ASTRO-CAptEvo
Dynamic Context-aware Adaptation for Service-based Systems

Antonio Bucchiarone, Annapaola Marconi, Marco Pistore, and Heorhi Raik

ServicesCUP– June 28, 2012
Honolulu, Hawaii
Car Logistic Scenario: Bremen Harbor

Bremerhaven Harbor
- 4 km², 120,000 storage places
- 2 Million vehicles per year
- 18 vehicle brands
- 12 technical treatment stations

Aim:
Develop a **Car Logistic System** supporting the management and operation of the port:
- covering the whole car delivery process, from delivery to disposition;
- supporting the co-operation of the numerous actors (i.e., cars, ships, trucks, storages, treatment areas, etc.) respecting their own procedures and business policies.
Car Logistic Scenario: Bremen Harbor

Challenges:

- **Customizable** procedures for each car (e.g., brand, model, order)
- **Heterogeneity** of actors/facilities involved (e.g., terminals, storage areas, treatment stations, consignment areas, ships, trucks)
- **System Dynamicity** (e.g., actors/facilities join/leave the system, changes in procedures of system facilities, changes in regulations and norms)
- **Context Dynamicity** (e.g., unavailability or malfunctioning of the different port facilities, accidental damages of cars and trucks, human errors)

Fully exploit the benefits of the service-oriented paradigm to develop a context-aware adaptive system.
A Framework for Adaptive Context-aware SBS

**Business processes:**
- Partial process specifications that allow **dynamic refinement** and **adaptation** according to available system functionalities
- Modeled via **Adaptive Pervasive Flow Language (APFL)** an extension of traditional workflow language (BPEL) with abstract activities + preconditions/effects

**Process Fragments:**
- Offered functionalities that can be **dynamically discovered/used** by other entities
- Modeled as business processes

**Context Model:**
- Important **characteristics of the environment** and of the entities that operate in it
- Used to define **context preconditions/effects** on process activities and **goals** on abstract activities
Context Model

- Shared representation of the **system operation environment** used to express **goals**, **preconditions** and **effects** on fragments

- Defined through a set of **context properties**, each capturing a particular aspect of the system and its evolution (modeled as state transition systems)
  - Context properties **evolve** as an effect of fragment activity execution (**normal behavior**) or as a result of exogenous changes (**improbable behavior**)

- **Context configuration**: snapshot of the context at a specific time
Business Processes and Fragments

- **Adaptable Pervasive Flow Language (APFL)**
  - Extension of traditional workflow language (BPEL)
  - **Standard constructs:** sync/async WS communication, data manipulation, complex control flow constructs
  - **Preconditions** on activities
    - Constrain the activity execution to specific context configurations
    - Used to catch violations and trigger adaptation
  - **Effects** on activities
    - Model the expected impact of activity execution on the context
    - Used to perform automatic reasoning on activities execution
  - **Compensation** goals on activities
    - Model the context configuration that has to be reached in case a (successfully executed) activity needs to be compensated
    - Used to (partially/completely) roll-back a process instance
  - **Abstract activities** annotated with **goals**
    - Defined at design-time in terms of the goal (context configurations) it needs to achieve
    - Refined at run-time into an executable process wrt the available fragments, the current context configuration and the goal to be achieved
Standard constructs used for:
- sync/async WS communication,
- data manipulation,
- complex control flow constructs

Business Processes and Fragments
Adaptable Pervasive Flow Language (APFL)

LANDING MANAGER

Process Fragments
Landing
Gate1 Booked
Gate2 Booked
Land Request
Departure
Departure Request
Departure Reply

Business Process
Land Request
Check Gate Avail
Prepare Gate1
Gate1 Booked
Prepare Gate2
Gate2 Booked

SHIP

Context Properties

SHIP

LANDING MANAGER

GATE1

Prepare Land S
Prepare Land L
Land Guidance
Prepare Dep S
Business Processes and Fragments
Adaptable Pervasive Flow Language (APFL)

Standard constructs used for
- sync/async WS communication,
- data manipulation,
- complex control flow constructs

+ Precondition/Effects used for
- detecting run-time violations
- automatic reasoning for adaptation

Process Fragments

LANDING MANAGER

P: gate1_status = ready
E: gate1_status = assigned

P: ship_loc = harbor & ship_status = registered

Business Process

GATE1

Context Properties

SHIP
Business Processes and Fragments
Adaptable Pervasive Flow Language (APFL)

**Standard constructs** used for
- sync/async WS communication,
- data manipulation,
- complex control flow constructs

**+ Precondition/Effects** used for
- detecting run-time violations
- automatic reasoning for adaptation

**LANDING MANAGER**

**Process Fragments**

LANDING MANAGER: *P, E*

- **Land Request**
- **Gate1 Booked**
- **Gate2 Booked**

**Departure**

- **Departure Request**
- **Departure Reply**

**G: gate1_status = ready**

**Business Process**

**+ Abstract activities / Goals** used for
- run-time context-aware service composition

**SHIP**

- **Land**
- **Leave**

**Context Properties**

**GATE1**

- **Prepare Land S**
- **Prepare Land L**
- **Land Guidance**
- **Prepare Dep S**

ServicesCUP 2012

ASTRO-CAdaptE: Dynamic Context-aware Adaptation for Service-based Systems
**Business Processes and Fragments**

Adaptable Pervasive Flow Language (APFL)

**Standard constructs** used for:
- sync/async WS communication,
- data manipulation,
- complex control flow constructs

**Precondition/Effects** used for:
- detecting run-time violations
- automatic reasoning for adaptation

**Process Fragments**

**LANDING MANAGER**

- **G**: gate1_status = unbooked

**Business Process**

- **Abstract activities / Goals** used for:
  - run-time context-aware service composition

- **Compensation Goals** used for:
  - roll-back executed activities
Adaptation Mechanisms and Strategies

- Adaptation needs
  - Need for **refining an abstract activity** within a process instance
  - Violation of a precondition of an activity that is going to be executed

- Adaptation mechanisms
  - **Refinement**: dynamic refinement of abstract activity by context-aware composition of available fragments
  - **Local adaptation**: identify a fragment composition that allows to re-start a faulted process from a specific activity
  - **Compensation**: dynamically compute a compensation process for a specific activity
Adaptation Mechanisms and Strategies

G3: CarProgressStoring = yes

Car Process:
- Unload
- Unpack
- Store
- WaitOrder
- Treatment
- Delivery

REFINEMENT: PLAN FOR G3

Registration@StorageManager
- Registration Request
- Registration Reply
- C1

StorageAssignment@StorageManager
- Assign Storage Request
- Assign StorageA Reply
- Assign StorageB Reply
- Assign StorageC Reply
- GA1
- C2
- GA2
- GA3

StoreToA@StorageAreaA
- BookA
- MoveA
- StoreAndDropA

StoreToB@StorageAreaB
- BookB
- MoveB

StoreToC@StorageAreaC
- GCl
- MoveC

REFINEMENT: PLAN FOR GA1

BookStorageA@StorageAreaA
- BookTicketA Request
- BookTicketA Reply

REFINEMENT: PLAN FOR GA2

Move2StorageA@MovementManager
- Move2StorageA Request
- Move2StorageA Reply
- P3
Adaptation Mechanisms and Strategies

G3: CarProgressStoring = yes

Car Process:
- Unload
- Unpack
- Store
- WaitOrder
- Treatment
- Delivery

REFINEMENT: PLAN FOR G3

A1: CarStatus = nok

P1: CarStatus = ok and CarRegistration = no

Local Adaptation: Plan for P1
Adaptation Mechanisms and Strategies

- **Adaptation needs**
  - Need for **refining an abstract activity** within a process instance
  - Violation of a precondition of an activity that is going to be executed

- **Adaptation mechanisms**
  - **Refinement**: dynamic refinement of abstract activity by context-aware composition of available fragments
  - **Local adaptation**: identify a fragment composition that allows to re-start a faulted process from a specific activity
  - **Compensation**: dynamically compute a compensation process for a specific activity

- **Adaptation strategies**
  - **Combine adaptation mechanisms** to solve complex adaptation problems
    - E.g., Re-refinement, Backward adaptation
  - **Search for alternative solutions**
    - E.g., Local on current activity -> Backward on current refinement -> Re-refinement -> ...
  - **One-shot vs incremental** adaptation
Implementation: ASTRO-CAptEvo architecture
Adaptation as AI Planning Problem

M_{\text{adapt}} \text{ composition of fragments that, if executed from the current configuration and in the absence of exogenous events, ensures that the resulting context configuration satisfies G.}
Adaptation as AI Planning Problem

GOAL BUILDER

APFL2STS, CM2STS

Transformation of fragments and context configuration in STSs and removal of improbable events

GOAL BUILDER

Translation of the adaptation goal in EAGLE planning goal
Adaptation as AI Planning Problem

\[ \Sigma_{\parallel} \quad \text{Product of fragment and context STSs synchronized on preconditions and effects} \]
Adaptation as AI Planning Problem

Adaptation Goal

Context Configuration

Process Fragments

PLANNER sophisticated AI planning techniques for WS composition developed within the ASTRO project (non-determinism, extended goals, data flow requirements)

(2002 – Today)

Adaptation Process
Adaptation as AI Planning Problem

Adaptation Goal

Context Configuration

Process Fragments

STS2APFL Translation of the synthesized plan into an APFL executable adaptation process
We implemented the framework and evaluated it on the CLS scenario.

Demonstrator ASTRO-CAptEvo:
- simulate the CLS scenario, view the adaptation/composition mechanisms in action, inspect the internal system operation

Implementation and Evaluation

- **Modeling Effort and Complexity**
  - 29 entity types and process models, 69 fragment models, 40 context properties
  - Refinement:
    - Modularization and re-use of different specification elements (Context Properties and Fragments)
    - Each fragment can be adjusted by its provider separately from other fragments of the application
    - Entities/fragments can join/leave the system without affecting the implementation
  - Adaptation:
    - N. different adaptation cases to define considering only car damages and stores unavailability (>100)
    - Adaptation processes correct by construction

- **Performances**
  - 2GHz, 3Gb Dual Core machine running Windows
  - Average: 0.5s for 22 context properties, 6 fragments, 8 activities
  - Max: 4.8s for 32 context properties, 16 fragments, 21 activities
Conclusions

- A framework for dynamic adaptation of context-aware SBS:
  - **Dynamic**
    - Entities can leave/join the system at run-time
    - Fragment/process models can change at run-time
  - **Adaptive**
    - Allows for partial specification of processes (abstract activities) and to leave the handling of improbable/extraordinary situations to run-time (activity preconditions)
    - Provides different adaptation mechanisms properly coordinated through strategies
  - **Context-aware**
    - Shared context model describing the system operational environment
    - Can be used to model exogenous events which may affect the system operation
  - **User-centric (on-going)**
    - Smart Mobility scenario within the SmartCampus living lab (Trento Digital City)
  - **Distributed (future work)**
    - Adaptation is done considering the “system of systems”

- ASTRO-CaptEvo Demonstrator on the CLS Scenario:
  - Simulation environment + Adaptation/Composition techniques in action
Mahalo !
References

- **Adaptation of Service-based Business Processes by Context-Aware Replanning**. A. Bucchiarone, M. Pistore, H. Raik, R. Kazhamiakin. (SOCA2011)
- **Control Flow Requirements for Automated Service Composition**. H. Raik, R. Kazhamiakin, M. Pistore, P. Bertoli, M. Paolucci and M. Wagner. (IEEE ICWS 09)
- **Automated Web Service Composition at Work: the Amazon/MPS Case Study**. A. Marconi, M. Pistore, P.Poccianti and P. Traverso. (IEEE ICWS 07)
- **Specifying Data-Flow Requirements for the Automated Composition of Web Services**. A. Marconi, M. Pistore and P. Traverso. (IEEE SEFM 06)
- **Automated Synthesis of Composite BPEL4WS Web Services**. M. Pistore, P. Traverso, P. Bertoli and A.Marconi. (IEEE ICWS 05)
- **Automated Composition of Web Services by Planning at the Knowledge Level**. M. Pistore, A. Marconi, P. Traverso and P. Bertoli. (IJCAI 05)
SmartCampus: Smart Mobility Services

Transport services and facilities in Trento area.

- Driving/Walking/Buses
  - Directions + time info
  - Info buying tickets
  - Bus schedule, delays

- Taxi
  - Booking via SMS
  - Special service for disabled

- Trains
  - Schedule info on costs, info on delays

- Parking (free/public/private))
  - Location, cost and closure info
  - Availability
  - SMS Payment

- Car sharing
  - Pick up locations, cost info
  - Booking

- Car pooling
  - Offer/search ride

- Bike sharing
  - Pick-up points
  - Availability
SmartCampus Platform: Architecture

- **Services and facilities provided by the territory (WS, HTML/PhP, Java API, SMS-based, e-mail based, .. ).**
- **Techniques for monitoring the expected behavior of the service platform.**
- **Technology- and provider-independent wrapping of territorial services.**
- **Formal representation of domain concepts and expected behavior.**
- **Techniques for dynamic user-driven composition/adaptation of service-based applications.**
- **HCI technologies for citizen-centric service delivery.**
- **Access control and privacy models and techniques, systematic security analysis for identification of weaknesses.**

**Domain Models**

- **Monitoring**
- **Customization & Adaptation**
- **Interaction/Presentation Models**
- **Security & Trust**

**Service Wrapping**

**HCI**

**Marco**

**Sara**

**John**