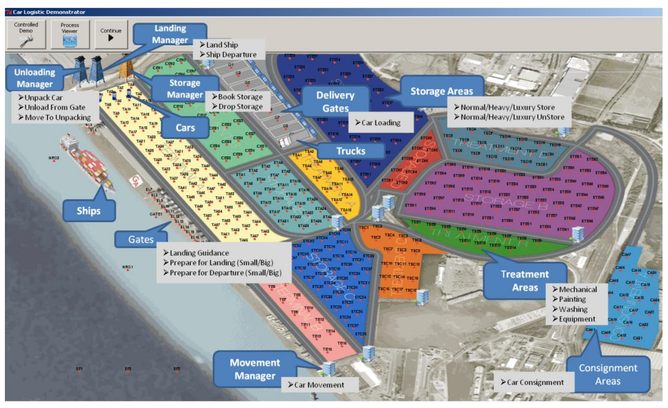
CAptEvo: Context-aware Adaptation and Evolution of Business Processes

Motivation



Car Logistics: The motivating case study is inspired by the operation of the sea port of Bremen, Germany. The port receives ships loaded with cars and has to organize their delivery. The operation of the system involves numerous entities (Ships, Cars, Gates, Landing Manager, Unloading Managers, Terminals, Treatment Areas, Delivery Managers, Delivery Gates, Trucks...), each responsible of specific tasks and characterized by its own needs and requirements.

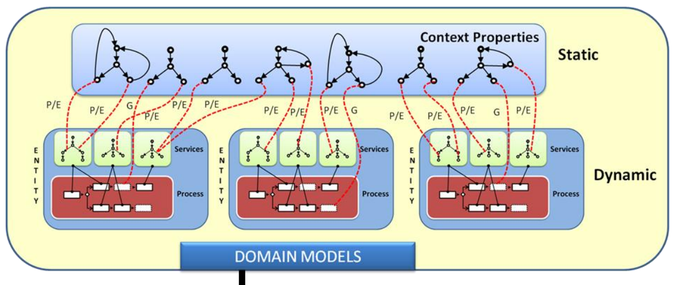
Research Challenges: The case study indicates relevant research challenges:

Context Awareness and Dynamicity. The system execution needs to take into account the dynamic nature of the context: availability of resources and services, and specific characteristics of the entities populating the system. Context awareness is concerned with the capability of the processes to "sense" their execution environment and to dynamically detect context changes.

Process Adaptation. Process adaptation guarantees the capability of running processes to adapt to context changes according to their goals. Adaptation needs may be triggered by specific cases to be handled, by dynamic service availability, by non-controllable situations depending on environmental conditions, as well as by changing requirements.

Process Evolution. The need for continuous adaptation results in a system characterized by a huge set of process executions that, although instantiated on the same process model, strongly differ in terms of process structure. The aim of process evolution is to derive long-term changes from short-term variants to progressively improve the process model.

The CAptEvo Framework



Domain Models

➢ **Context Properties**: We consider context as a set of properties representing important characteristics of the environment and of the entities that operate in it. Each context property is explicitly modeled with a *context property diagram*.

➢ **Entities.** We model the set of entities considering its own business process and its provided services. To model services with complex protocols we use *state transition systems*. To capture impact of service execution on the context and to encode business policies we annotate service transitions with context effects and context preconditions respectively. The business processes can contain *abstract activities* that are refined at runtime to achieve a predefined *goal.*

Viewer. We have implemented a simulation environment that allows for displaying our framework in action, visualizing the *execution* and the *adaptation* of business processes in concrete application scenarios.

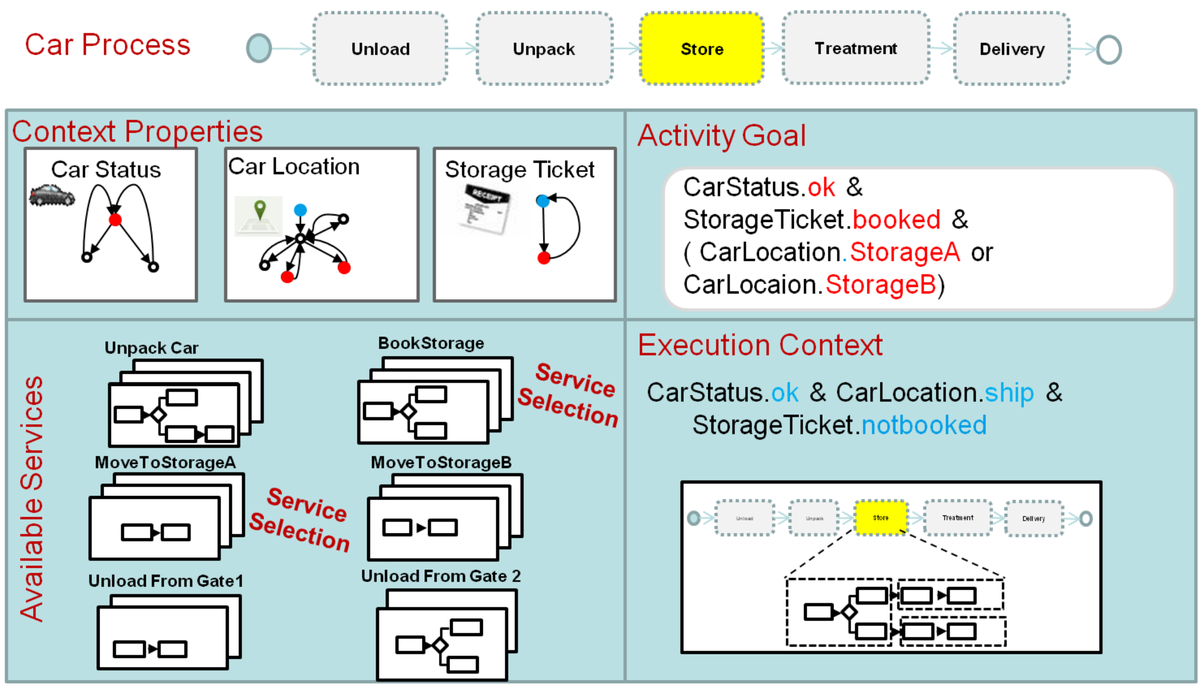
System Operation

➢ **Process Engine**: The activities of the each process instance are executed by the *Process Engine*. Before executing next activity it checks if the process needs to be adapted. When the observed values of the context properties violate the preconditions of the next activity to be executed, the process adaptation is initiated. Based on the current configuration of the context and of the process, the Process Engine derives adaptation problem and sends a corresponding request to the Adaptation Engine.

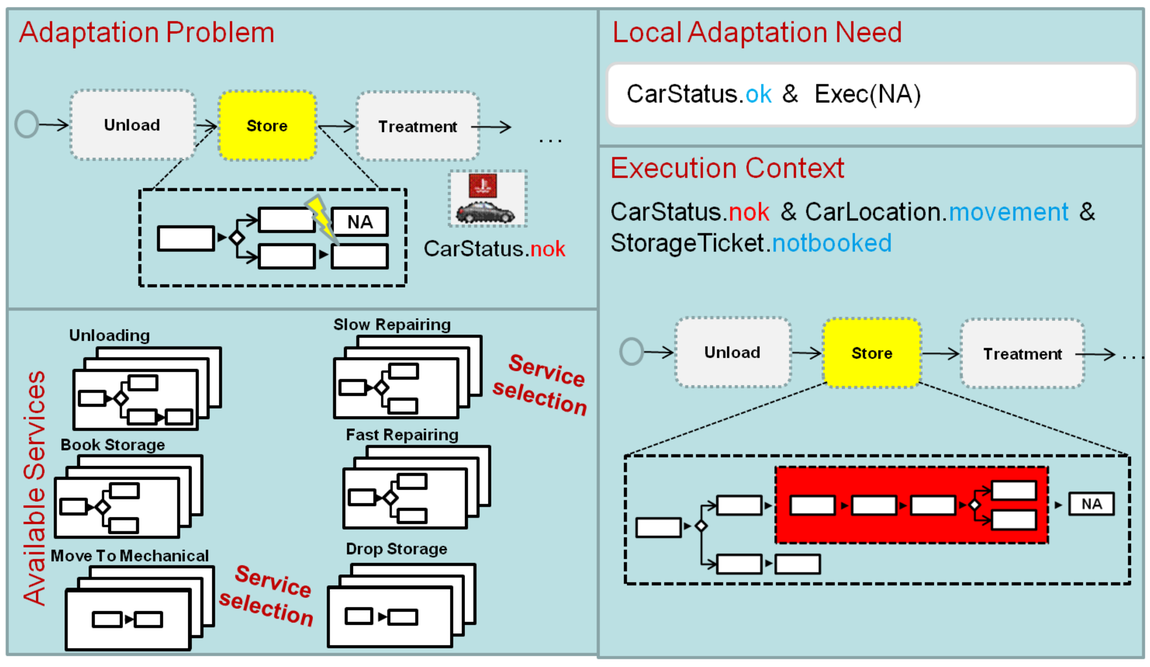
➢ **Adaptation Engine.** An adaptation problem sent to the *Adaptation Engine* comprises the current status of the system (values of the context properties, states of the available services involved), a set of available services that may be used for adaptation, and adaptation goals. The primary objective of the adaptation is to "unblock" the process, i.e. to satisfy the precondition of the next activity in the process. To accomplish this, the adaptation engine generates a composition of services that provides the necessary effects on the context.

➢ **Planning Techniques**. The construction of the service composition is performed with the use of automated planning techniques: the service specifications, the model of the context (i.e., context property diagrams), and the goal specifications are transfomed into a planning problem and the resulting plan is then transformed back into a composite service.

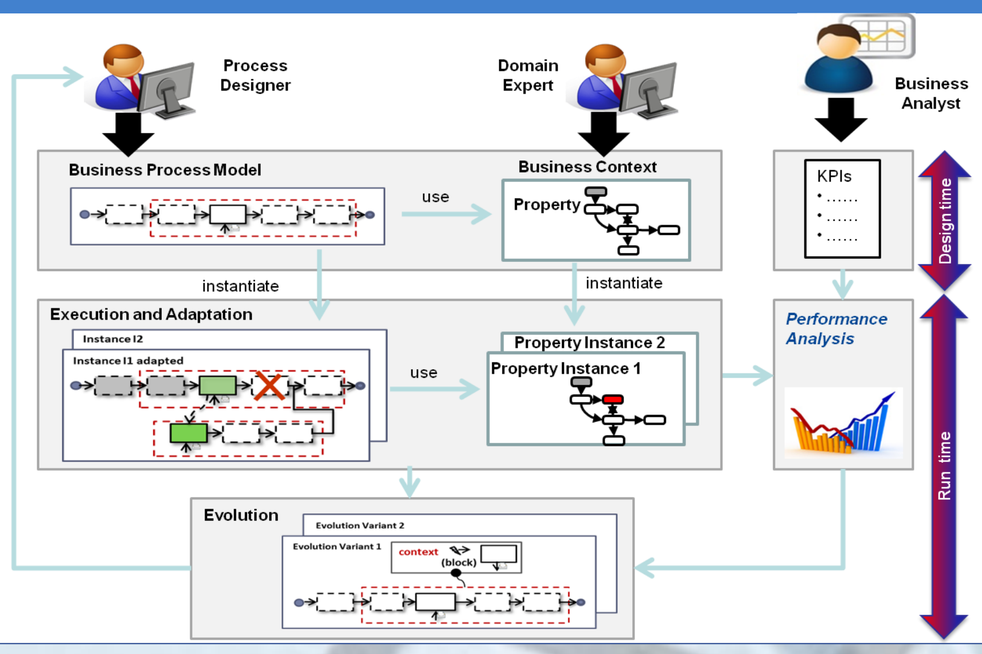
Dynamic Service Composition



Dynamic Process Adaptation



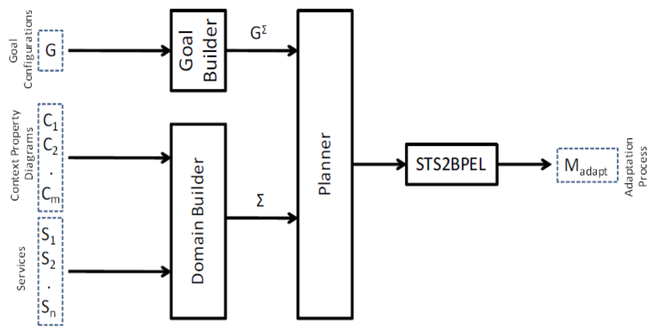
Business Process Evolution



Adaptation techniques

Adaptation Problems as Planning Problems

We exploit the ASTRO automated composition approach:
➢ Sophisticated AI planning techniques (Planning as Model Checking)
➢ Asynchronous domains, non-determinism, partial observability
➢ Complex goals: preferences, recovery, incremental goals



Simulation Environment: We have created a visualization environment enabling interaction between the framework and the user and simulating execution, adaptation and evolution of business processes in our case study.

Antonio Bucchiarone, Annapaola Marconi, Marco Pistore and Heorhi Raik
[bucchiarone, marconi, pistore, raik]@fbk.eu