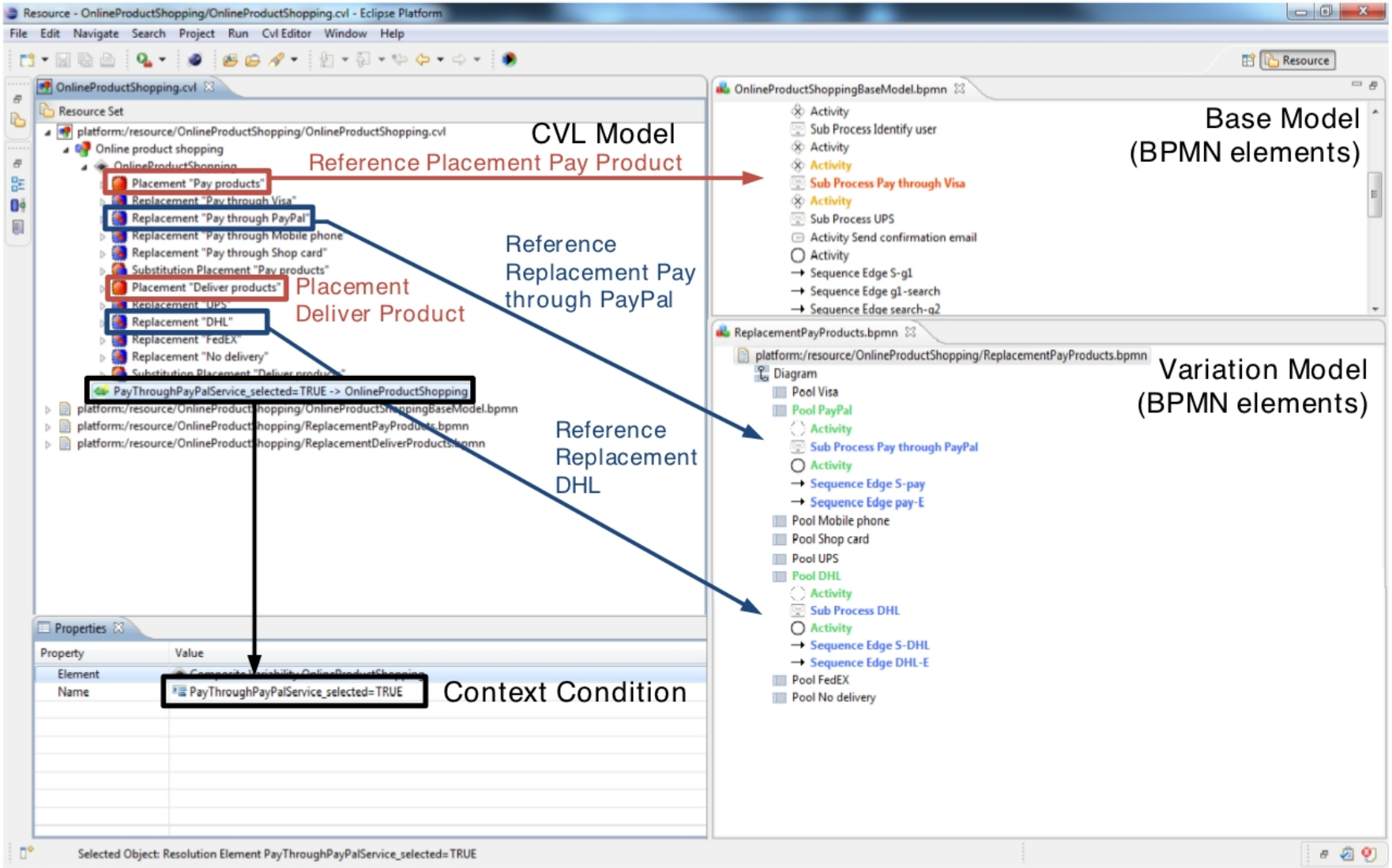
Prototype

- The **base model** and the **variation model** are defined with the Eclipse BPMN Modeler.

- CVL Editor to create a CVL Model.

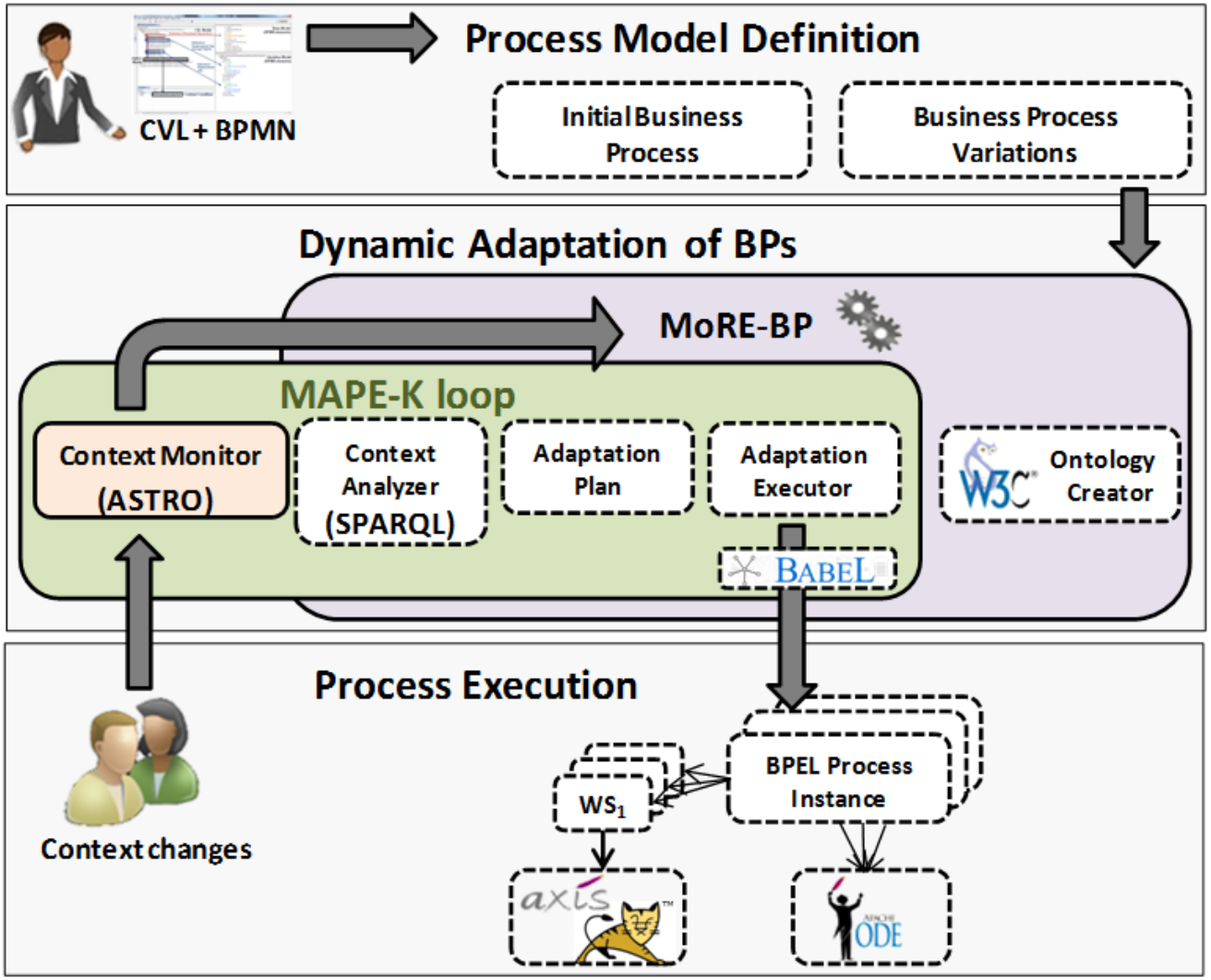
Centro de Investigación en Métodos de Producción de Software

Prototype



UNIVERSIDAD POLITECNICA DE VALENCIA

Prototype


Process Model Definition
CVL + BPMN
Initial Business Process
Business Process Variations
Dynamic Adaptation of BPs
MoRE-BP
MAPE-K loop
Context Monitor (ASTRO)
Context Analyzer (SPARQL)
Adaptation Plan
Adaptation Executor
Ontology Creator
BABEL
Process Execution
Context changes
WS1
BPEL Process Instance
axis
ODE

Conclusions

- Proposal to manage BP variability based on CVL.

- At design time, we have used CVL to model the possible process variants.
  - Since CVL is an independent language, no annotations or variability concepts need to be added to the original DSL (i.e., Business Process Modeling Notation).
    - **CVL improves the quality of the model in terms of legibility, understandability, and scalability.**

- At runtime, MoRE-BP uses the CVL specifications to perform dynamic adaptations of the process variants.

Further Work

- Investigate the adaptation of process variants in response to **unexpected context changes**.

- Provide **constraint mechanisms** to **ensure consistent resolutions** leading to **well-formed process variant models**.

Centro de Investigación en Métodos de Producción de Software

Thank you!!

cayora@pros.upv.es

UNIVERSIDAD POLITECNICA DE VALENCIA