



QUASAR

Quantitative Assessment of UML Dynamic Models



Aline Lúcia Baroni

Advisors: Prof. Fernando Brito e Abreu

Prof. Pedro Guerreiro

New University of Lisbon
Portugal

Lisbon, September – 2005



Summary



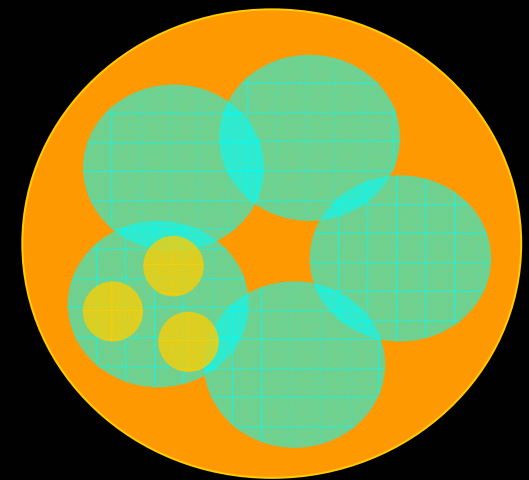
- Context
- Motivation
- Common problems x consequences
- Our solution
- One example
- Conclusions
- Where to find out more



Context: the MD* world



- MD[S]E - Model Driven [Software] Engineering
 - Model Driven Experimental Software Engineering
 - Model Driven Measurement (MDM)
 - Model Driven Estimation
 - Model Driven Quality Evaluation
 - Model Driven Refactoring
 - Model Driven Impact Analysis
 - Model Driven Evolution
- Assessment





QUASAR Research threads:



Core techniques:

- Metamodel-based evaluation
- Metrics formalization and collection automation
- Cluster analysis
- Multivariate non-linear correlation models
- Principal component analysis
- Time-series analysis

Target applications:

- Metrics-guided model refactoring
- Product quality improvement assessment
- Resource estimation modeling
- AOP migration impact
- Model evolution
- Impact analysis
- M-Level mapping (M1-M2, M2-M3) enlightening



Motivation

- Understand the existing metrics
- Help designers to make better decisions
- Cover untouched areas
- Define the metrics according to what the methodologies offer to their users
- Unfortunately...

Metrics definition and validation: problems and consequences



- **Lack of formality**
 - Metrics are informally defined
 - It may lead to misleading interpretations
- **Lack of context**
 - Entities in the definitions are not mapped to elements of a meta model or ontology → ambiguity
 - Metrics do not capture the requirements of a particular quality model → difficulties to understand and apply the metrics

Metrics definition and validation: problems and consequences



■ Lack of validation

- Simple experimentation provided, lacking replication, significant data or statistical analysis
- It hampers the proof of usefulness of the sets and their widespread acceptance

■ Lack of open tools

- How are the computations performed?
- How are the results extracted? Do the implementations really correspond to what they are supposed to be?

Metrics definition and validation: problems and consequences



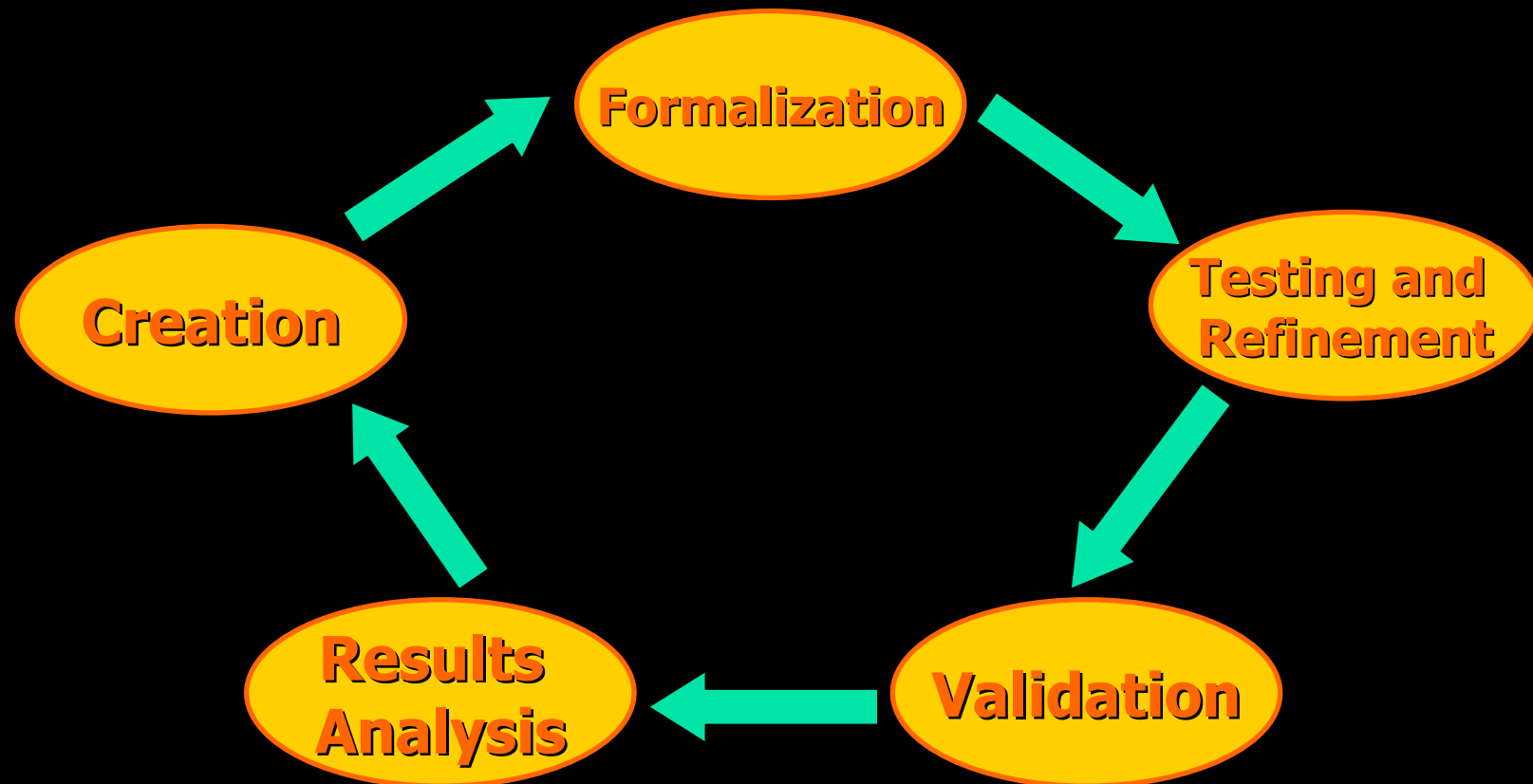
- **Lack of coverage**
 - Quality characteristics are disregarded
 - Only structural aspects are measured

Metrics definition and validation: our proposal



- **Goals** (for each kind of UML behavioral model):
 - To: ■ Classify the existing work
 - Analyze what is sensible to be measured and why
 - Compose a suite of metrics
 - Formalize and contextualize the metrics using OCL 2.0 and the UML 2.0 MM
 - Perform theoretical validations
 - Perform empirical validations
 - Generalize the conclusions
 - Make the results public

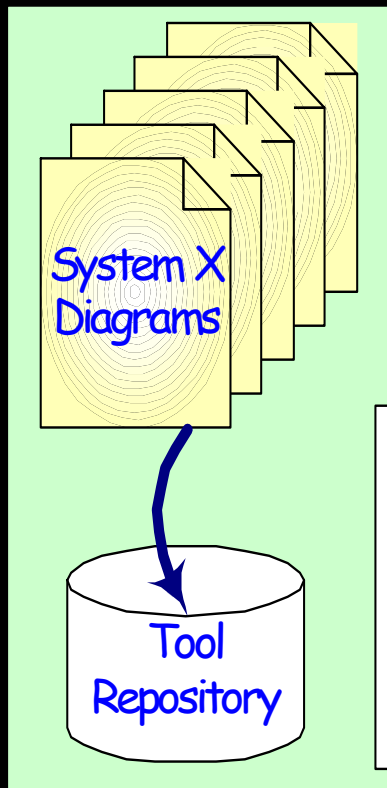
Metrics definition and validation: our proposal



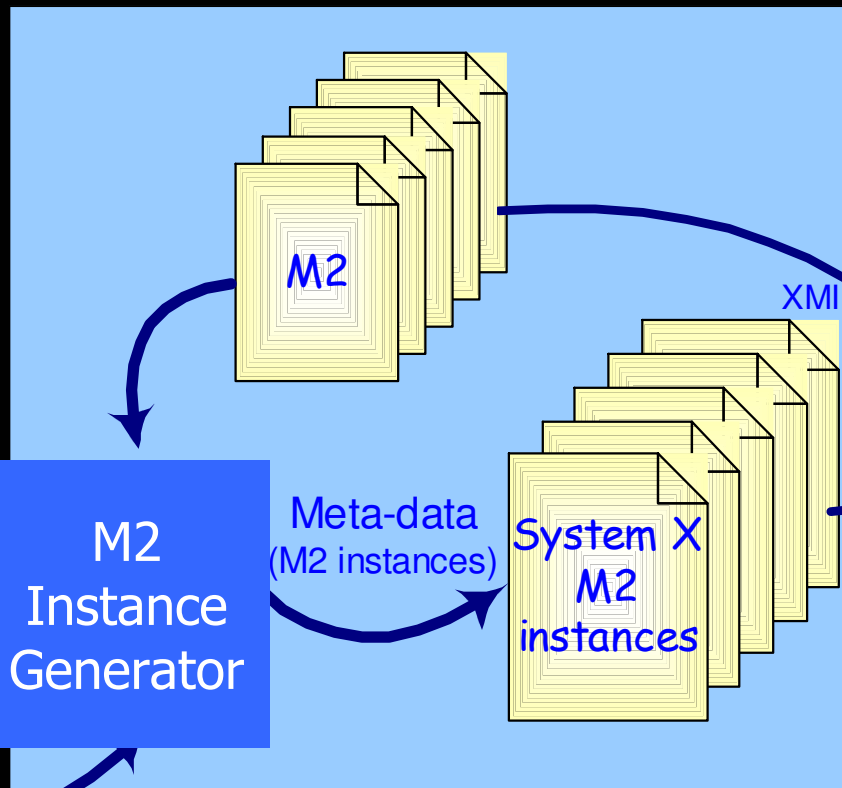
MDM Framework



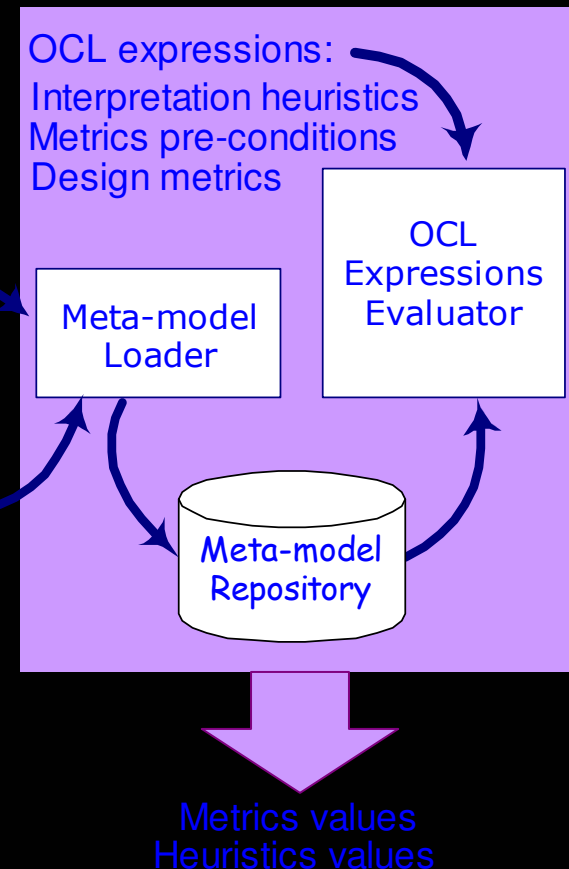
M1 Modeling Tool



M2 Instance Generator Tool



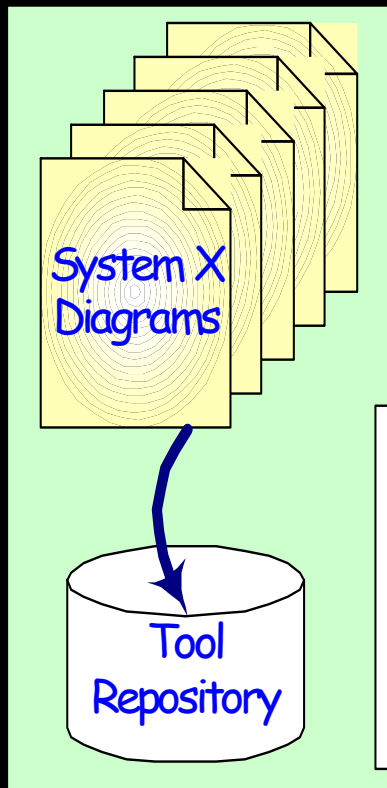
MDM Tool



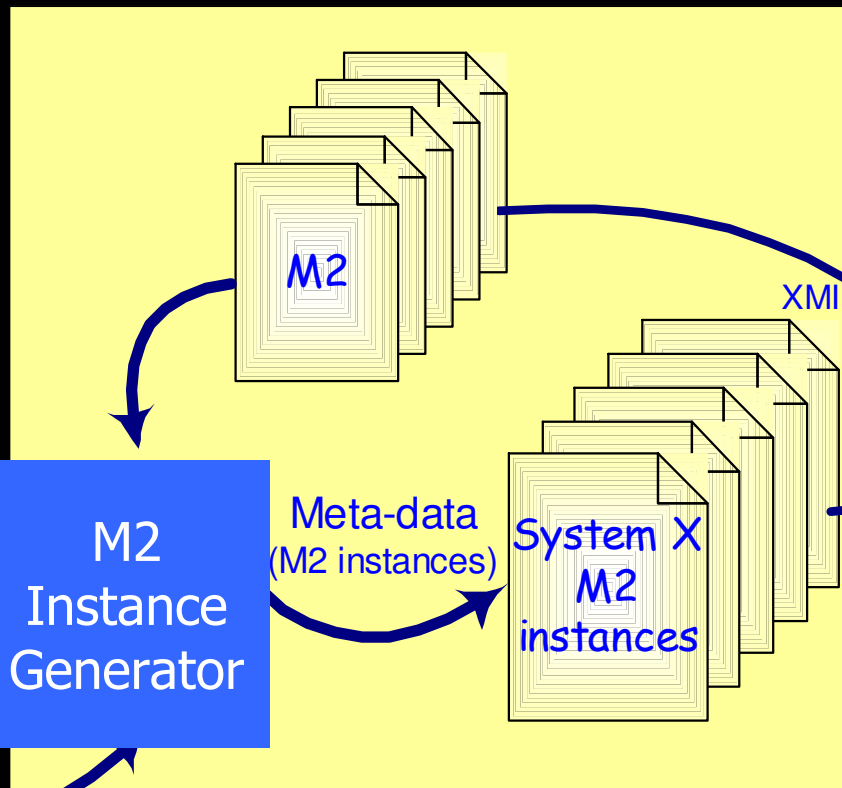
MDM Framework



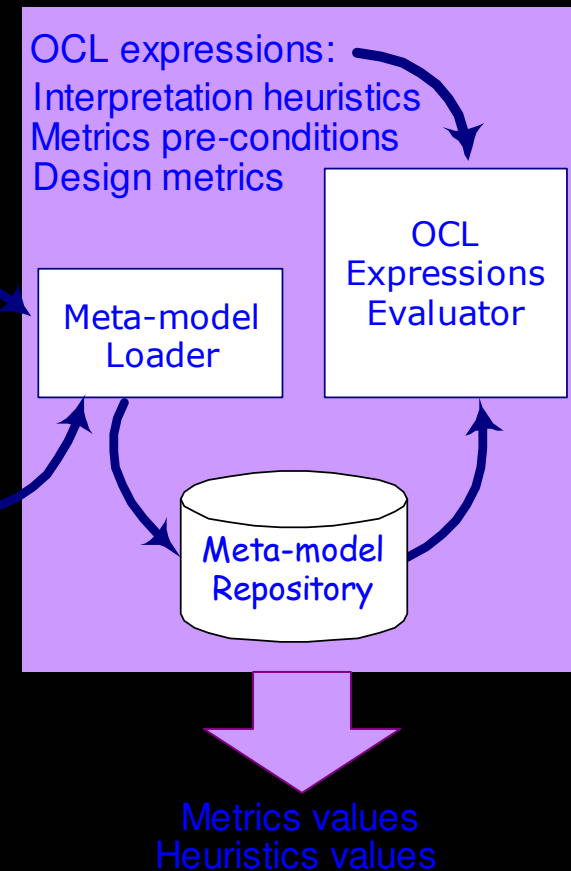
M1 Modeling Tool



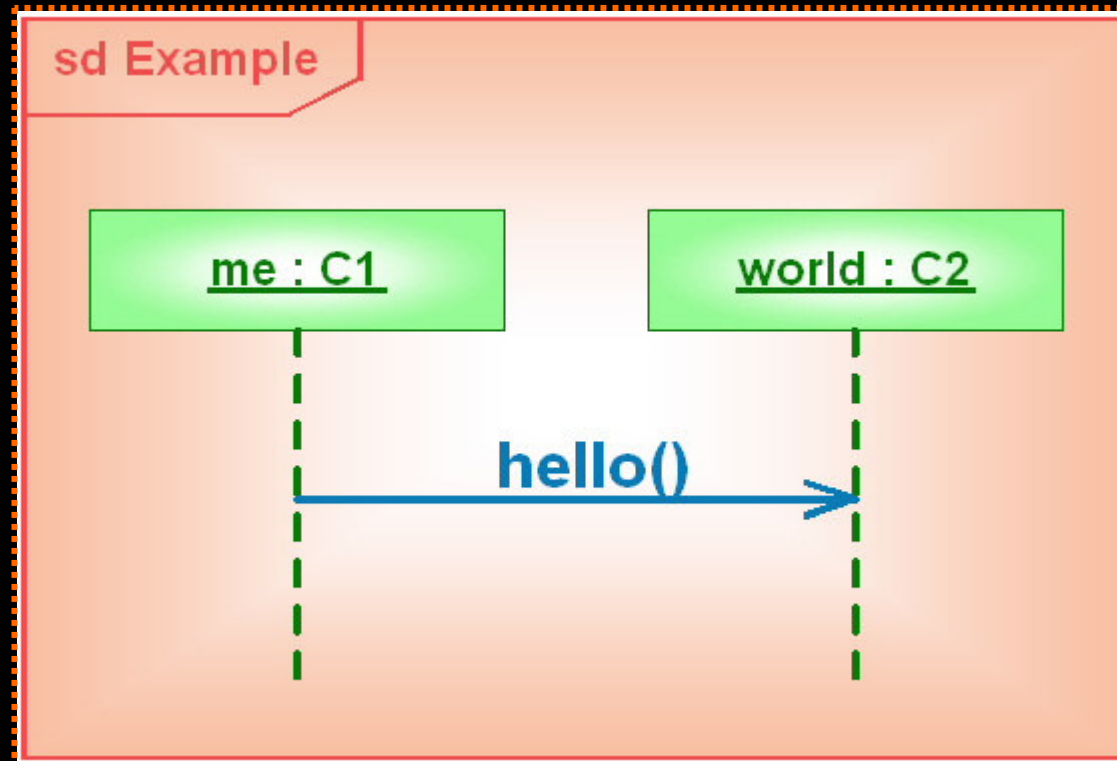
M2 Instance Generator Tool



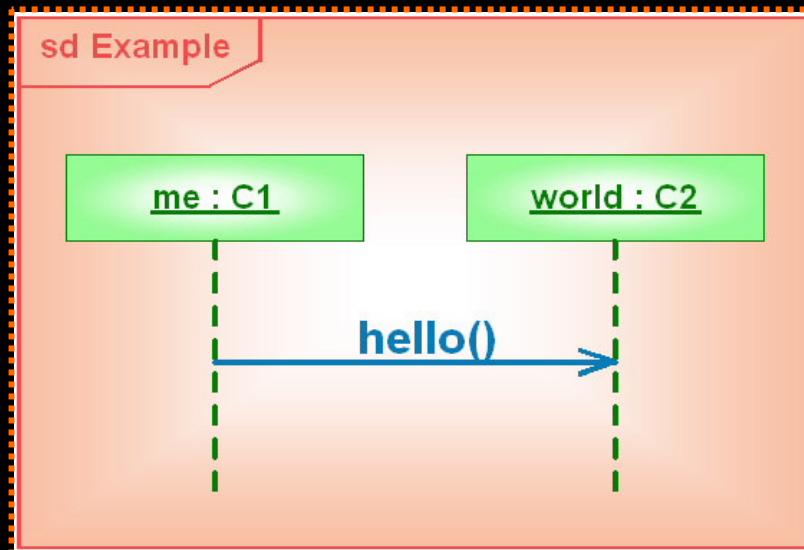
MDM Tool



MDM Example

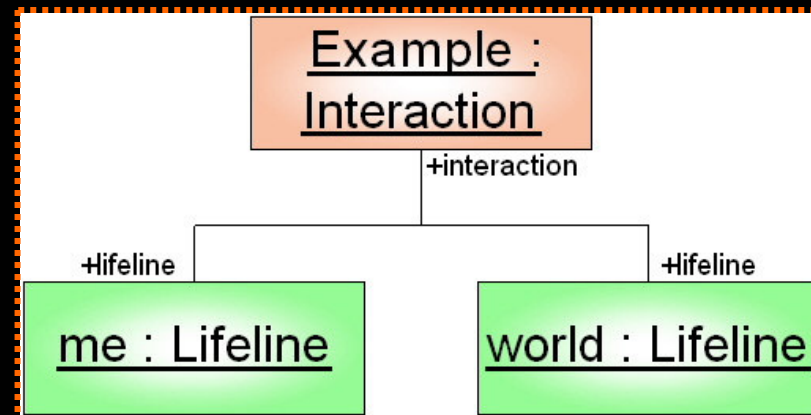
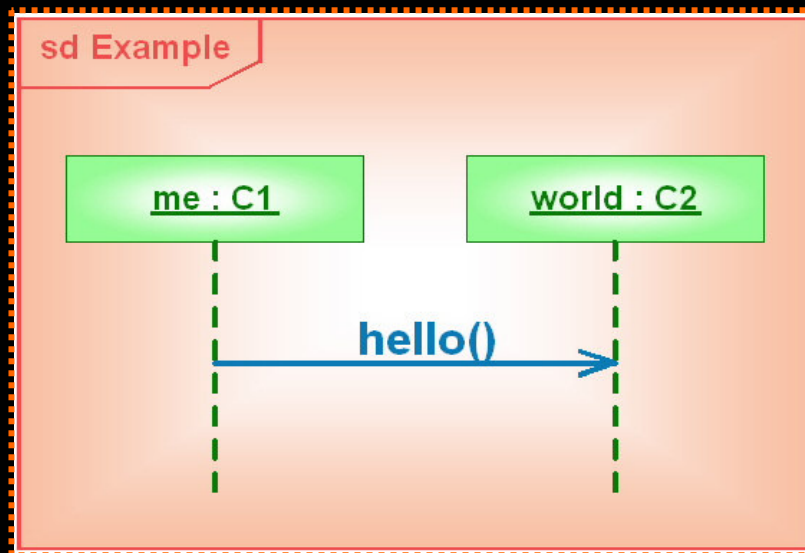


MDMExample – Step 1

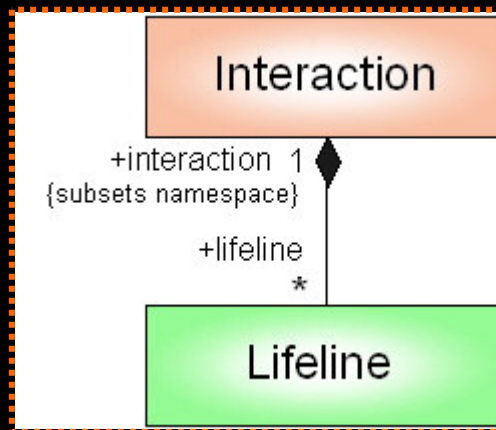
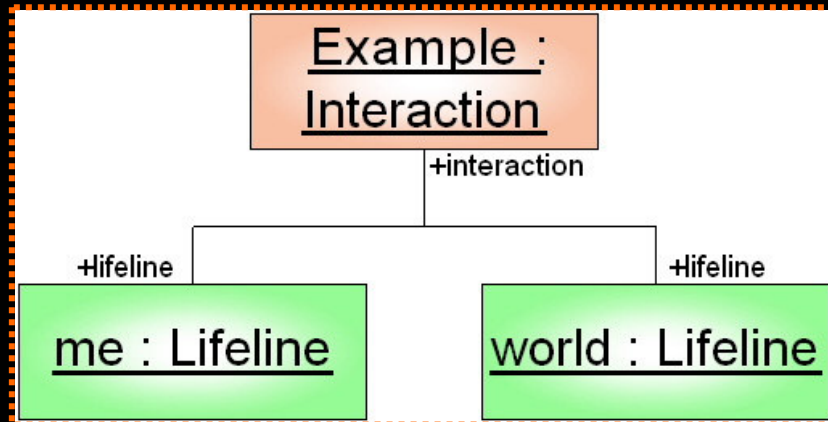
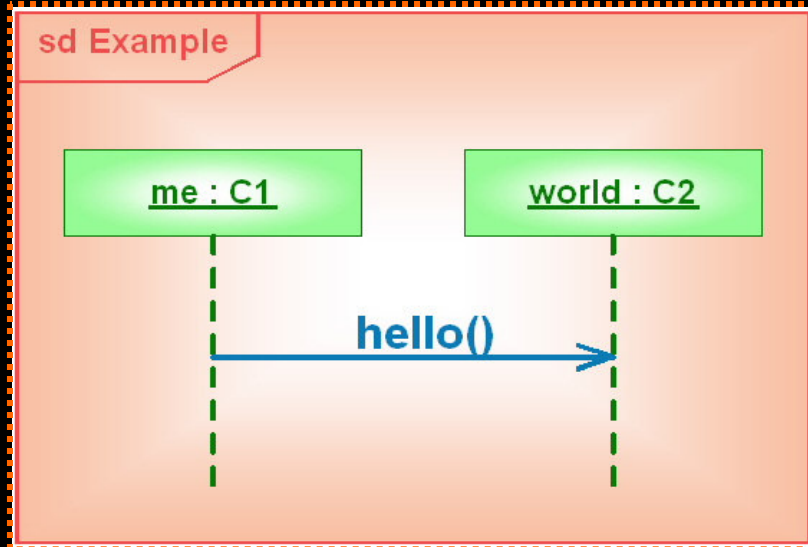


Example :
Interaction

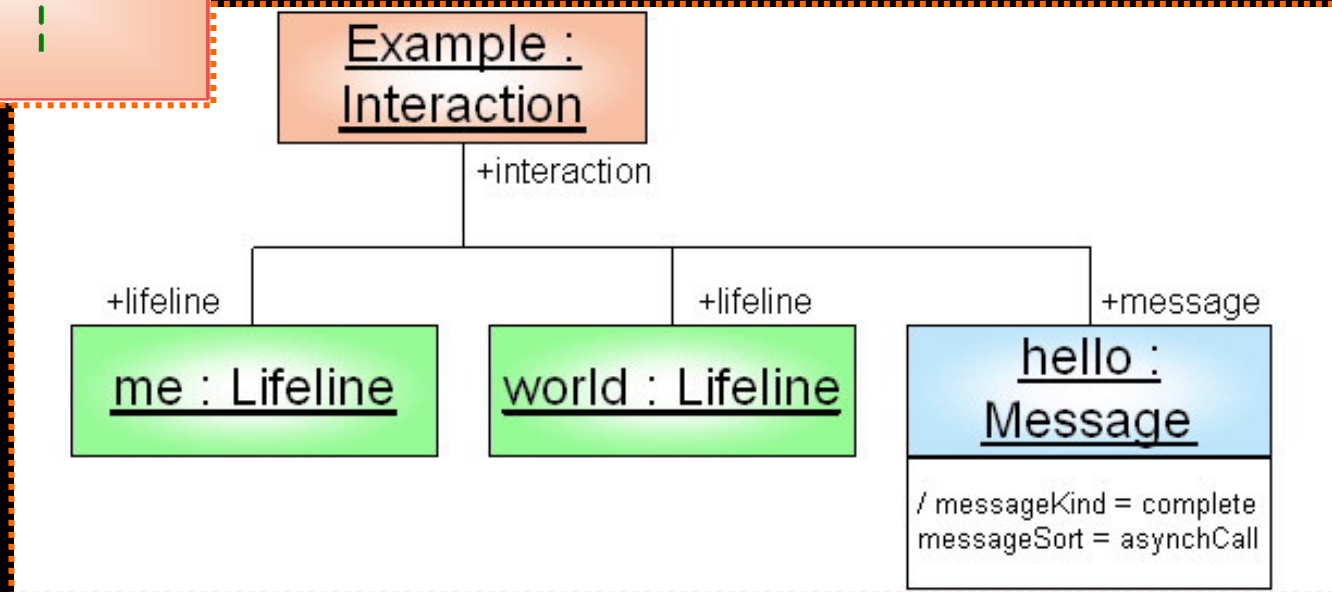
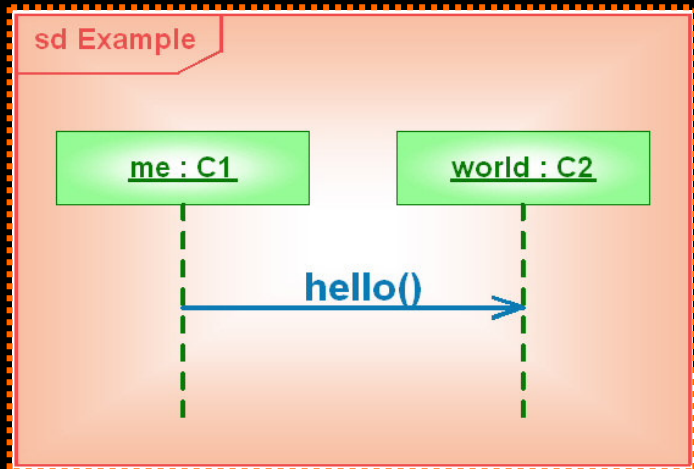
MDM Example – Step 2



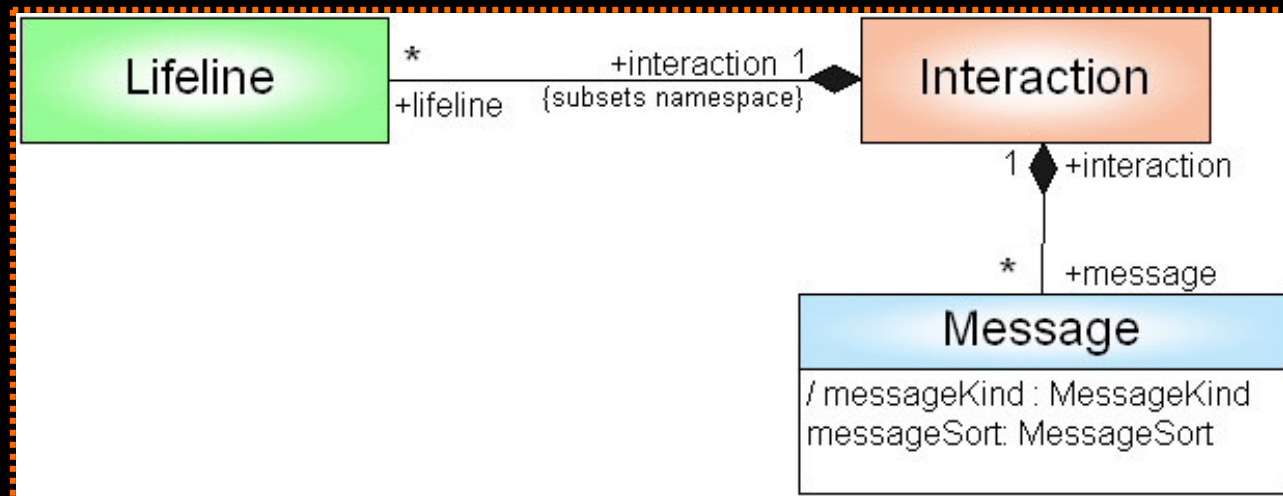
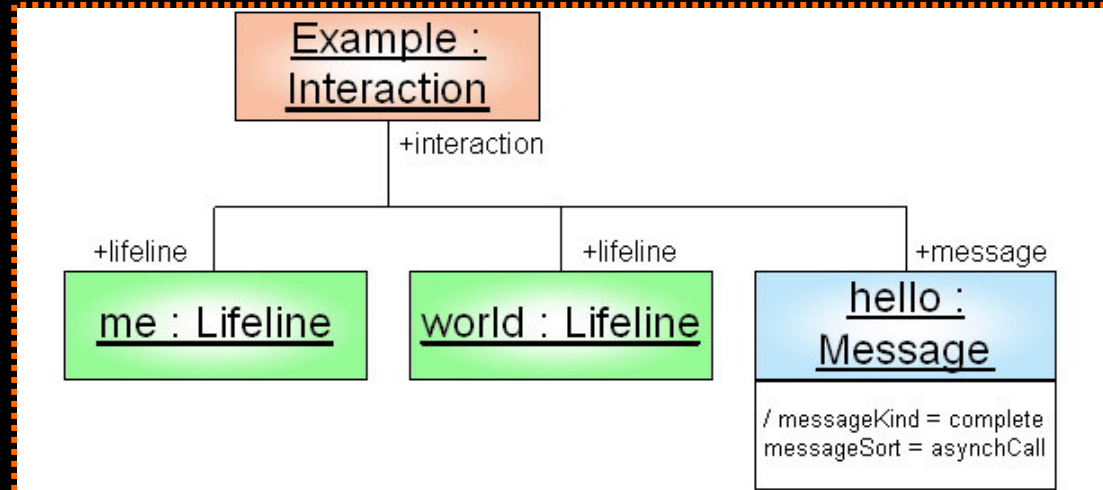
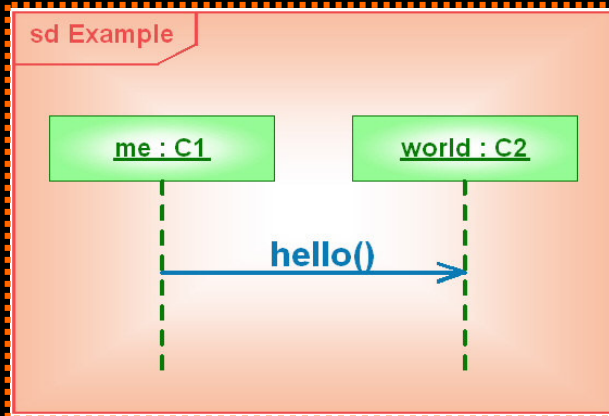
MDM Example – Step 2



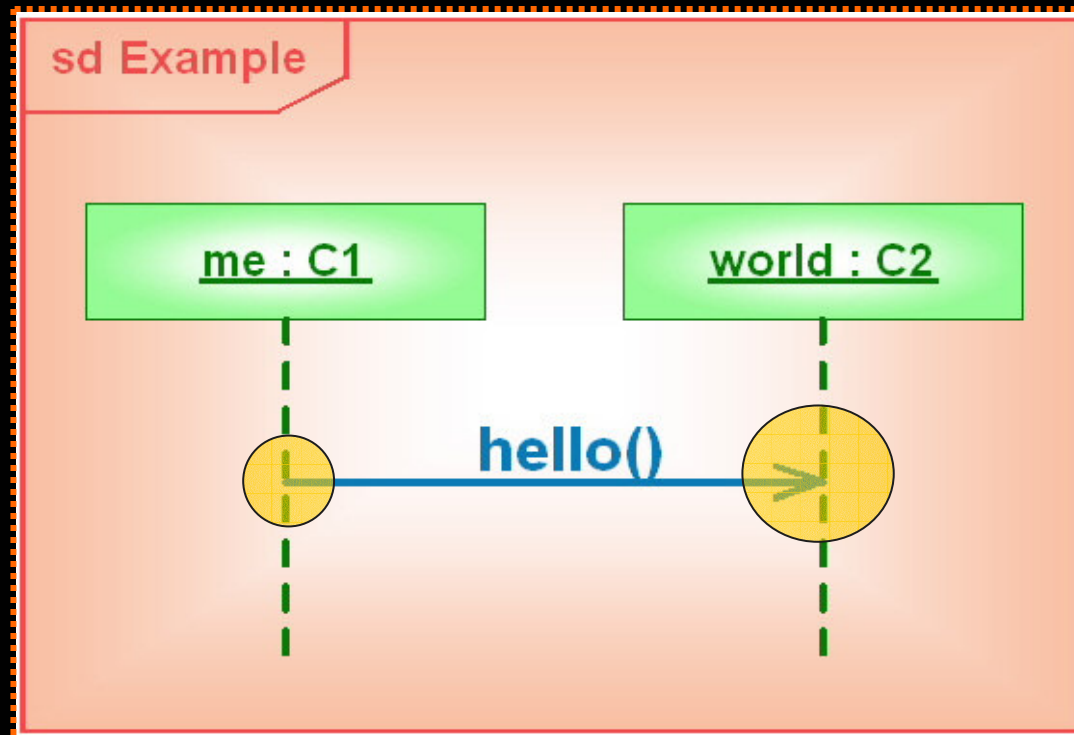
MDM Example – Step 3



MDM Example – Step 3



MDM Example – Step 4

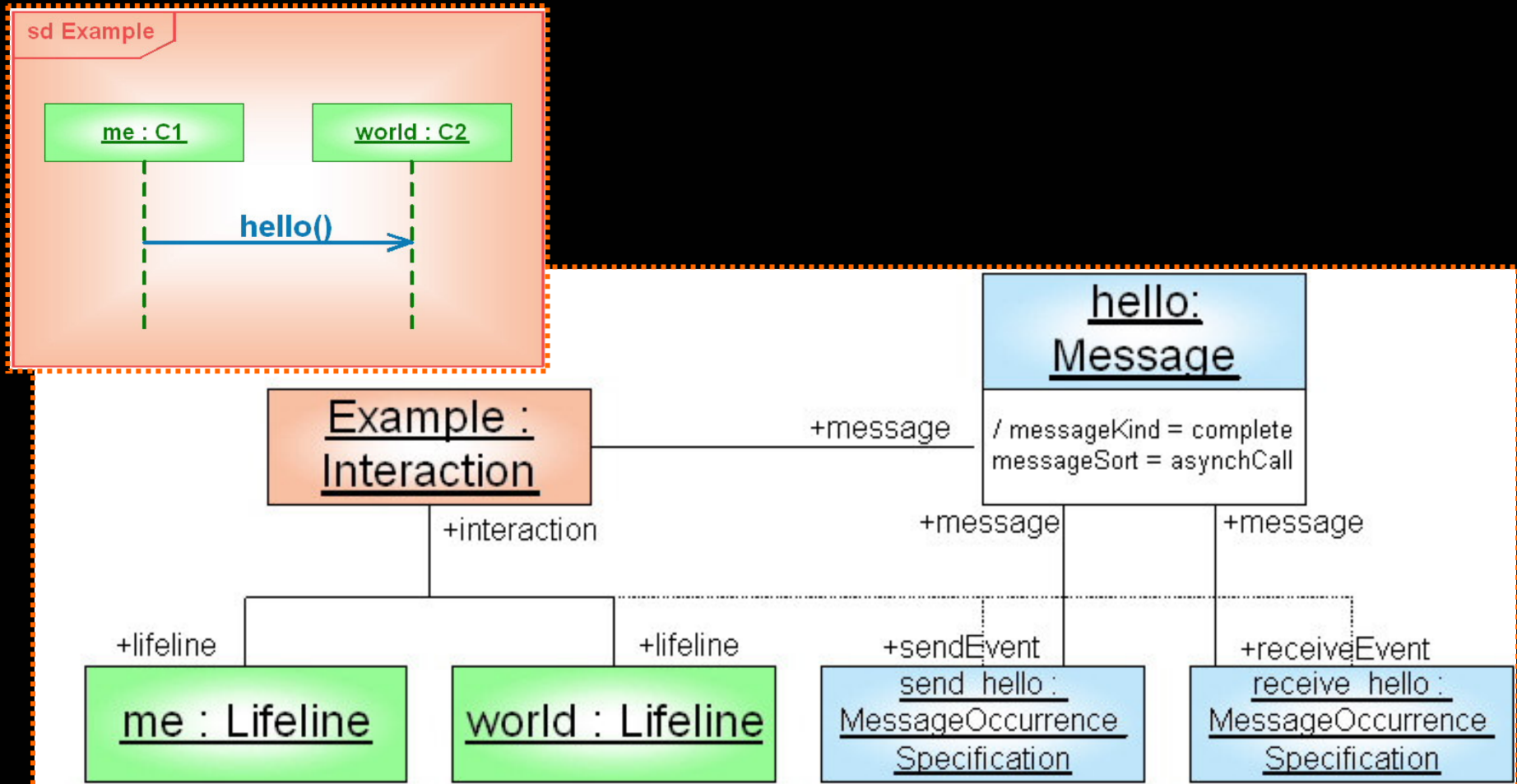


sendEvent

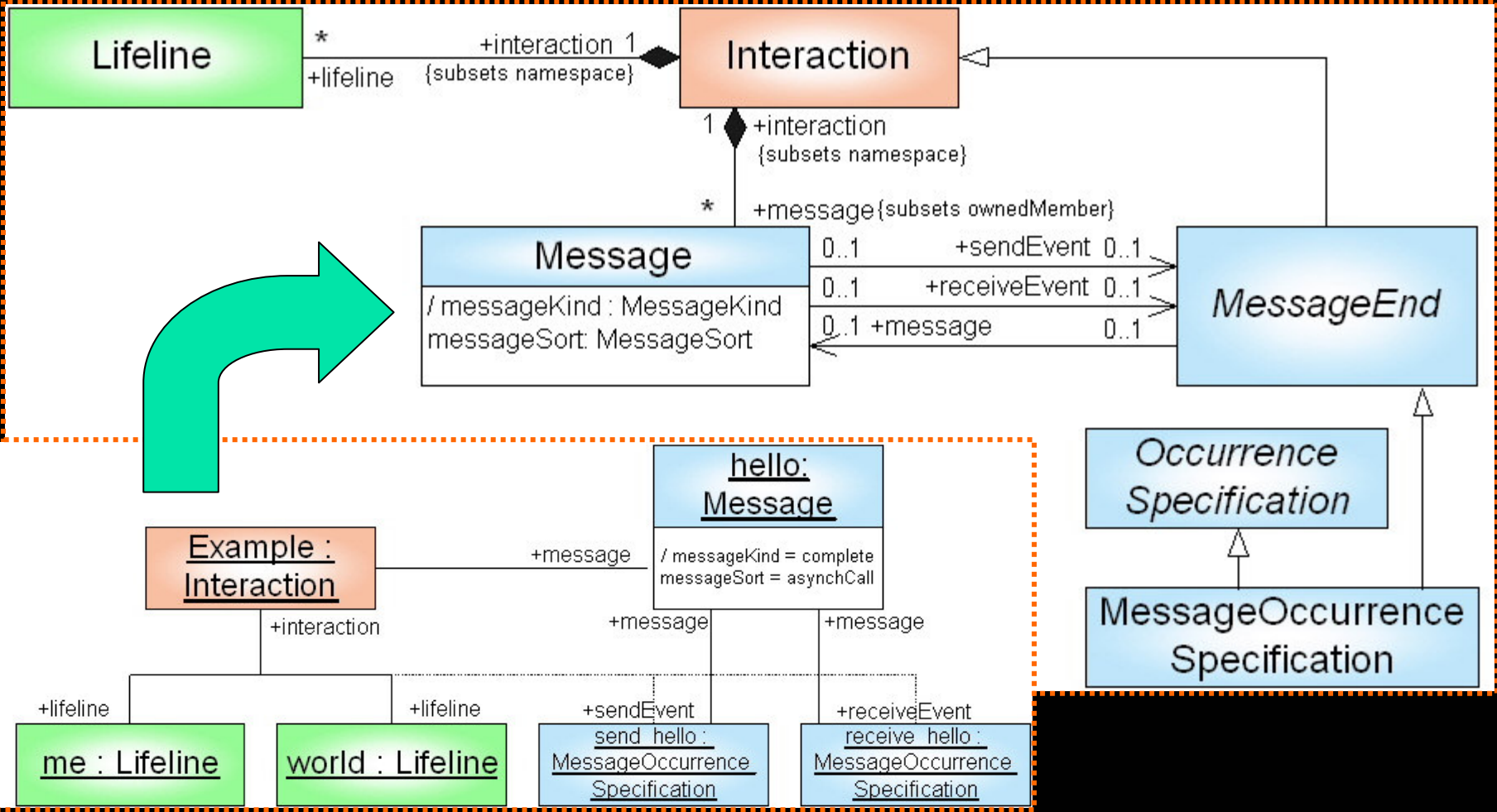
receiveEvent

messageOccurrenceSpecification

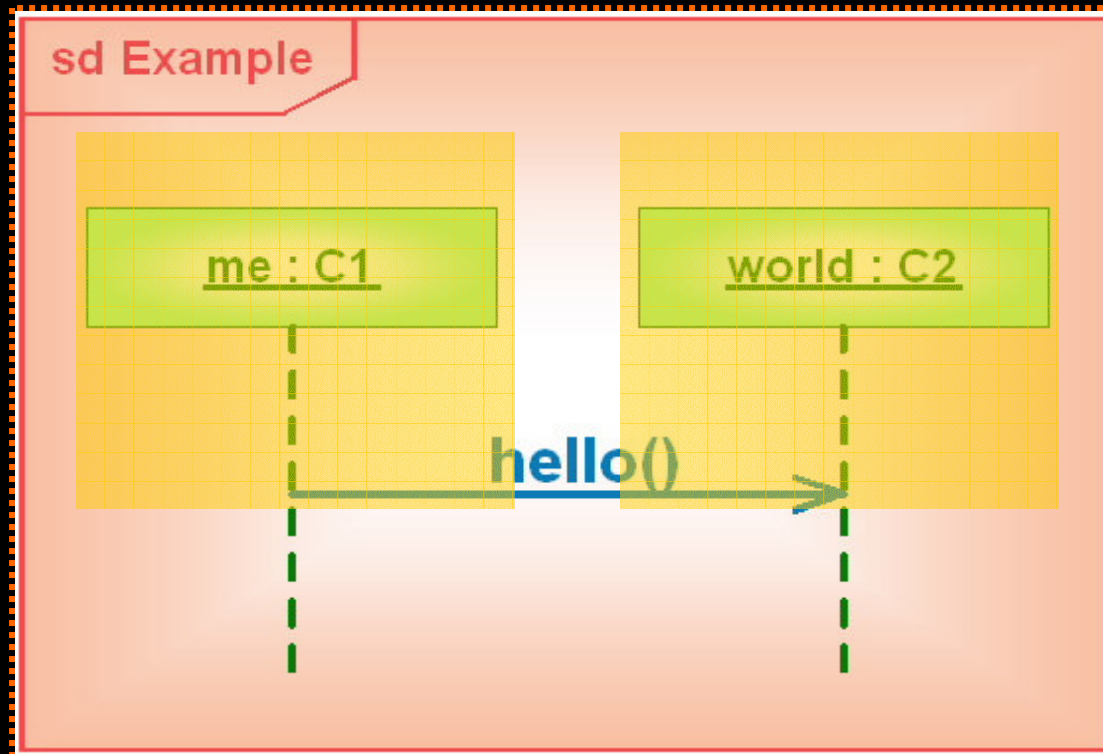
MDM Example – Step 4



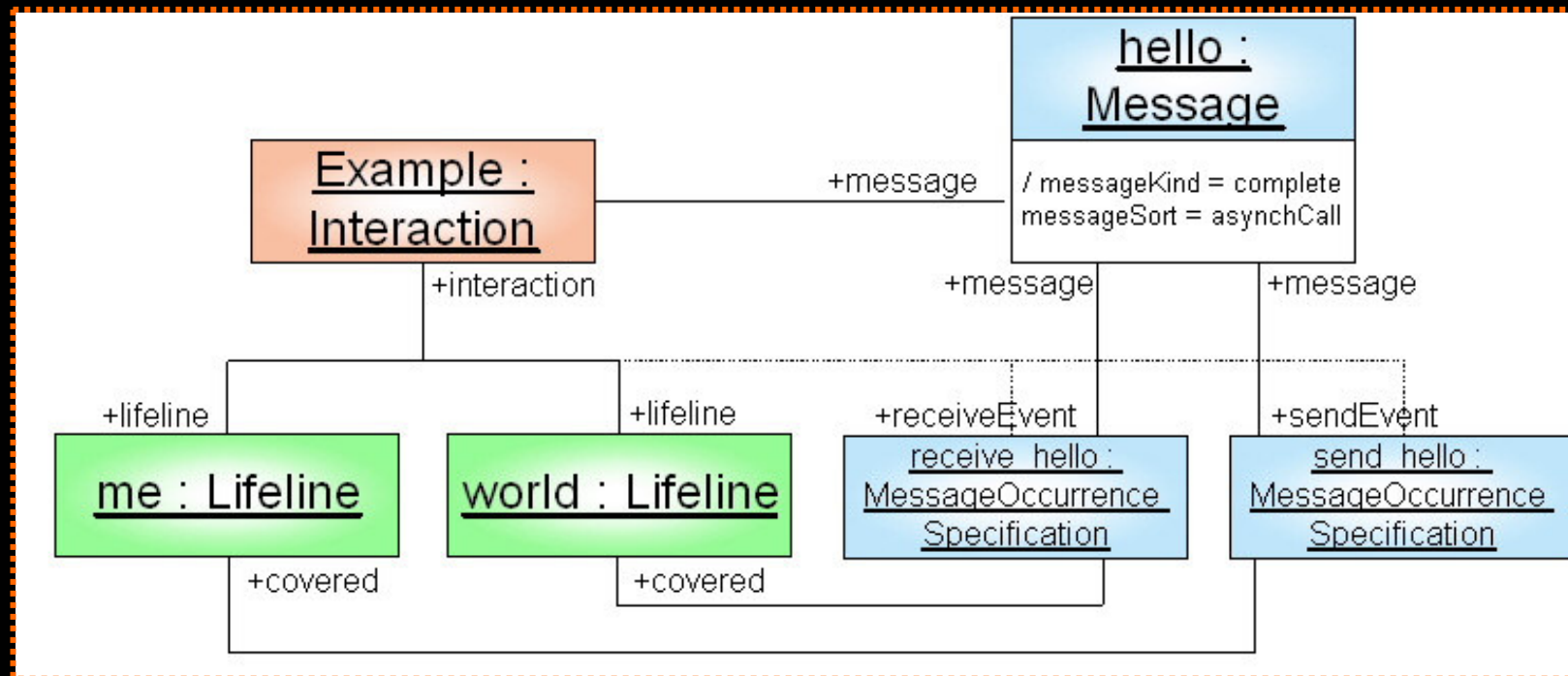
MDM Example – Step 4



MDM Example – Step 5

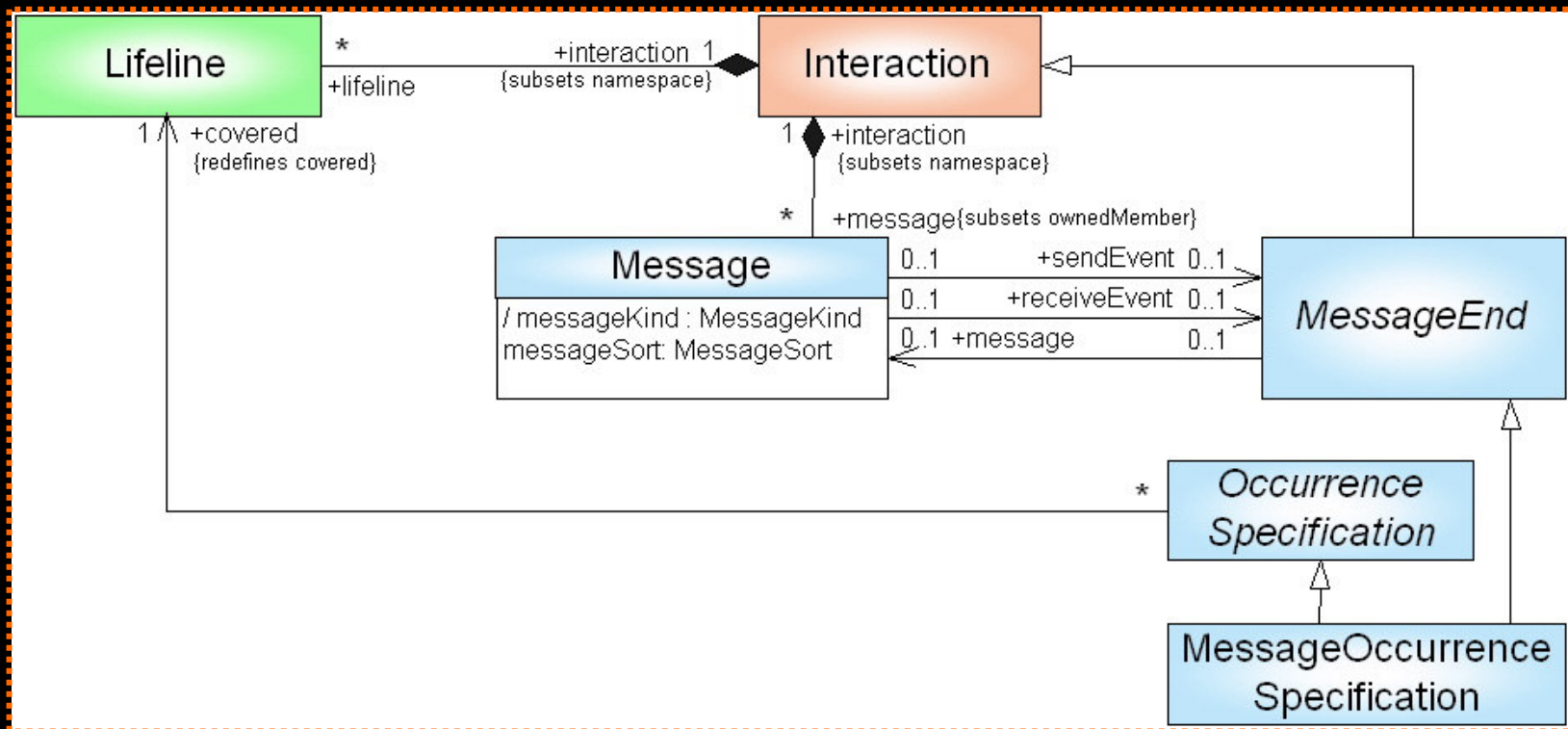


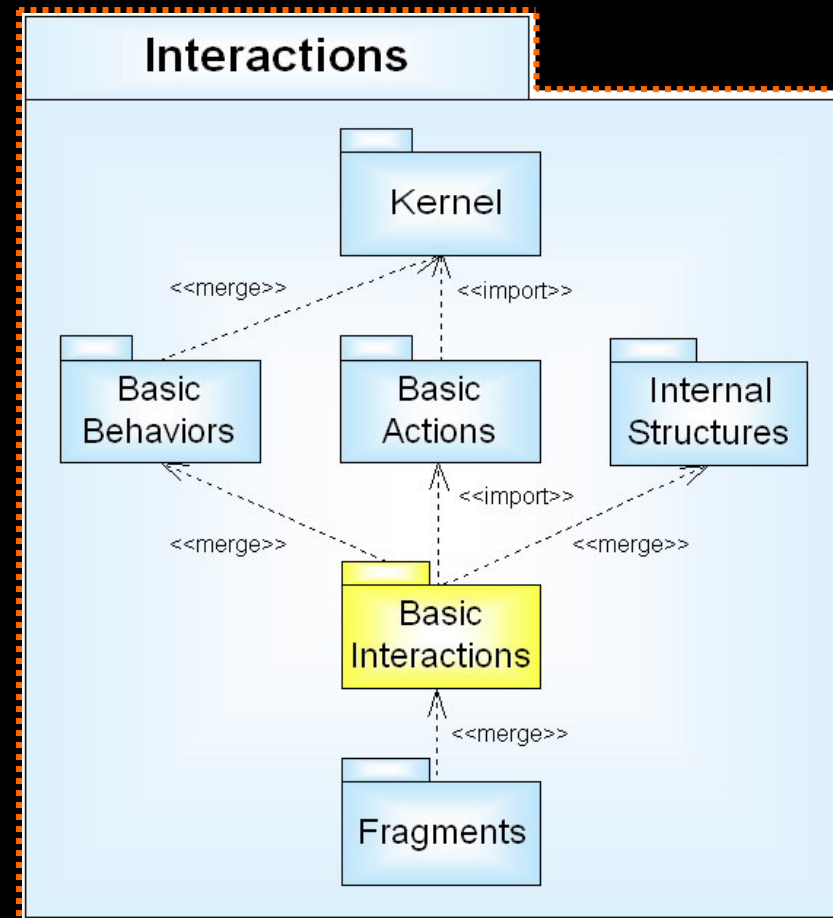
MDM Example – Step 5





MDM Example – Step 5





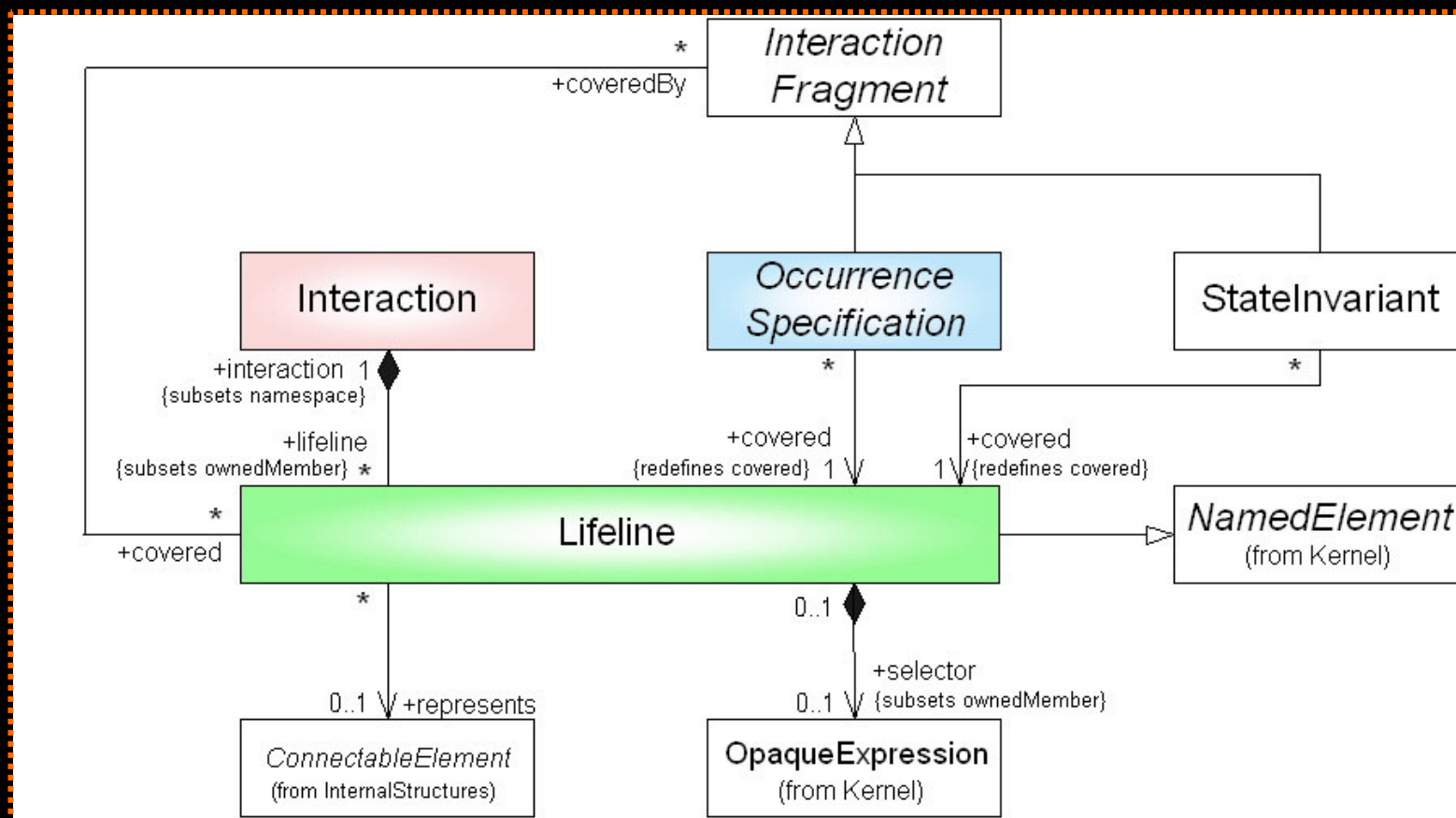


BasicInteractions Interactions



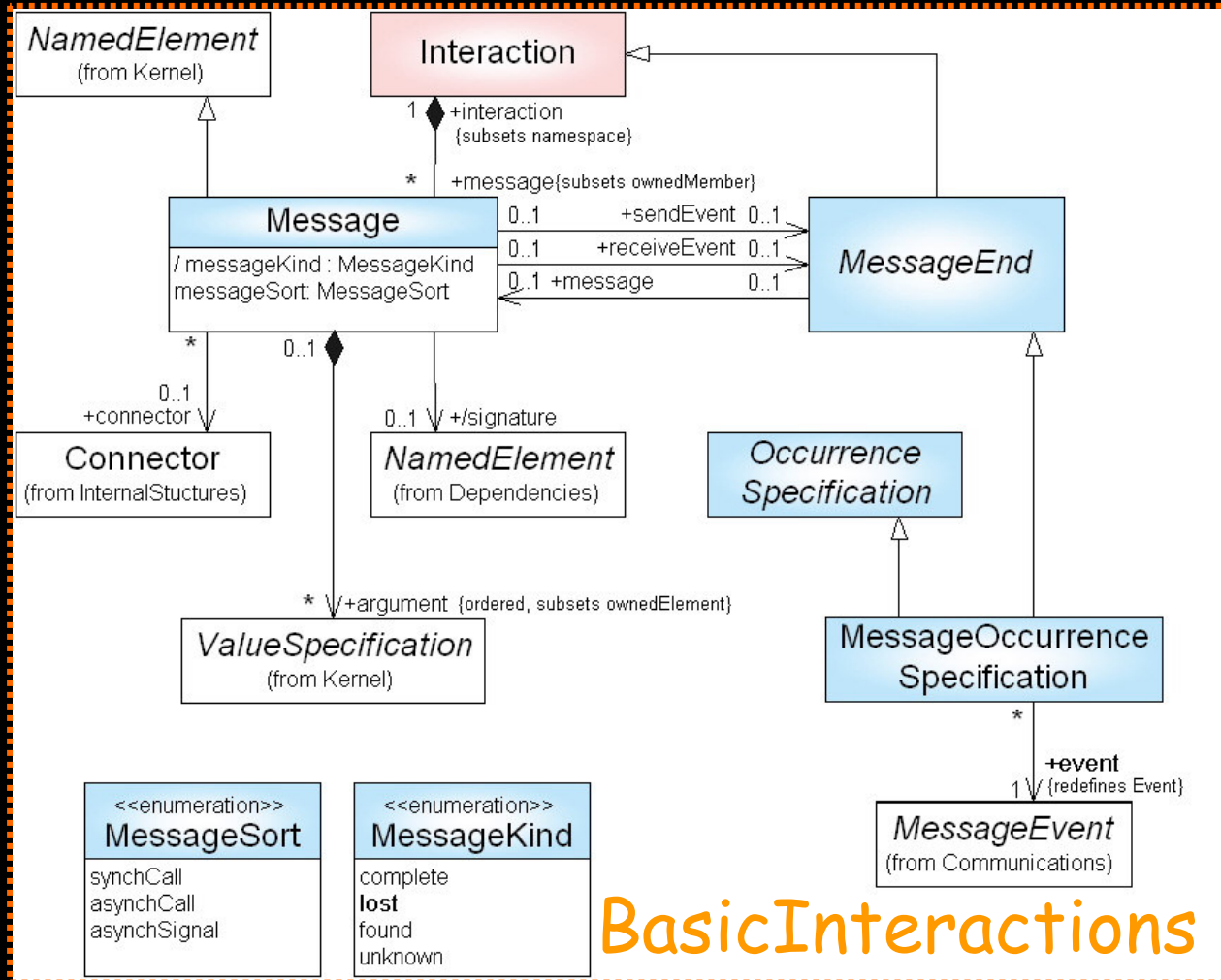
UML Metamodel Extracts

BasicInteractions - Lifelines





UML Metamodel Extracts

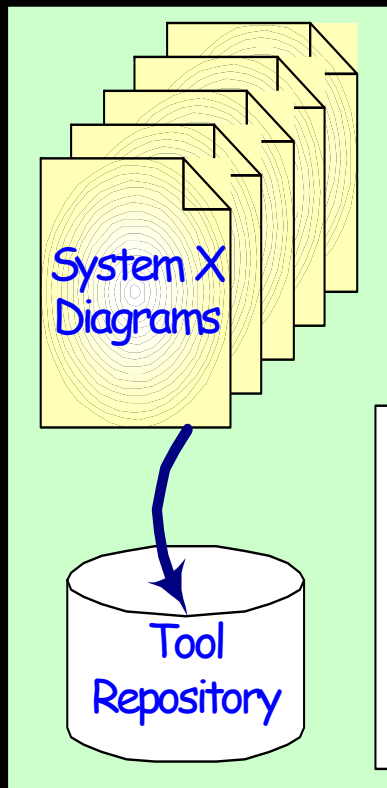


BasicInteractions - Messages

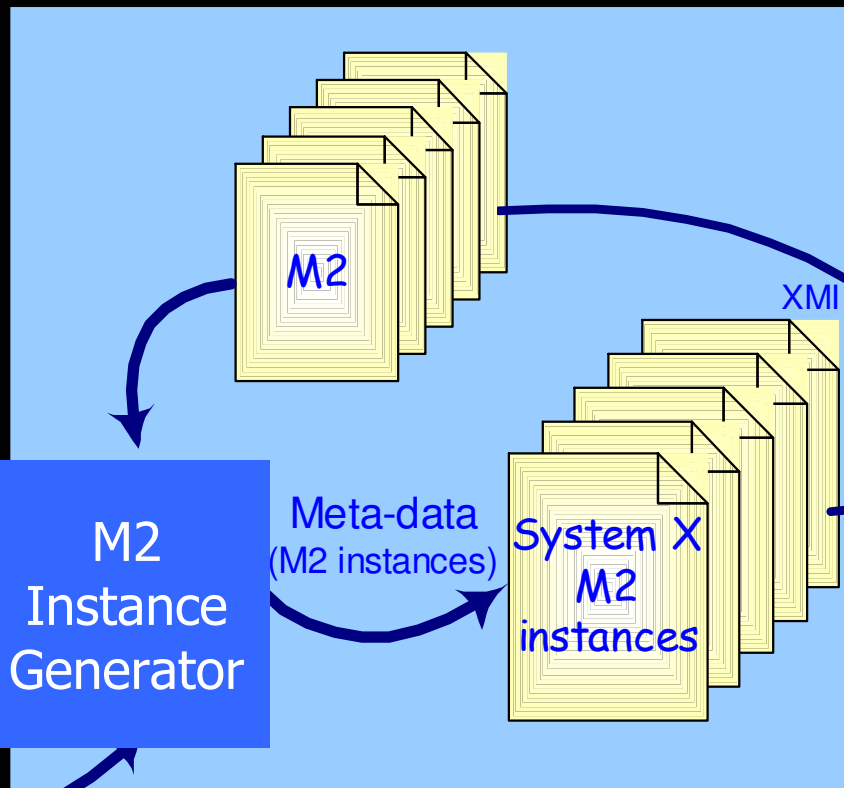
MDM Framework



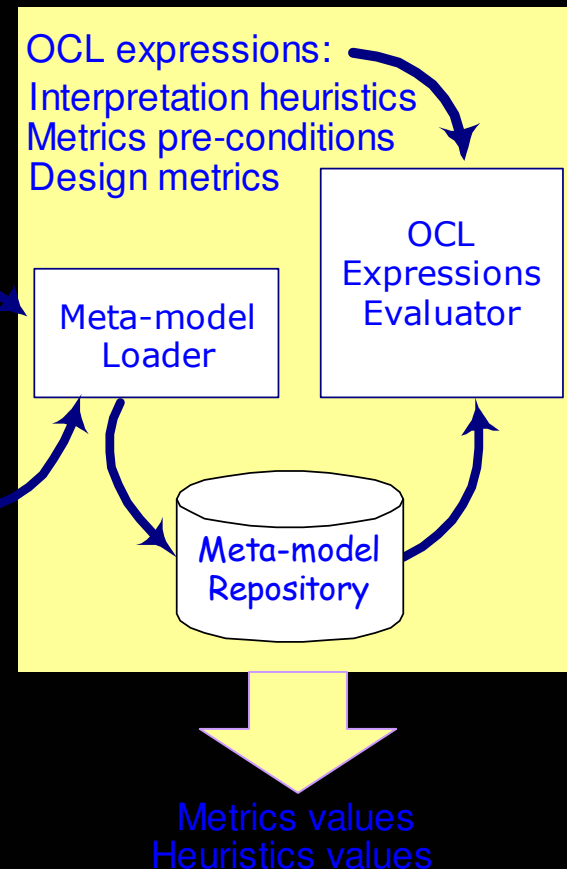
M1 Modeling Tool



M2 Instance Generator Tool

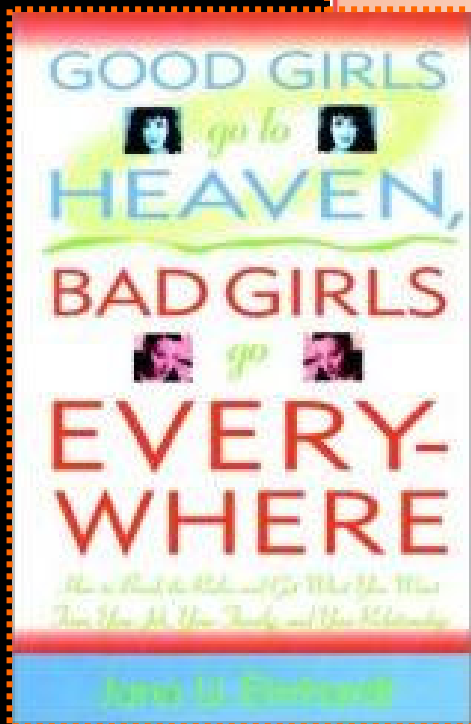
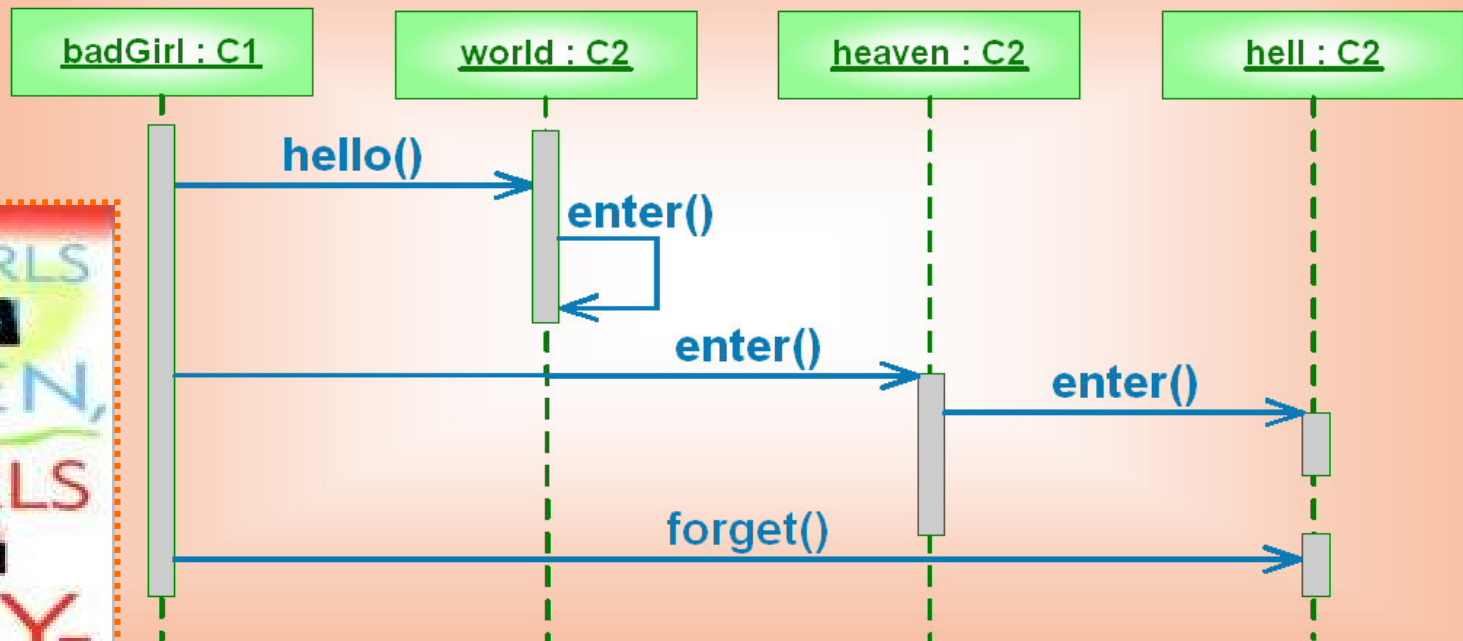


MDM Tool

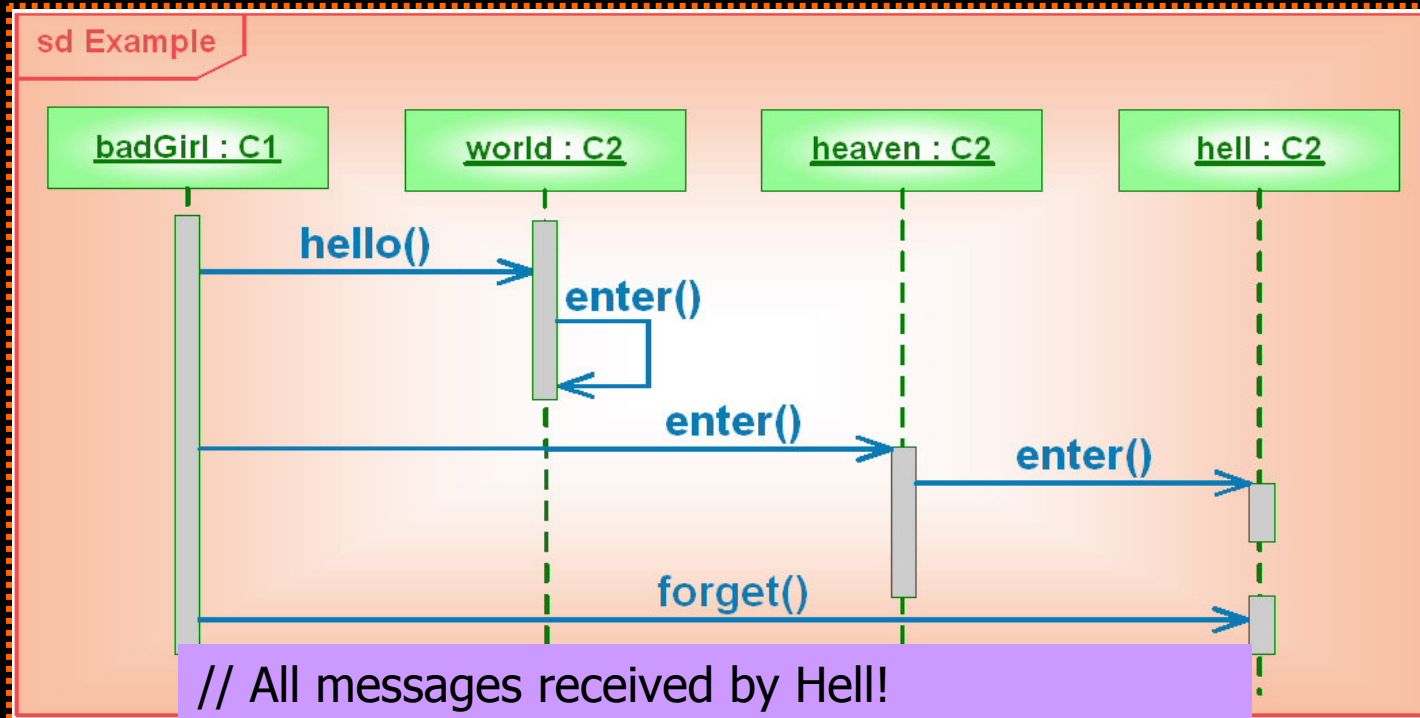


MDM Example

sd Example



MDM Example



// All messages received by Hell!
 ? hell.MessagesReceivedNumber()

Lifeline::AllMessagesReceivedNumber()

Occurrence

// Number of requested Hell entrances

->collect(message)

Lifeline::MessagesReceivedNumber()

? hell.NamedMessagesReceivedNumber("Enter")

self.AllMessagesReceivedNumber()

>> 1

Lifeline::NamedMessagesReceivedNumber()

(designation: String). Integer = self.AllMessagesReceived->select(name = designation)->size



Conclusions

- Metrics definition and validation: mitigated problems
 - Lack of formality
 - Lack of context
 - Lack of validation
 - Lack of open tools
 - Lack of coverage
- Generic MDM framework for MDE
 - UML MM1.* & 2.0, CWM, SQL:2003 Ontology, SPEM, Corba Component Model MM, ACME



- Create and formalize new metrics for uncovered behavioral models
- Formalize UML behavioral metrics defined by other researchers
 - In the context of the UML 2.0 MM
- Validate the metrics for the following purposes:
 - Model refactoring
 - Product quality improvement assessment (detection of faults and inconsistencies)
 - Resource estimation modeling
 - Impact analysis



More on this subject...



- Related papers may be found at our web site:
<http://ctp.di.fct.unl.pt/QUASAR>
- Our email addresses:
alinebaroni@di.fct.unl.pt
fa@di.fct.unl.pt
pg@di.fct.unl.pt
- FCT: <http://www.fct.mces.pt>
- Questions and comments?