17th October 2010, DSM Workshop 2010 @SPLASH

Experiences with Automotive Service Modeling

Akihito Iwai, DENSO CORPORATION

Norio Oohashi, NEC Corporation

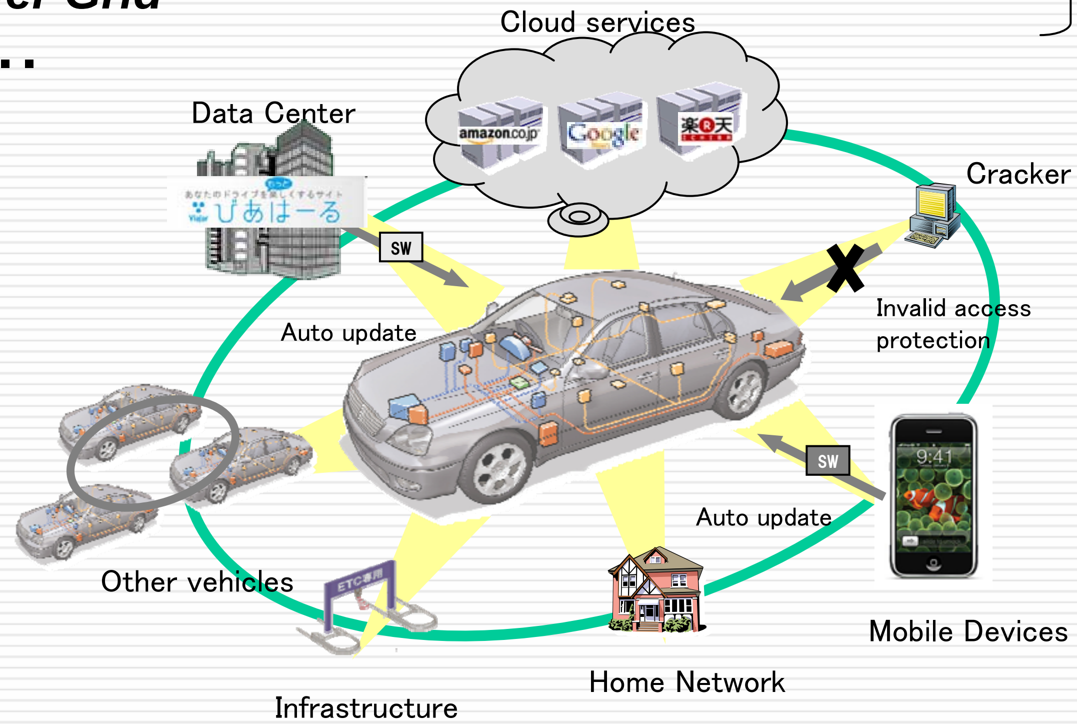
Steven Kelly, MetaCase

Background: Service Integrated Systems

- Automotive Software Systems become more large-scale and complex year by year
- ⇒ Evolving to “***Service Integrated System***” as “***Connected Vehicle***”
  - Vehicle + “***Traffic Infrastructure”***
  - + “***IT service***”
  - + “***Power Grid***”
  - ・・・・・・・・

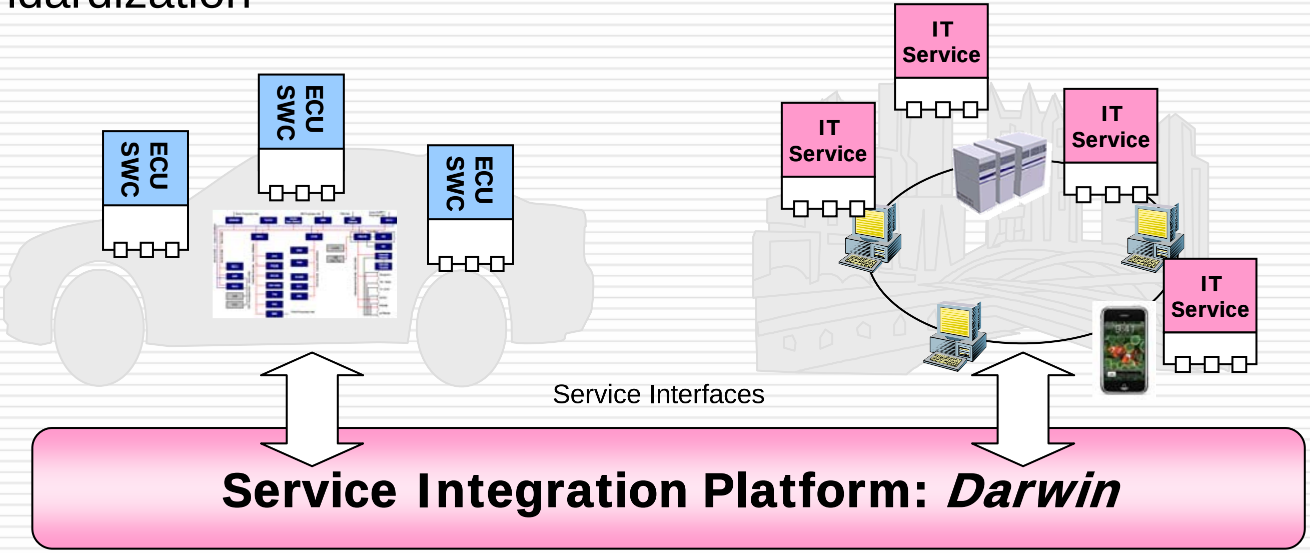
ITS, Telematics

Smart Grid



Motivation: Need for a Service Integration Platform

- **Big GAP** between vehicle world and IT world!
  - Static vs Dynamic, Quality vs Speed, ….
- Current Approach
  - Component-based approach
- Further Approach
  - ***Service-oriented*** approach
  - Service modeling
  - Standardization



Service Integration Platform: *Darwin*

DENSO

Challenges for Service Integration

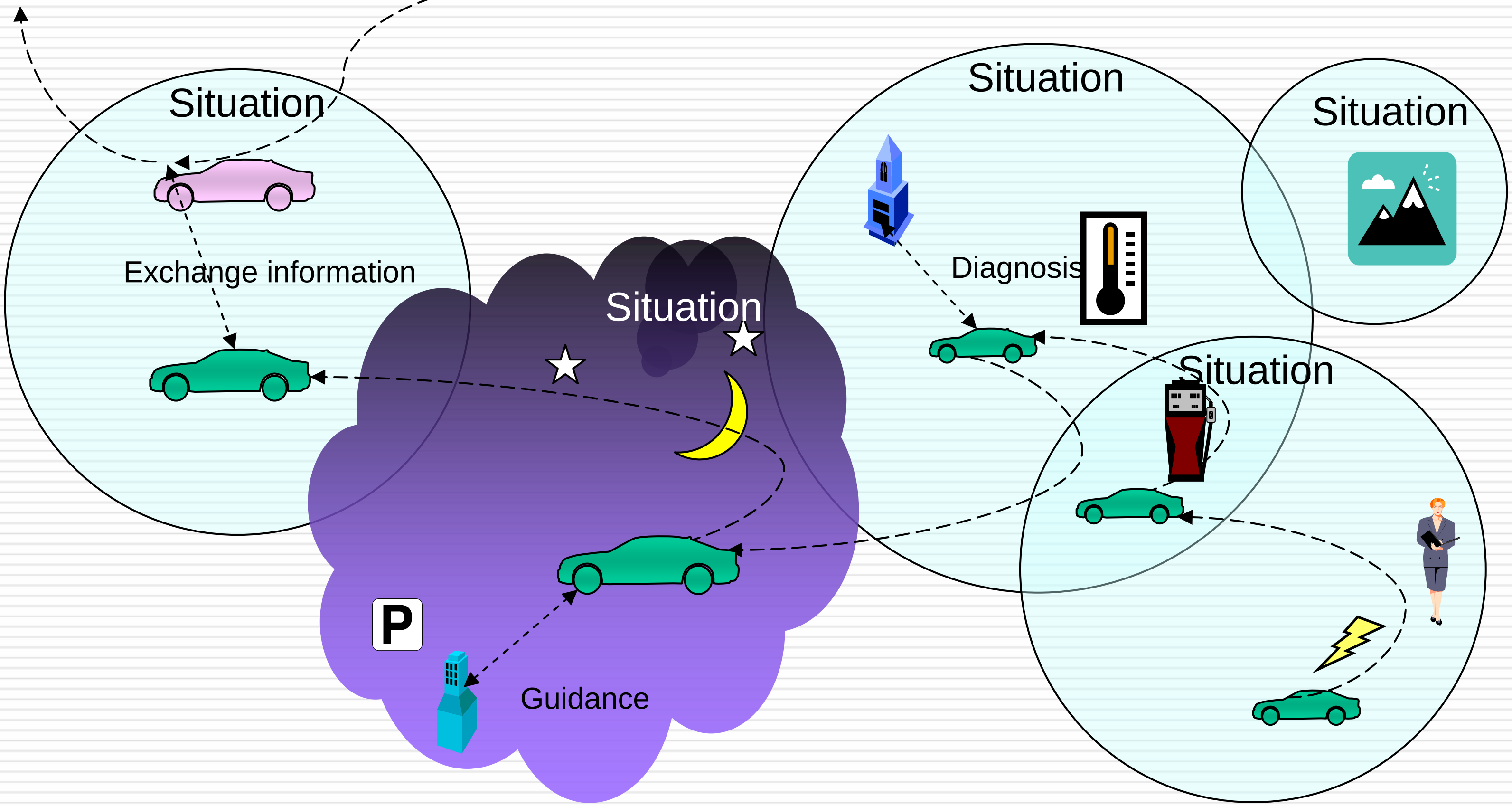
- **Service Modeling *(Today's topic)***
  - Service model definition and implementation
  - Abstract model of vehicle service
  - Capturing requirements from *multiple stakeholders*
  - Developed by *multiple vendors*
- **Secure Platform**
  - Protection mechanism against invalid external access
  - Highly dependable OS
  - Firewall

- **Pervasive Computing**
  - Adapting to dynamic change of system configuration
  - Installing ad-hoc communication system
  - Dynamic configuration

Darwin Service Concept

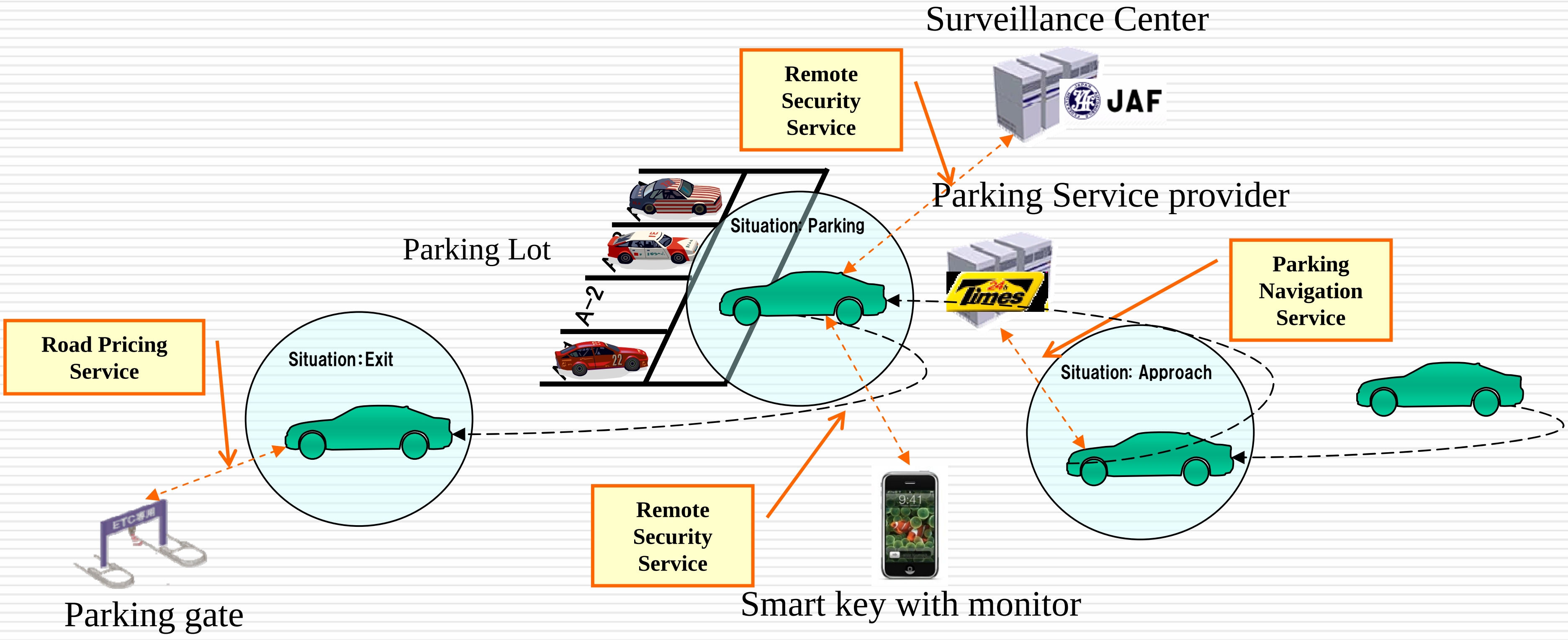
Situation = Space（Where） & Time（When）

- Situation Matching
  - Car moves through various situations.
  - Service Integration platform executes appropriate services according to the requirements of the situation autonomously.



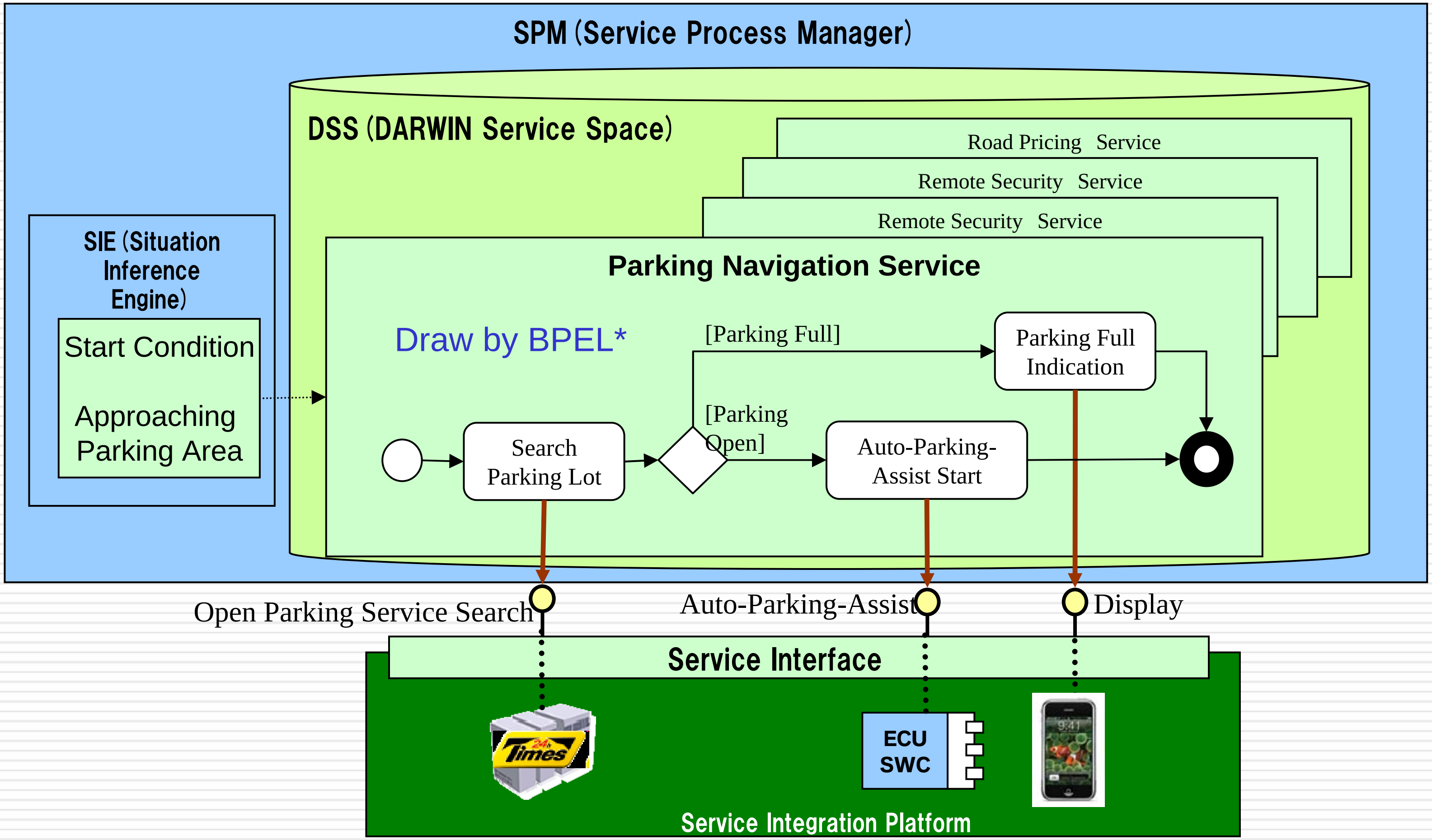
Case study: Intelligent Parking Service

- Car, service provider and mobile phone work collaboratively to provide parking navigation, remote security and road pricing.
- Car provides appropriate services according to the requirements of the situation autonomously.



Automotive service integration on service integration platform with BPEL

- Easy Integration of services using Service Interface

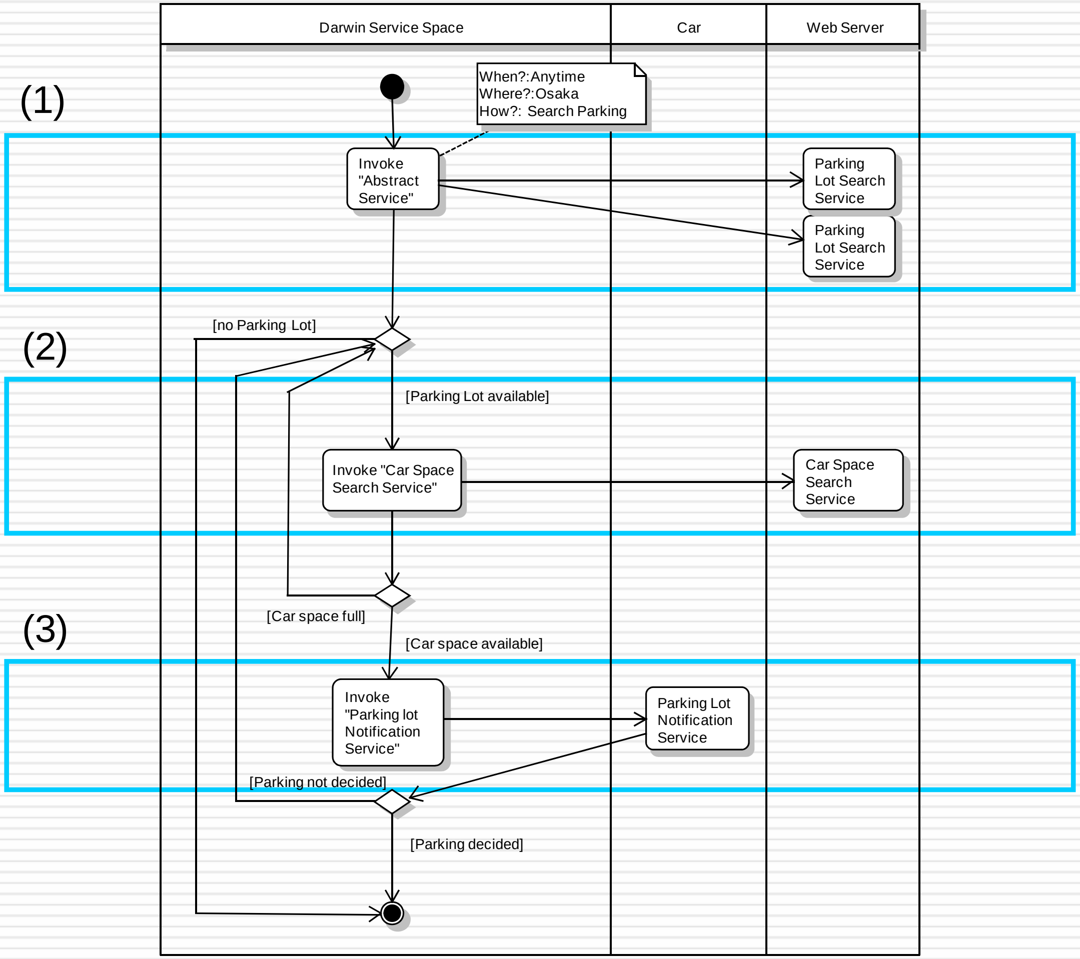


*BPEL: **Standard** for describing "*business process orchestration*" by using **XML** representation.

Experiences with BPEL

- Problems with BPEL
  - No facilities for describing the ***dependability*** of a service such as real-time guarantee, safety, reliability, and security
    - This capability is strongly required for automotive modeling
  - No model of ***resources***
    - Many of the choices we want to make in the models are based on whether resources are available
  - No native facilities for ***autonomous*** choice among multiple possible services
    - Start and end conditions had to be expressed outside of BPEL.
  - Poor facilities for ***fault tolerance***
    - e.g. modeling the behavior of a system with failures. Higher-level facilities than try-catch would be needed.
  - Poor facilities for splitting a model into ***multiple parts***, with each part only ultimately being decided at ***runtime***
    - The underlying assumption in BPEL is more that the whole model of a service is available in one place at design time.

Attempt to minimally extend BPEL



Example of BPEL description

```
(1)
<invokeAbstractService when="always" where="area:Osaka"
what="Search parking" execute="all" timing="start">
  <params>
    <param type="int">latitude</param>
    <param type="int">longitude</param>
  </params>
  <return type="string">ParkingServiceName</return>
</invokeAbstractService>
(2)
<invoke name="InvokeNotifyEmptySpaceNumber"
partnerLink="ParkingServer" operation="GetEmptySpaceNumber"
portType="GetEmptySpaceNumberPT"
inputVariable="ParkingServiceName"
outputVariable="ParkingNumber">
</invoke>
(3)
<invoke name="InvokeCheckParkingCar" partnerLink="CAR"
operation="CheckParkingCar" portType="CheckParkingCarPT"
inputVariable="ParkingNumber" outputVariable="bParkCar">
  <toParts>
    <toPart part="partnerLinkName" toVariable="ParkingServiceName"
/>
    <toPart part="partnerLinkName" toVariable="ParkingNumber" />
  </toParts>
</invoke>
```

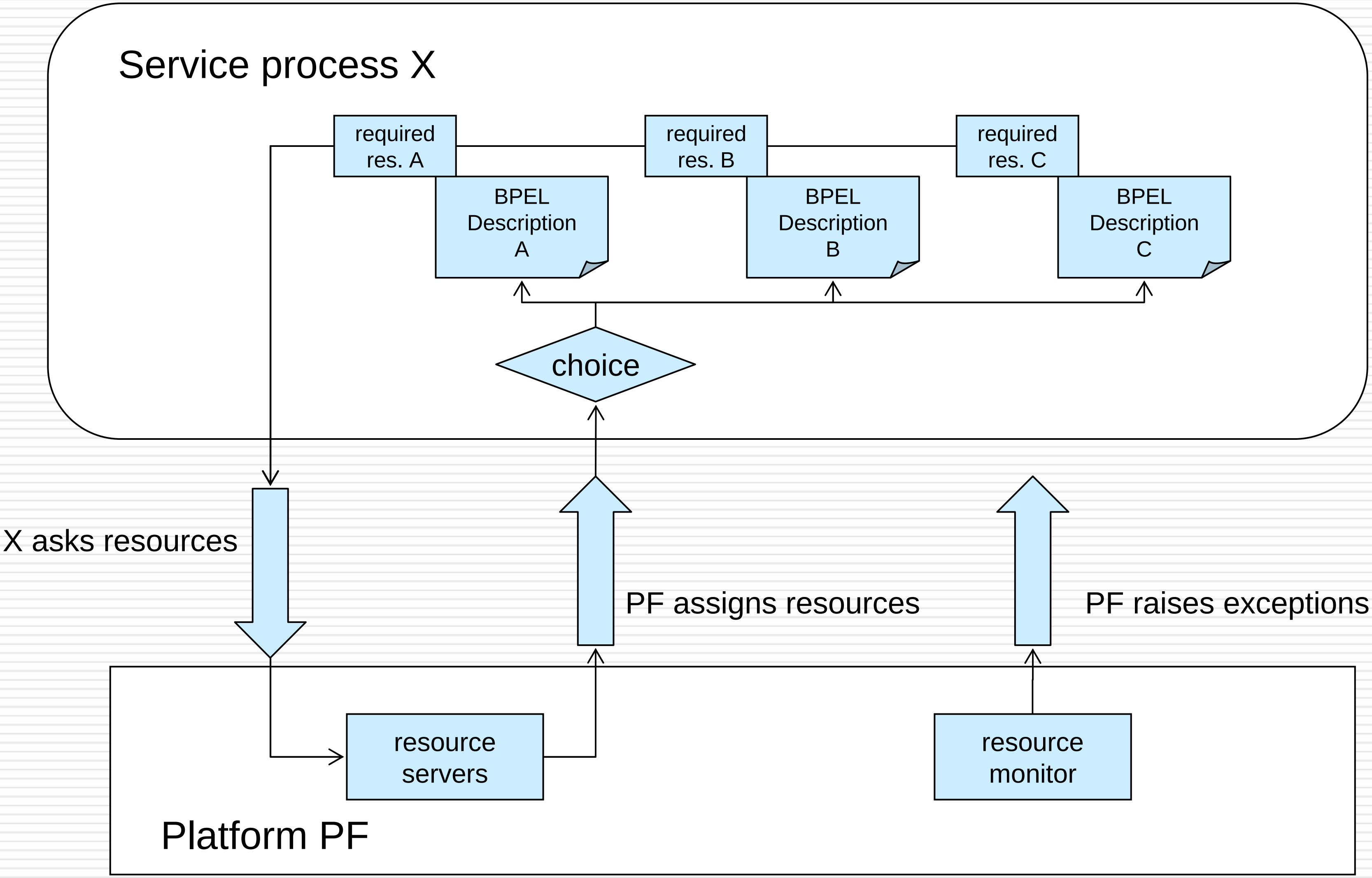
Proposed Solutions

- Domain Specific Approach will be needed
- Two possible approaches
  - Extend BPEL fundamentally (chosen as first step)
  - New DSM Language from scratch
- Proposed BPEL extension
  - **Resource Contract Function (RCF)**
    - Resource Model for choosing appropriate BPEL description along with platform resource capability
    - (See details on later slide)
  - **Fault Tolerant Network**
    - Monitoring running service, failures, degradation of service for reliability
    - No example created yet
    - (Skip in this presentation)

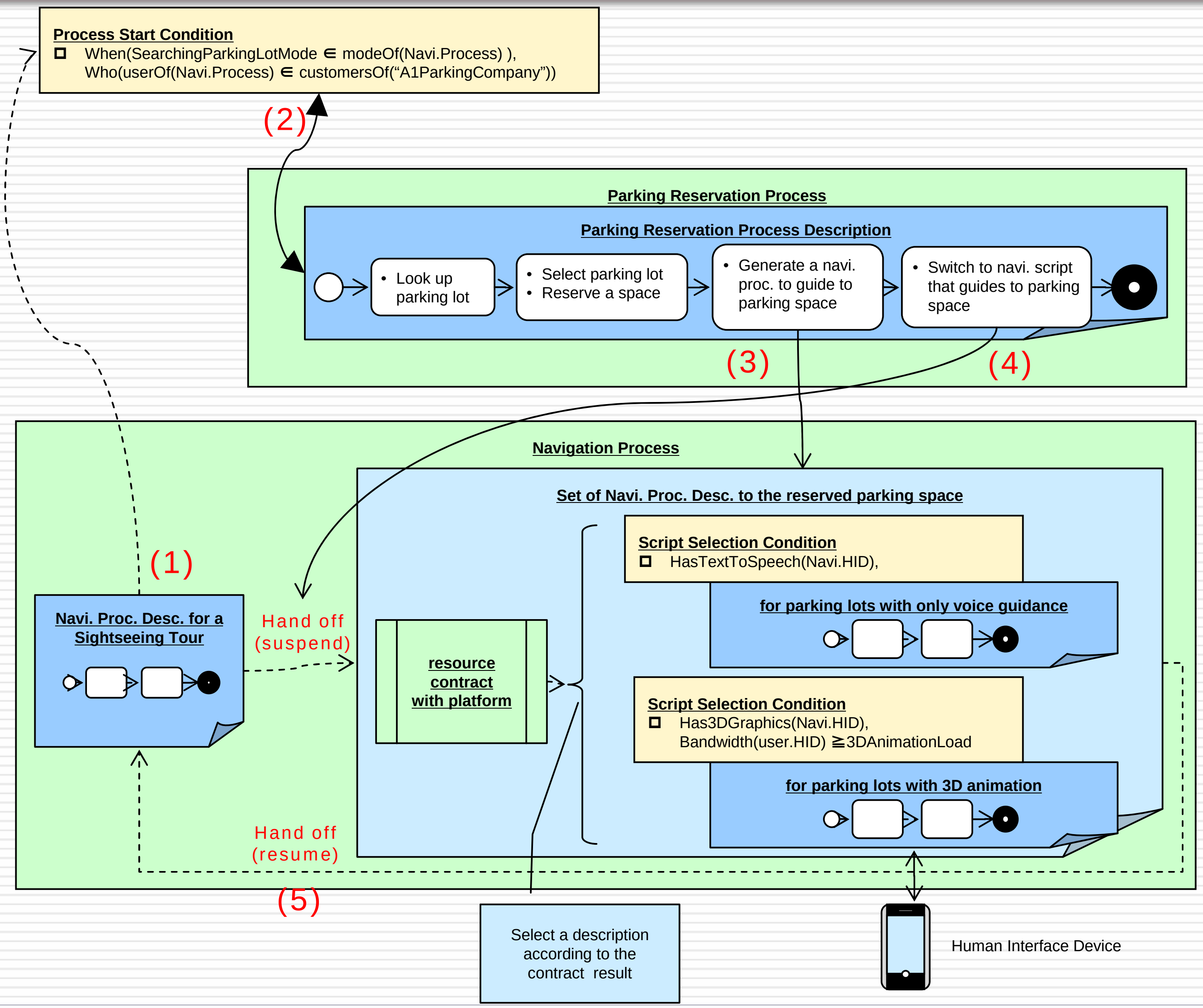
Resource Contract Function (RCF)

- Motivations
  - A service process reserves resources needed to execute, so that loading of servers and network should affect its execution.
  - If platforms could not keep resources assigned to a process, the process could switch an alternative service description.
- Example
  - When a platform cannot keep a bandwidth for a service process which uses video, the platform warns the process to degrade its quality of service: to use only voice guides, when it cannot keep the bandwidth, to use text data for text-to-speech.

Resource Contract Function (RCF) mechanism



Example of Resource Contract Function



Process Start Condition

□ When(SearchingParkingLotMode ∈ modeOf(Navi.Process)), Who(userOf(Navi.Process) ∈ customersOf("A1ParkingCompany"))

(2)

Parking Reservation Process

Parking Reservation Process Description

- Look up parking lot

- Select parking lot
- Reserve a space

- Generate a navi. proc. to guide to parking space

- Switch to navi. script that guides to parking space

(3)

(4)

Navigation Process

Set of Navi. Proc. Desc. to the reserved parking space

Script Selection Condition

□ HasTextToSpeech(Navi.HID),

for parking lots with only voice guidance

Script Selection Condition

□ Has3DGraphics(Navi.HID), Bandwidth(user.HID) ≧3DAnimationLoad

for parking lots with 3D animation

(1)

Navi. Proc. Desc. for a Sightseeing Tour

Hand off (suspend)

resource contract with platform

Hand off (resume)

(5)

Select a description according to the contract result

Human Interface Device

DENSO

Summary and Future Work

- Growing importance of approaches like SOA in automotive
  - to integrate work by multiple partners
- “One size fits all” often doesn’t fit so well
  - for our needs, BPEL could not be applied unaltered
  - main problems: low level, necessary things missing
- Altering or extending a standard stops it being a standard
  - loses its main value
  - in our case, extensions didn’t help enough anyway
- Creating a new language is a viable alternative
  - good tools allow experimentation and evolution
- Future Work
  - Virtual models of service elements in the real world
  - Models of implicit synchronization of service processes
  - Situation description models

Thank you for your attention!

Prototype DENSO Electronic Vehicle,
which commemorates the 60th anniversary of DENSO COPORPORATION