PRAXEME and UML2

Workshop

Praxeme Institute

Content

- History
- UML team internal stories, UML issues
- Interesting features
- How formal can be UML?
- Support et ressources
- Conclusion

History

ad/97-08-02: UML 1.1

Rational Software, Microsoft, Hewlett-Packard, Oracle, Sterling Software, MCI Systemhouse, Unisys, ICON Computing, IntelliCorp, i-Logix, IBM, ObjecTime, Platinum Technology, Ptech, Taskon, Reich Technologies, Softeam

UML2.0 : A wider target

- Component modeling support (EJB, .NET, etc.)
- Architecture modeling
- System engineering
- Emerging new areas : XML, EJB, SOAP, .NET, etc.
- Network & telecom, real time
- Business process modeling (activity diagrams, workflow), connexion EAI, WEB services, etc.
- Executable UML

UML2 : We want more … versions history (S Cook)

- UML 2.0 RFI (Request for Information) issued August 1999.
- RFP (Request for Proposals) issued September 2000.

Meetings, meetings, meetings …

- UML 2.0 July 2005.
  - No machine-readable specification due to structural inconsistencies in the spec.
- V2.1.1 August 2007 == V2.1.2 November 2007
  - The first version available in machine-readable form
- V2.2 February 2009
  - Fixes bugs
- V2.3 May 2010
  - Fixes bugs
- V2.4 to be released early 2011
  - Focus on fixing interoperability bugs

Where next?

UML : A family of languages

- Union of the most used models : E/R, SM, behavioral models, MSC, Components, ADL, SDL, etc.
- Merging several viewpoints on IT & systems: Technical systems, IS, embedded systems, BPM, etc.
- Domain specific targeting using UML profiles

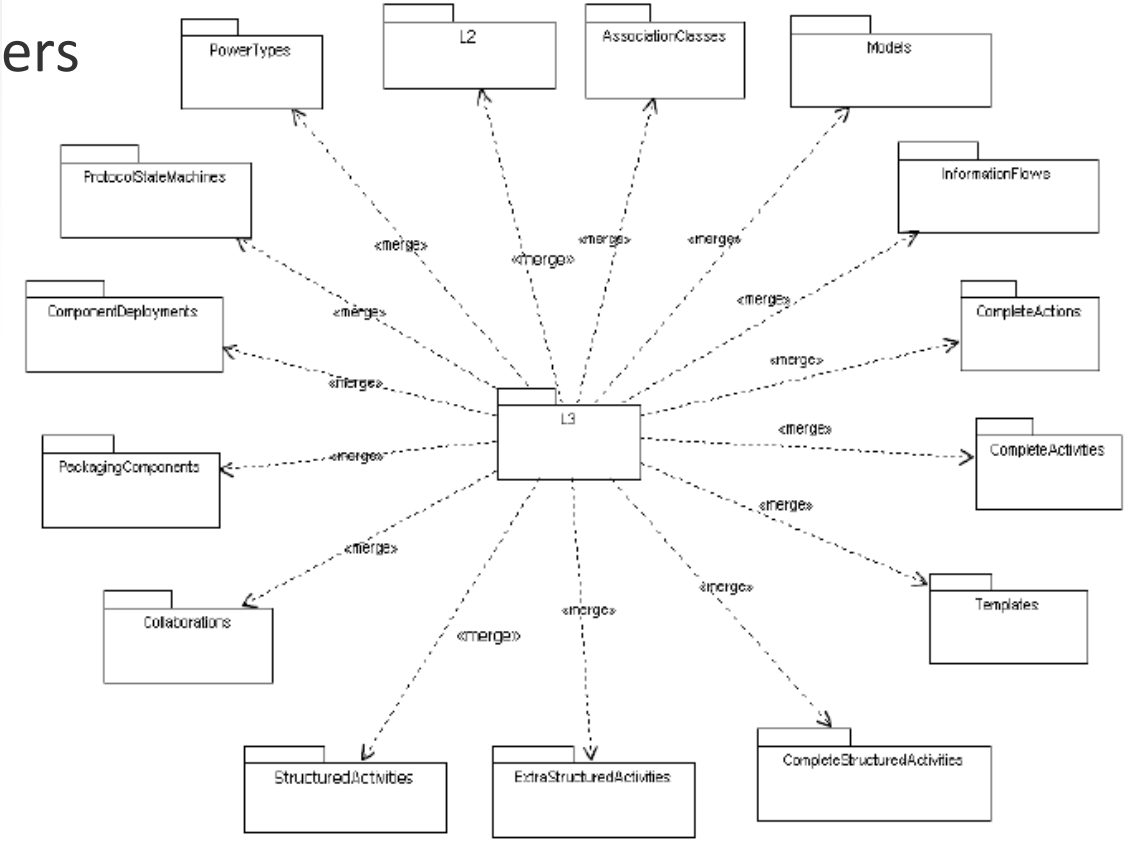
- ➔ There exist a very wide number of interpretations of UML, and representation modes of similar things.
- UML is a modeling toolbox.

OMG Stories

- UML2 : ditching ternary associations?
- The MOF/Profile fight
- Package template
- Information flows

UML metamodel construction – Package merge

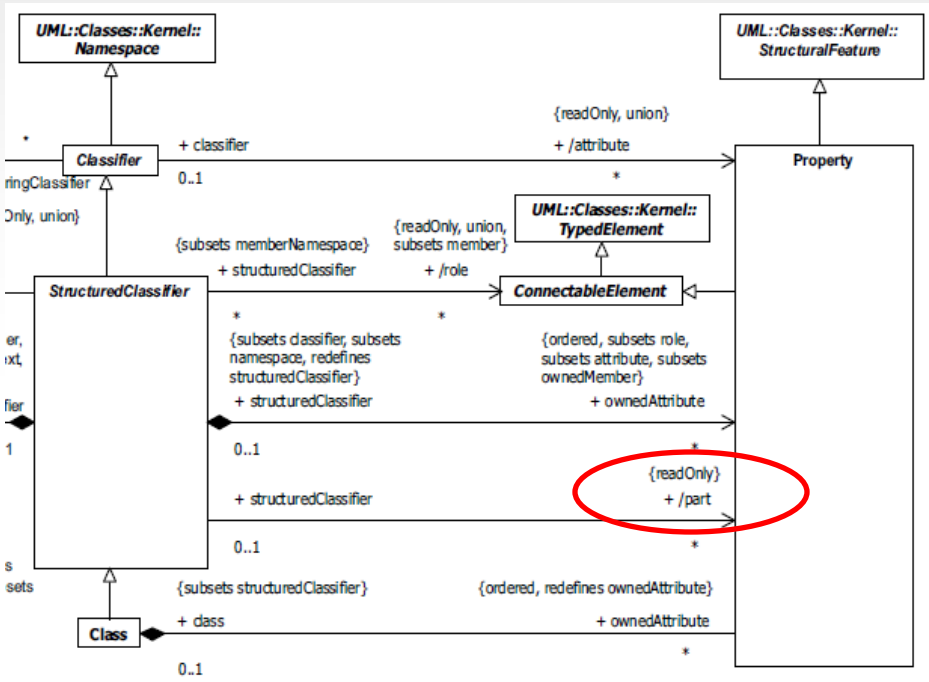
- Un-readable for a non user
- Unclear standard instanciation, HUGE XMI problems
- Never used by end users

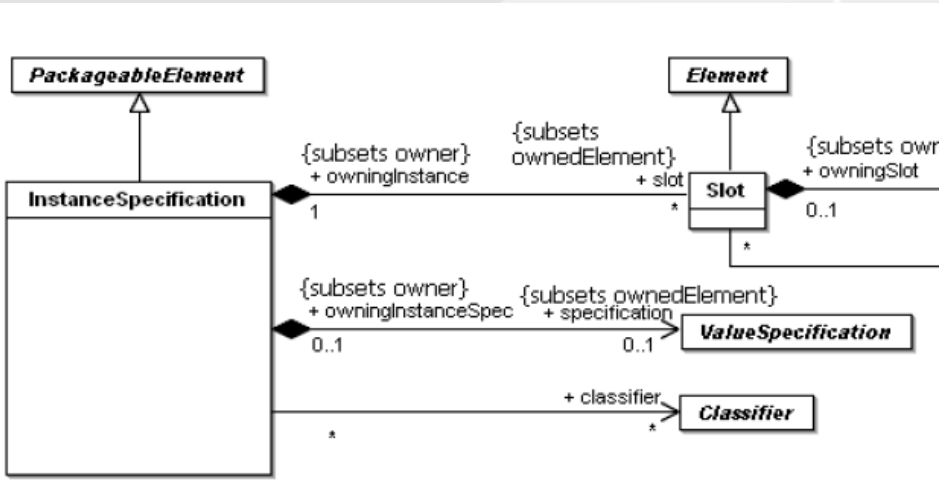


“Instance Specification” vs Part

- Platon (and Magritte) told us that we cannot really represent the reality
- An instance in a model is not … a real instance
- A part describes an occurrence within the context of its parent

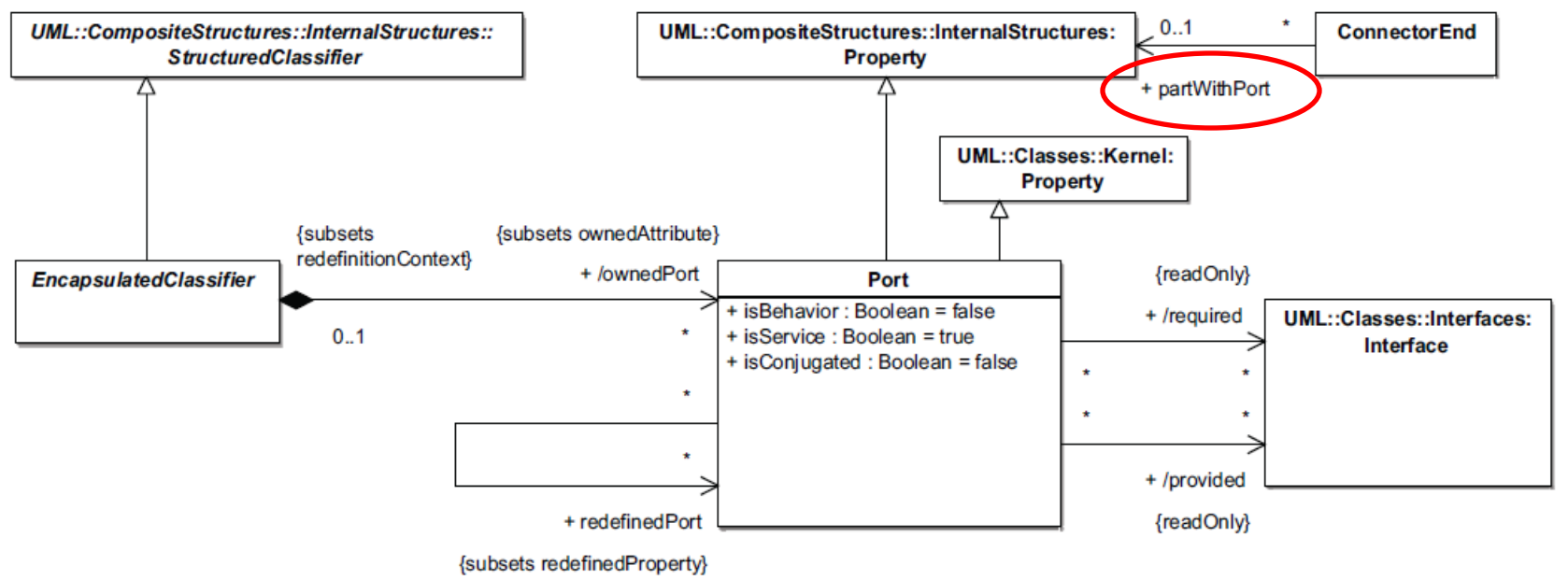
Instance specification/part = two very different metamodel constructs





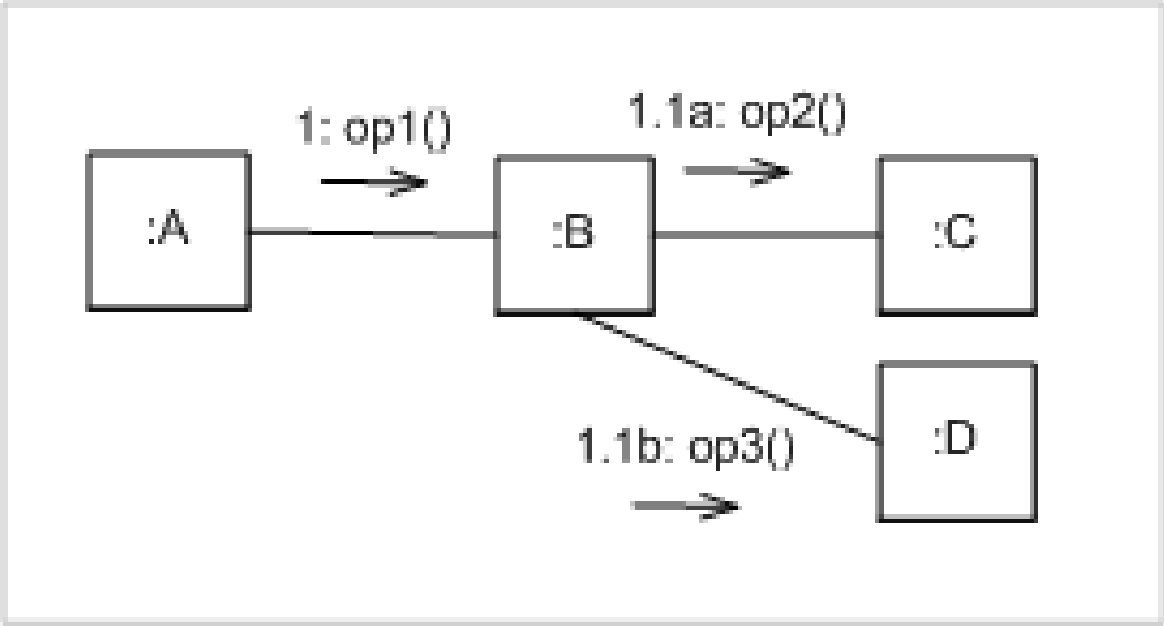
- That is an error prone issue : doubling the metamodel size, prohibiting some usages of instanceSpecification, …

The “part/port” metamodel : tricky



If a connector end references a *partWithPort, then the role must be a port that is defined by the type of the partWithPort.*

Communication diagram



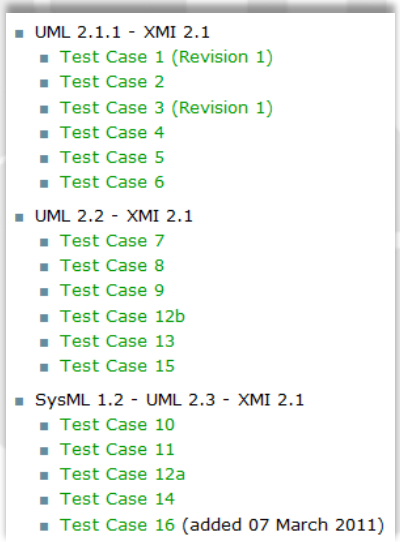
- Do you know the semantics of this syntax?

The MIWG effort

- Until now, XMI was a theoretical spec, “MOF” based, without any validation
- US Government and defense agencies have requested that model interchange works in practice
- MIWG has created UML & SysML interoperability test cases
- Peer to peer tools test (Modelio, RSA, MD, EA, Rhapsodhy, …)
- “Pivot” reference XMI

- ➔ Considerably helped to strengthen XMI and the UML spec.

- www.omgwiki.org/model-interchange/doku.php

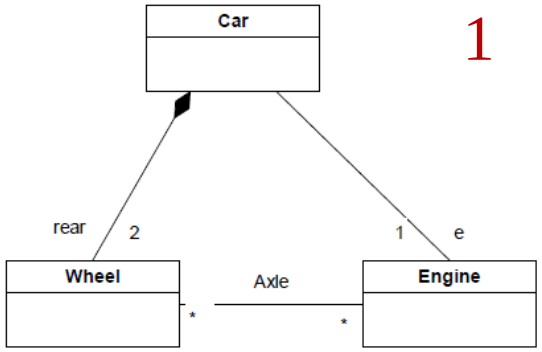


Integrating standards : UML and BPMN

- These two close standards are not integrated
  - Activity diagrams and BPMN compete
  - UML profile for BPMN
  - BPMN has no support for data modeling, but defines connections
- You can always integrate two metamodel
  - Put them aside
  - Ditch redundancies, by choosing your priority reference
- Example (demo)

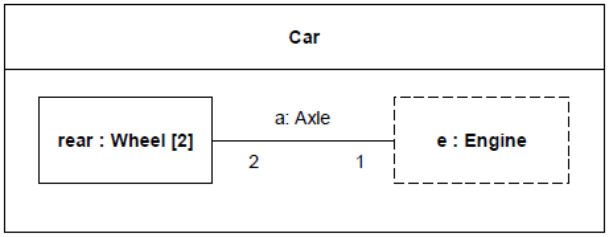
UML 2 – new features and expression means

- Internal structures : expressing the configuration of parts in the context of the embedding class

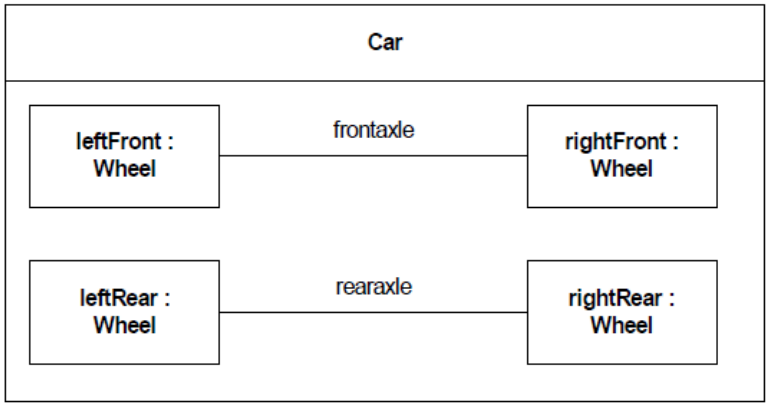


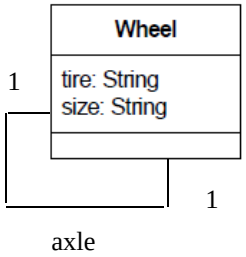
1

(i)

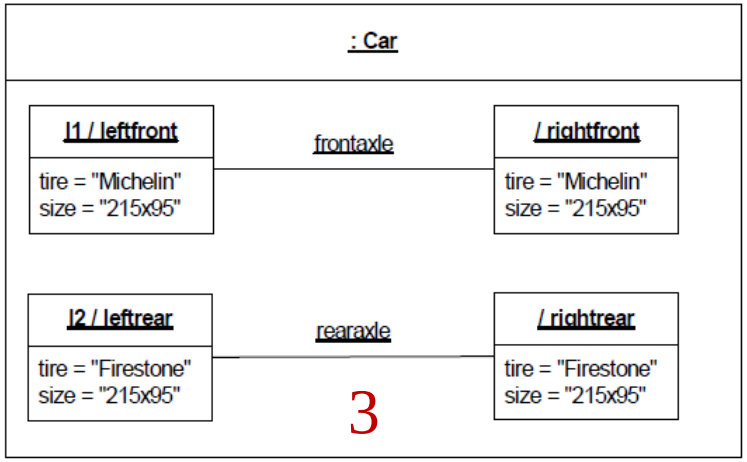


(ii)



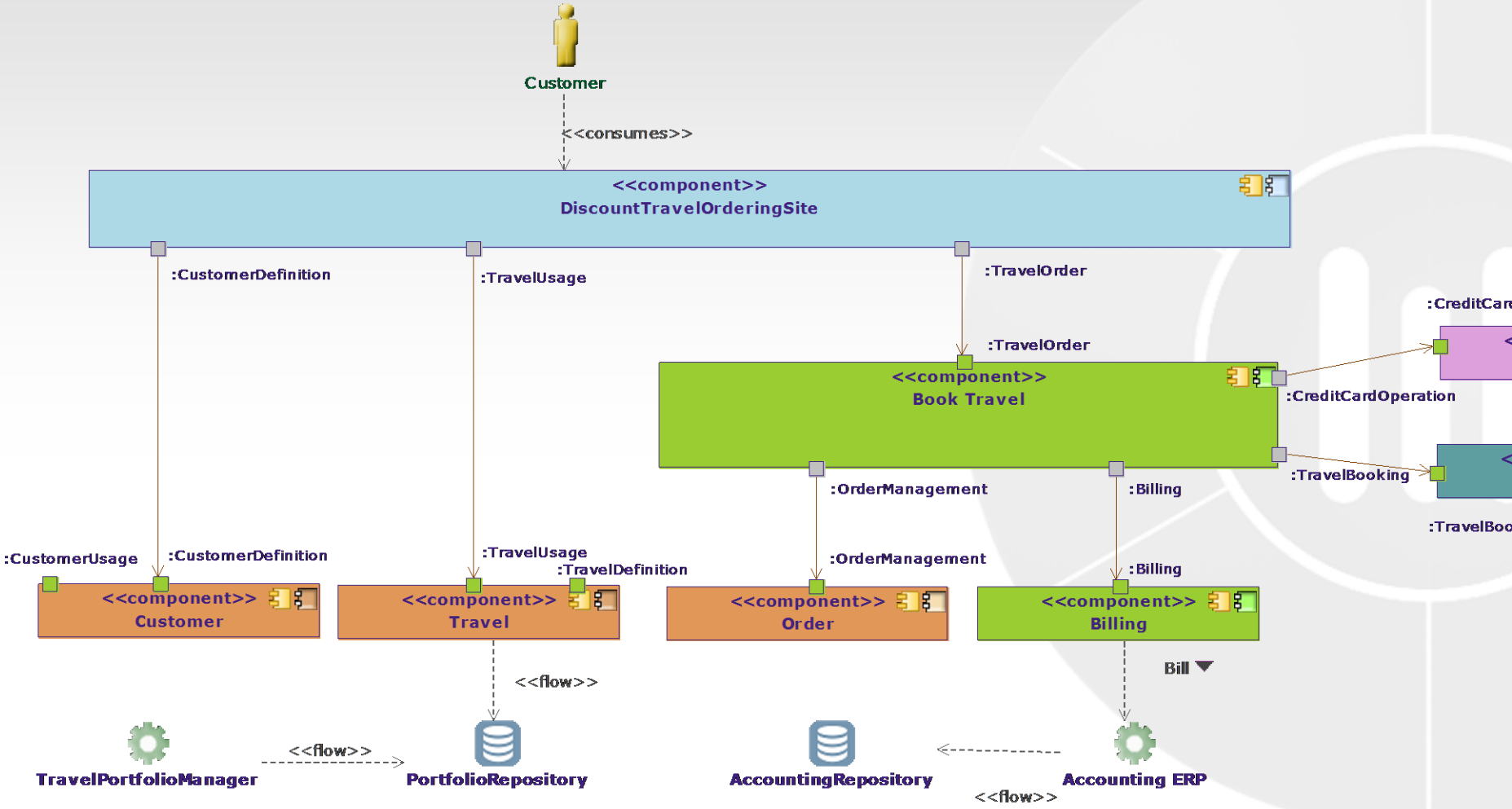


2



3

Internal structure and ports provide a great benefit for SOA



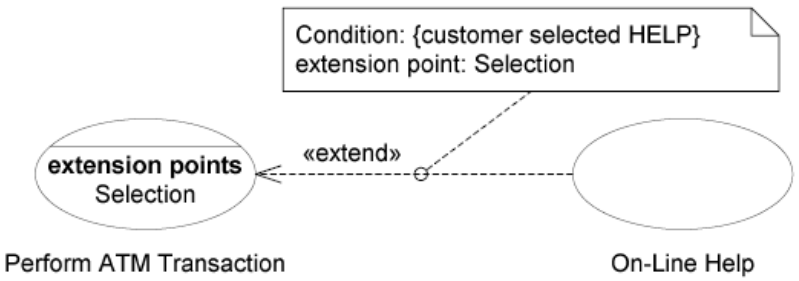
Can UML be used in a more formal way?

- Yes and No
- Yes
  - Action semantics allows the model execution/simulation
  - OCL extends UML with a formal language
  - You can always add a profile that reinforces the UML semantics for a formal purpose
- No
  - Informal parts are necessary for the preliminary phases (e.g. : Information Flows, Use Cases)
  - UML cannot embrace such a large number of domains and be accurate in its semantics
  - Programming language have different OO semantics : one unified and accurate semantics definition that works for each of them is not feasible.

UML: Semantics ambiguities (1)

Exemple des Use Case

- The concept of an “extension location” is **intentionally left underspecified** because use cases are typically specified in various idiosyncratic formats such as natural language, tables, trees, etc. Therefore, **it is not easy to capture its structure accurately or generally by a formal model**. The **intuition behind** the notion of extension location is best explained through the example of a textually described use case: …



Do you understand OCL?

(Context : Association - invariant) self.memberEnd->exists(aggregation <> Aggregation::none) implies self.memberEnd->size() = 2

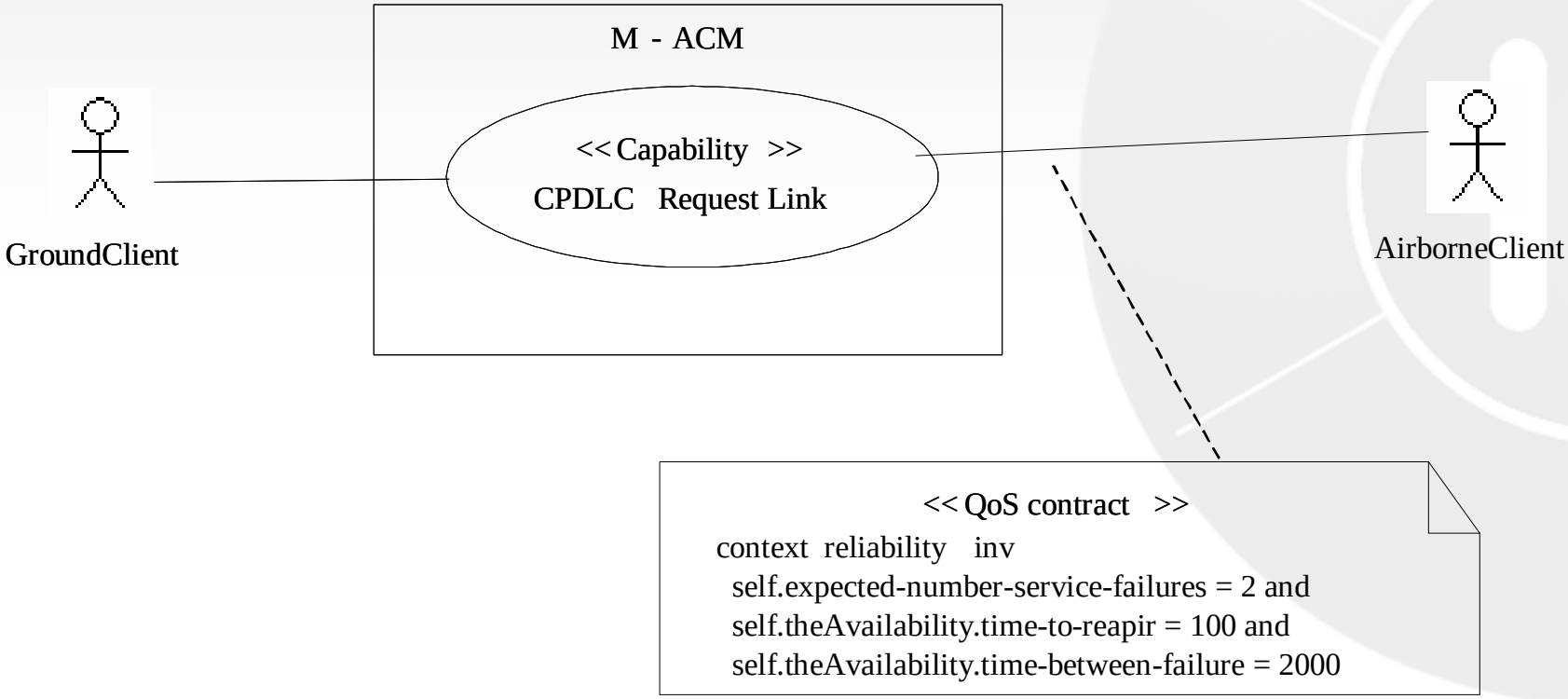
Only binary associations can be aggregations

➔ Low OCL dissemination (language adoption)

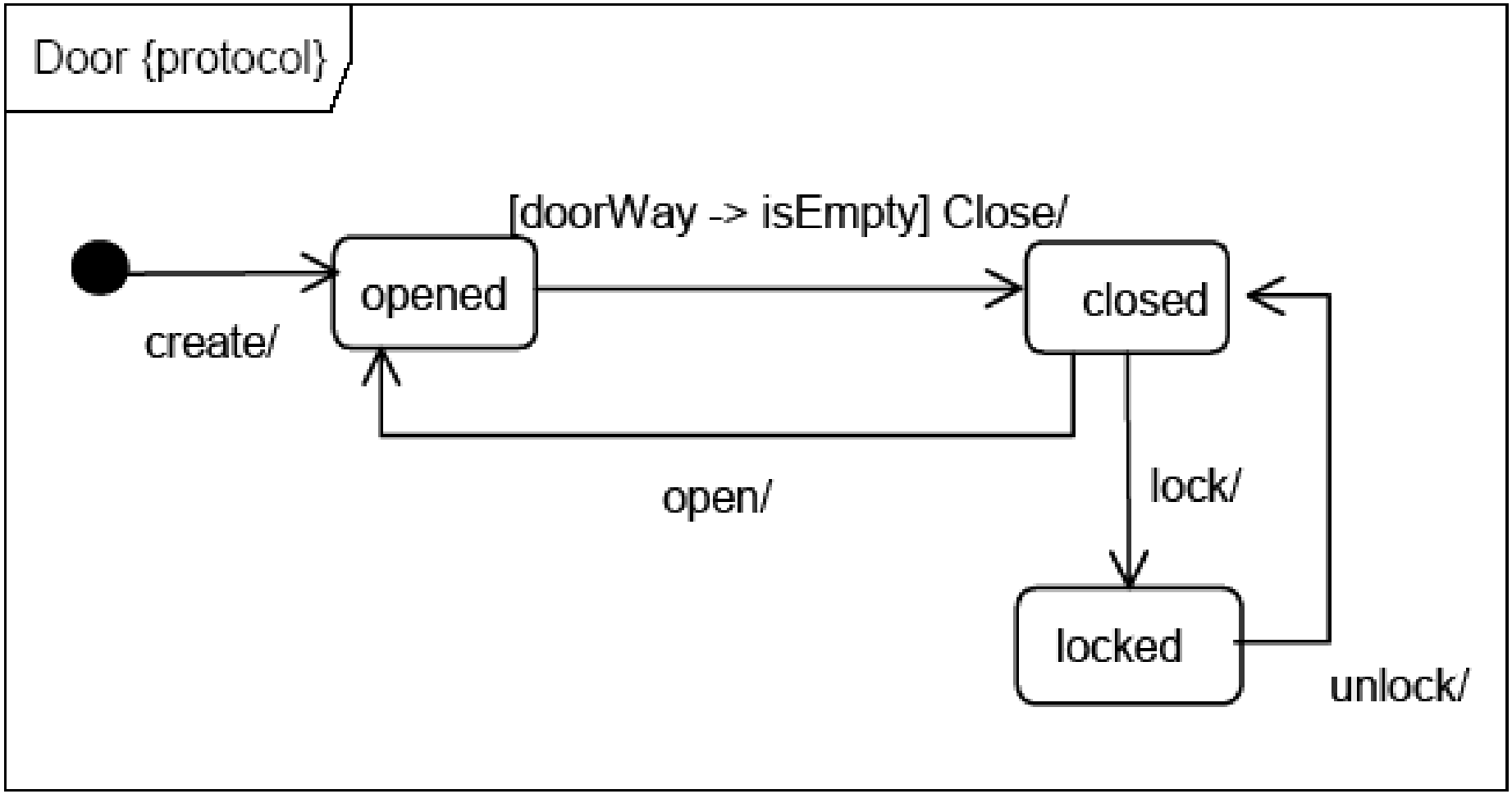
➔ OCL limited services: Evaluating assertions, Java pre/post generation, …

OCL usage example: UML Profile for QoS

OCL can model expressions that restrict the allowed values for *QoS Characteristics*. These expressions can describe some kind of relationship between characteristics (the allowed values are not independent).

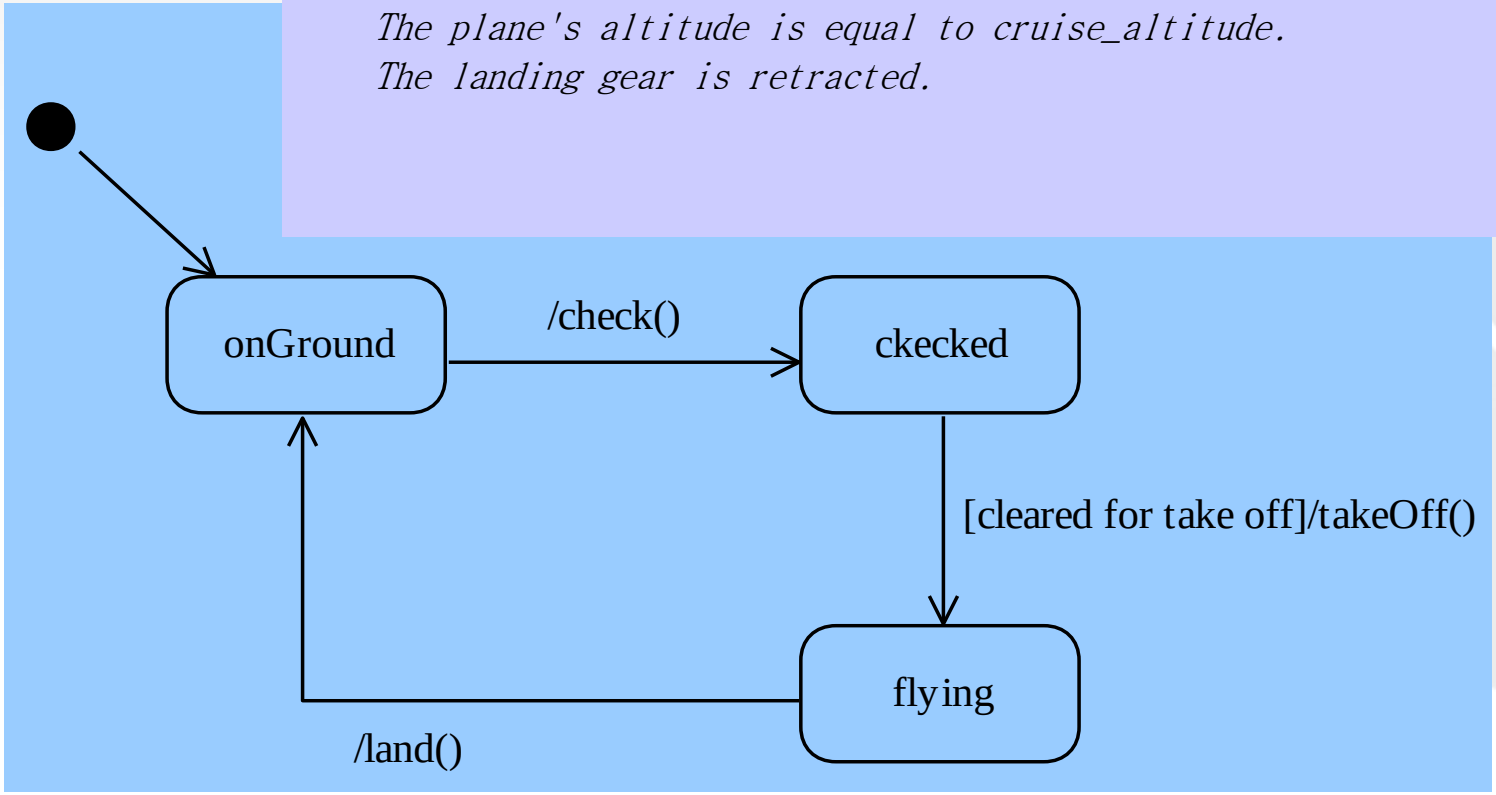


Reinforce the UML formalism : Protocol state machines

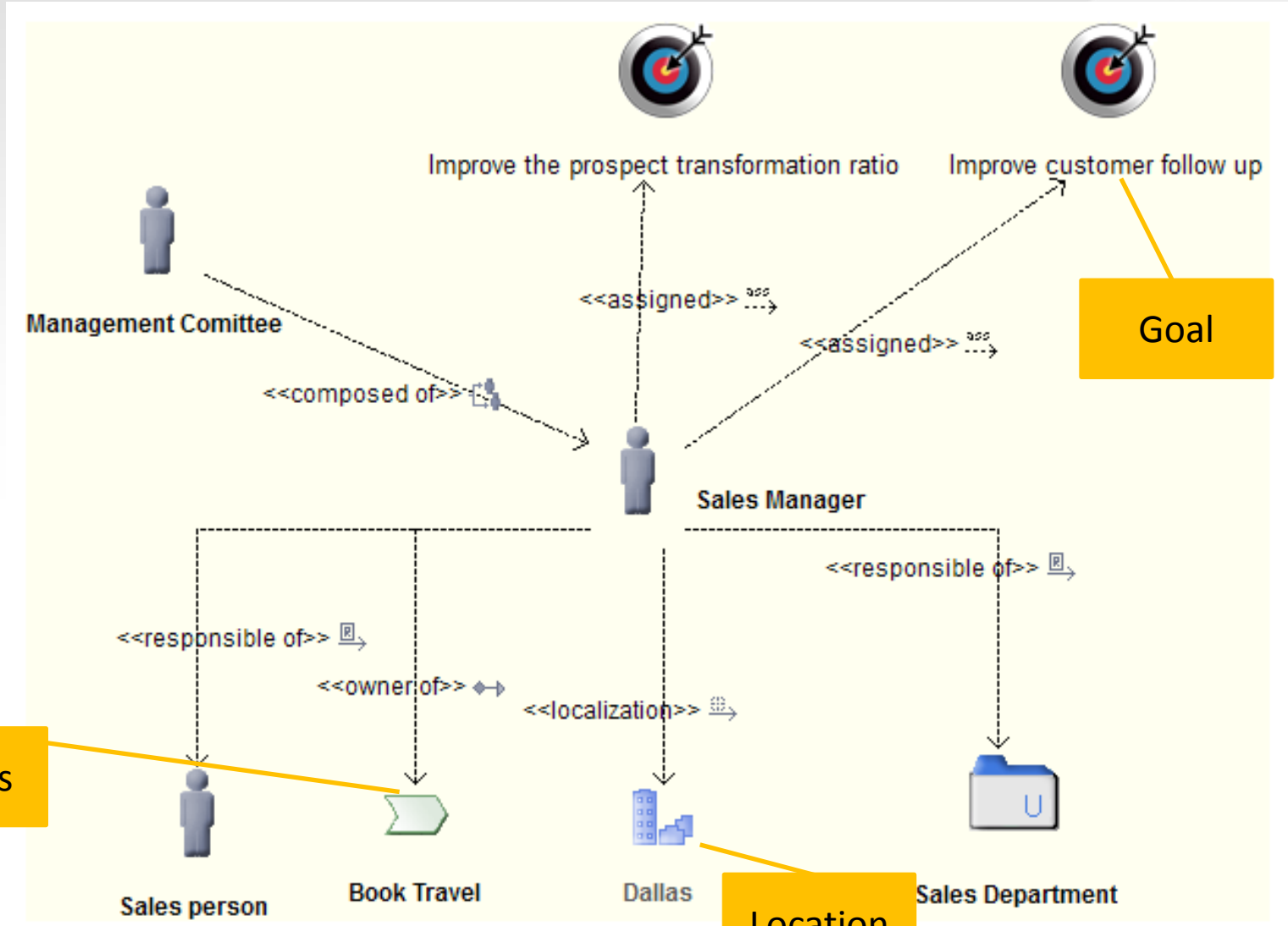


Protocol STD correspond to pre/post conditions

```
Plane: takeOff (runway_number : in integer,   cruise_altitude : in altitude)
pre
    The plane is on the ground.
    Pre-take-off checks have been conducted.
    Air traffic control has cleared it for take-off.
post
    The plane's altitude is equal to cruise_altitude.
    The landing gear is retracted.
```



UML Profiles : Examples of UML usages for Enterprise Architecture

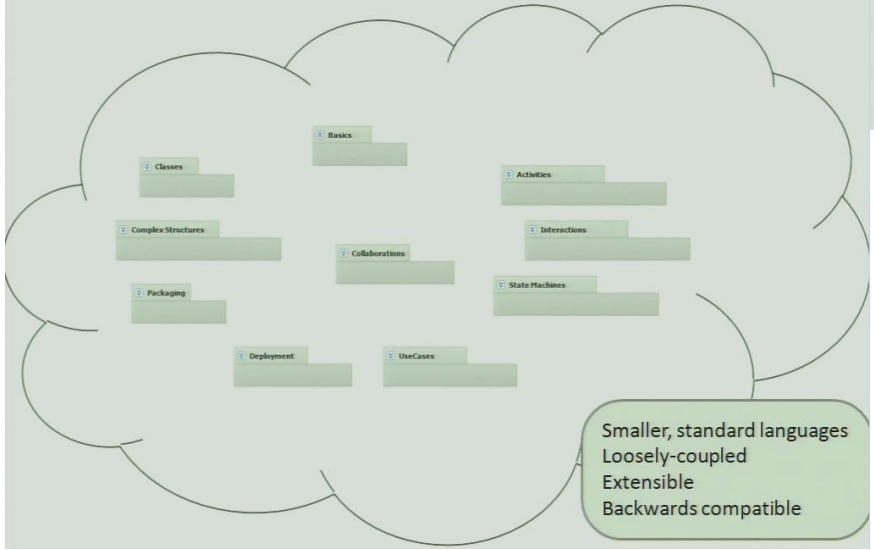


What's next

- **Simplify** and **consolidate**
- Improve **extensibility**
- Retain **compatibility**
- Define **diagram syntax**

- UML 2.4, MOF 2.4 and XMI 2.4 will be aligned to:
  - Make MOF an exact subset of UML
  - Get rid of the special cmof format
  - Enable any compliant UML tool to import/export MOF metamodels

Unbundling UML



UML roadmap

- Complete **Diagram Definition** capability.
- **UML Specification Simplification** RFP. Asks to keep the UML definition the same, but reorganize the specification so that it is consumable and manageable
  - Remove redundancy ("package merge"), consolidate, and define notation
  - Generate specification from metamodel
  - In progress now; planned completion 2011
- Improve OMG "**modelling architecture**"
  - Integrate profile mechanism and MOF using SMOF
  - Enable UML to be **refactored**, **unbundled**, and **reused**

Conclusion

- UML and BPMN are not perfect and will never be
- UML and BPMN are the modeling standards in practice
- Praxeme may benefit from some features of UML
  - Internal structures
  - UML extensions (SoaML, SysML)
- UML profile for Praxeme
  - Move it as an open source profile
  - Let the praxeme community consolidate it, so that it becomes the community tool.
- UML Tooling for Praxeme
  - Let an open source community provide Praxeme dedicated toolings : Modelio modules, macros, mode transformations, …