EMERGENCY RESOURCE MANAGEMENT AND COORDINATION

İlker M. Karakaş
Doğan K. Berktaş
Eren Algan

MAY 20TH 2009, ANKARA

AGENDA

- Introduction; Problem Statement
- Description of the domain
- Domain analysis
- Mapping of domain concepts to grammar
- Metamodel Definition
- Concrete Syntax
- M2M Transformation
- M2T Transformation
- Lessons learned and conclusions
- References

INTRODUCTION

- **Aim** is to come up with a **domain-specific language** that would be of use in the **resource management** sub-domain of the **emergency management** information systems space

- **Why** this domain?
  1. Existing domain knowledge; partly overlaps with professional work done by a group member
  2. Easy to map to and visualize real-world scenarios
  3. Interesting, yet untouched

Emergency resource management and coordination - example usage scenarios (Cont.'ed)

- ***Pandemic Influenza Scenario***: The scenario follows scenario models an Influenza Pandemic outbreak. It includes such activities as requesting medical facilities to take stock and determine what resources are readily available and on hand (inventory of available supplies). It includes a wide range of resource messages such as requests for vaccines and antivirals, etc.

Full use case available @ http://www.oasis-338 open.org/committees/download.php/26806/EDXL_use_example_Influenza_06152005%20LaniGrahmRev.doc

Scenario Events Chronology / Resource Request List

GENERAL PRECONDITIONS:

1. There is an Influenza Pandemic outbreak at Phase 6 (Increased and sustained transmission in general population) as determined by the State Health Agency/Public Health Department.
2. The activity for dealing with an outbreak of Influenza Pandemic is spread out over a large period of time (weeks – months)
3. Protocols and aid agreements are already in place with the Centers for Disease Control for some medical resources to be obtained using the Strategic National Stockpile.
4. Mutual aid agreements are already in place within the region.

EVENT CHRONOLGY:

1. Request to assemble Incident Management Team.
    a. This team determines:
        i. resource and personnel needs
        ii. roles and responsibilities
2. Notify Governor's Office/top state officials that Phase 6 has been declared.
3. Declare a Public Health Emergency for the affected state if and when applicable (Governor decides upon advice of State Health Agency and Emergency Management Agency). With pandemic flu I cannot imagine a situation where the Governor of any state would declare a Public Health Emergency for a region within his or her state.
4. Request to activate the Influenza Surveillance Operation (this includes coordination and reporting between a network of doctors, healthcare facilities, labs, schools and State Health Agency/Public Health Department. This may be already in place at an earlier Phase or because it is the season).
5. Request involvement from the state Emergency Manager.
6. Request medical facilities to take stock and determine what resources are readily available and on hand (inventory of available supplies).
7. Distribute Public Health Alerts/Messages (these need to be consistent and continuous – these public health alerts will emphasize pre-cautions to take in order to avoid spread of infection respiratory etiquette and what to if you are ill. We will also need at this point to provide information about both the expected availability of vaccine and anti-virals.
8. Distribute Healthcare personnel Alerts/Messages (these will be much more detailed and be oriented for medical personnel).

Emergency resource management and coordination - example usage scenarios (Cont.'ed)

- More scenarios (showing the need for automation of resource messaging to the extent possible)
  - Fire Incidents
  - Hurricanes
  - Floods
  - Earthquakes
  - Man-made disasters
  - …

PROBLEM STATEMENT

- An industry standard on resource management specification is available.
- However the **standard is limited** since it does not provide solutions for **composition (coordination) of Resource Management 'messages'**.
- Moreover, there is a need for **tooling** that support efficient utilization and coordination of resources during emergencies.

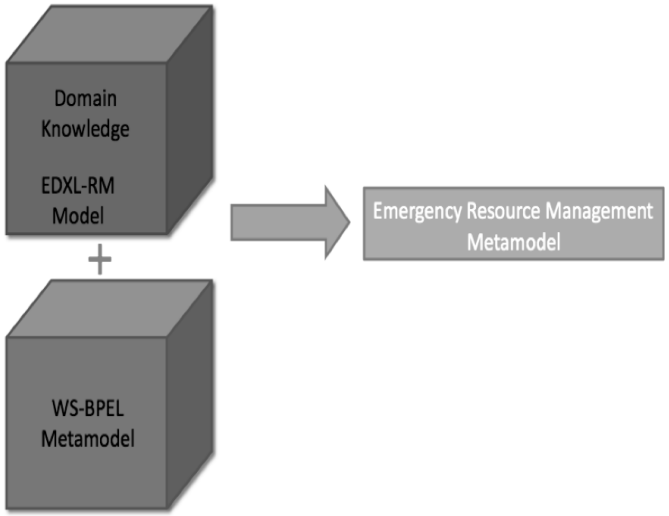
DOMAIN DESCRIPTION

- Emergency
  - Happenings/incidents including natural disasters (earthquakes, floods, fires, ..), man-made disasters, etc.
- Resource
  - Things that are exchanged during emergencies: search and rescue teams, equipment, food, mobile shelters/tents, etc..
- Resource Management and Coordination
  - Being able to coordinate (i.e. plan, collaborate, track) discovery, deployment, utilization and restoration (i.e. freeing up) of resources during emergencies

DOMAIN ANALYSIS

In summary, we did exploit:

- Our existing domain knowledge and experience
- OASIS EDXL-RM specification/model
- OASIS WS-BPEL Specification, and meta-model



EMERGENCY MANAGEMENT DATA EXCHANGE LANGUAGE

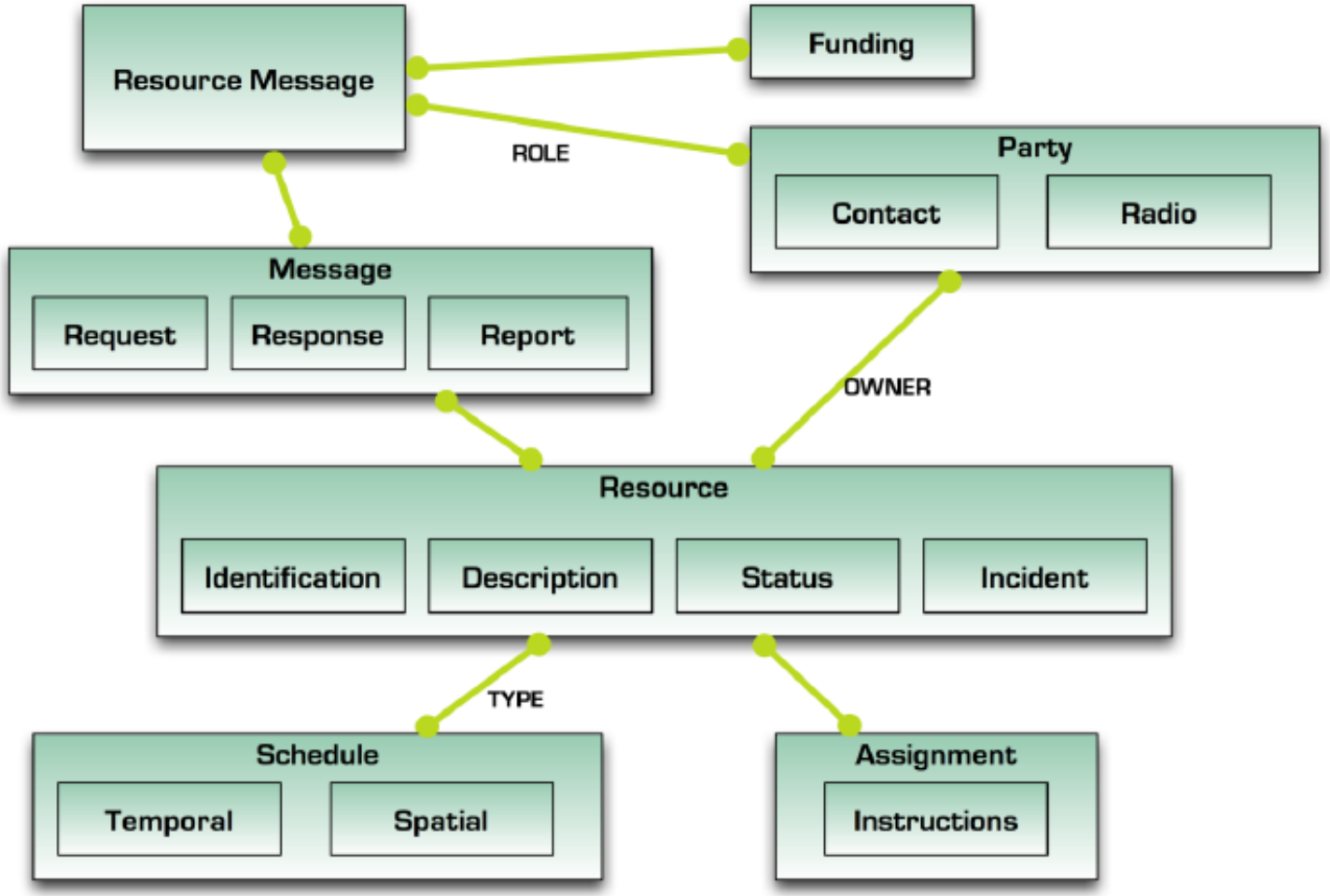
- Defined by Organization for the Advancement of Structured Information Standards (OASIS) Consortium (an official OASIS recommendation as of 1 November 2008). OASIS Emergency Management TC has other EM specifications.

Technical Work Produced by the Committee

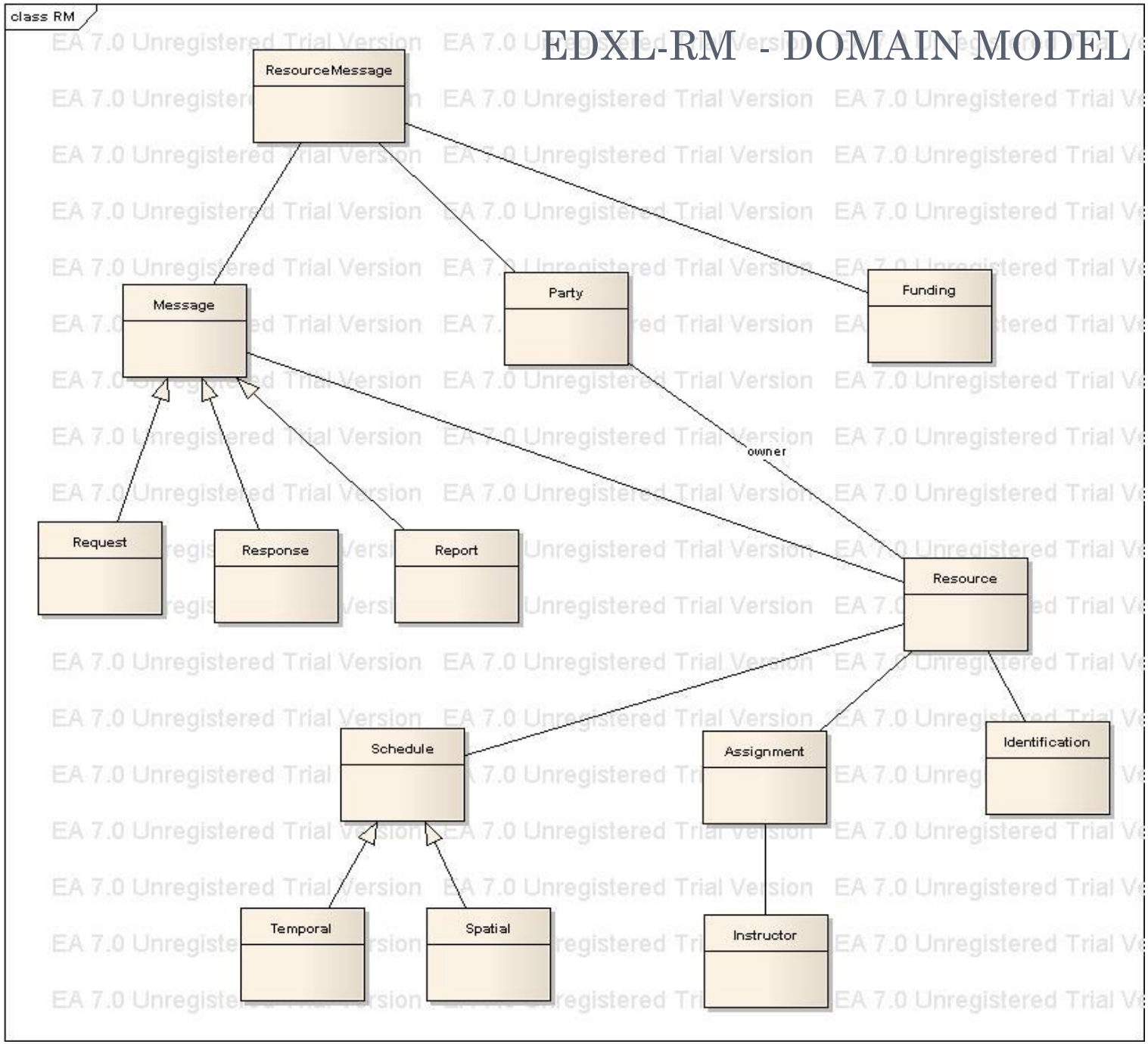
- Emergency Data Exchange Language Resource Messaging (EDXL-RM) 1.0 Public Review Draft 02, 31 January 2008.
- Emergency Data Exchange Language Hospital AVailability Exchange (EDXL-HAVE) 1.0 Public Review Draft 05, 04 March 2008.
- Emergency Data Exchange Language Distribution Element (EDXL-DE) 1.0 OASIS Standard EDXL-DE v1.0, 1 May 2006.
- Common Alerting Protocol (CAP) 1.1 OASIS Standard CAP-V1.1, October 2005 (Corrected DOM).
- Common Alerting Protocol (CAP) 1.1 OASIS Standard CAP-V1.1, October 2005.
- Common Alerting Protocol (CAP) 1.0 OASIS Standard 200402, March 2004.
- Wiki for OASIS Emergency Management TC member collaboration

- Focused on message definitions only; coordination of messages is not defined

EDXL-RM - ABSTRACT REFERENCE MODEL



EDXL-RM - DOMAIN MODEL



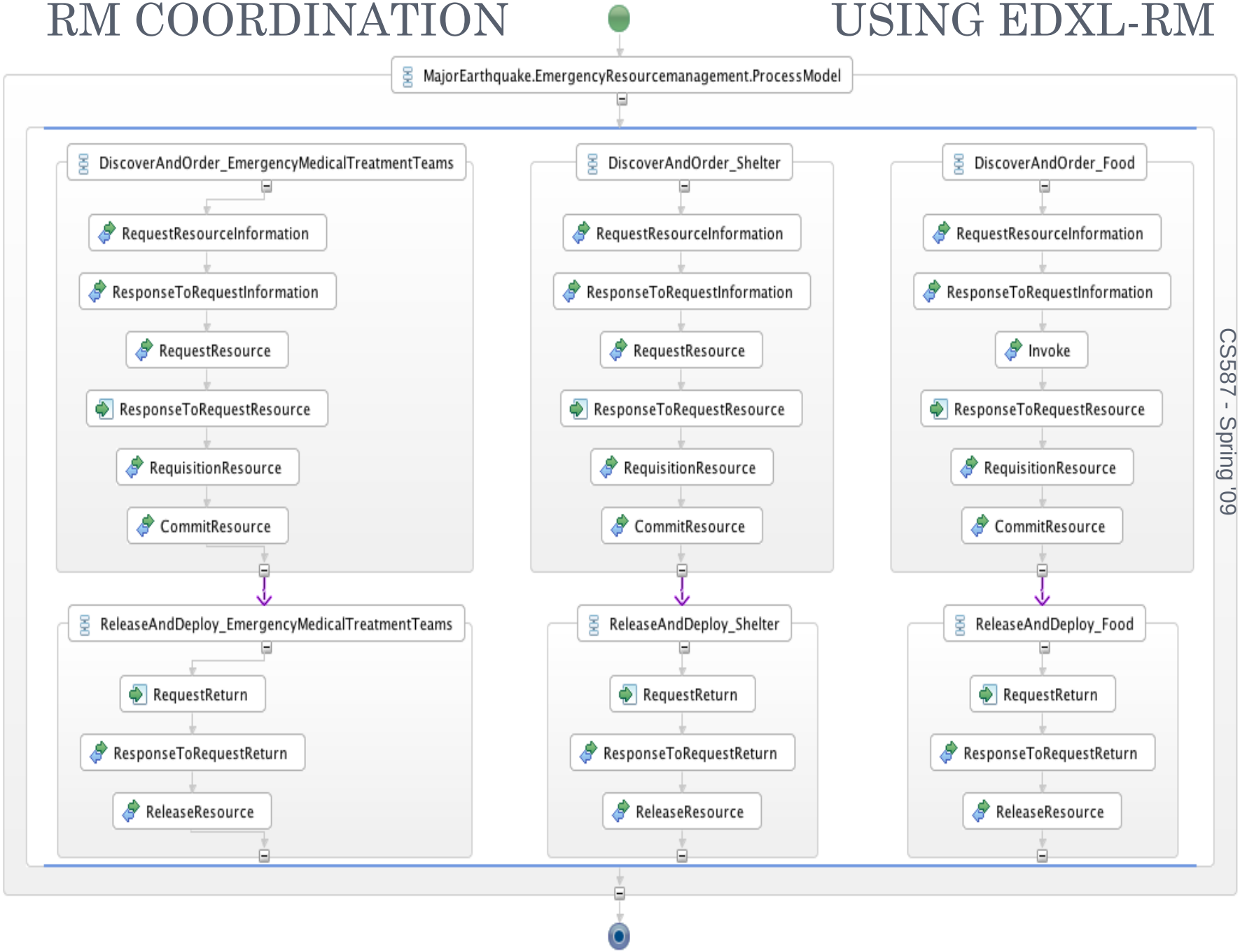
EDXL-RM - MESSAGE TYPES

Message Type	Description	Message Sender
RequestResource	Message used to request needed resources from one or many recipients, possibly spawning multiple responses.	Resource Consumer
ResponseToRequestResource	Message used as the response to a "RequestResource". Allows sender to list resource(s) which they feel represent suitable match with a resource request.	Resource Supplier
RequisitionResource	Message used to "order" specific resource, or to confirm specific resource to be "ordered" relating to one or more responses to a "RequestResource".	Resource Consumer
CommitResource	Message used to agree or commit specific resource in response to a RequestResoure or RequisitionResource,".	Resource Supplier
RequestInformation	Message used to ask resource questions or provide general description of situation and general resources needs.	Resource Consumer, Resource Supplier
ResponseToRequestInformation	Message used as the response to a RequestInformation message providing general information or to list resource that may meet the specified need.	Resource Supplier, Resource Consumer
OfferUnsolicitedResource	Message used to offer available resources (that have not been requested) to assist with an emergency response.	Resource Supplier
ReleaseResource	Message used at the incident to "release" (demobilize) resource back to its original Supplier.	Resource Consumer
RequestReturn	Message used to request release (demobilize) of resources back to its original point of assignment or to another location / assignment ("I want my stuff back").	Resource Supplier

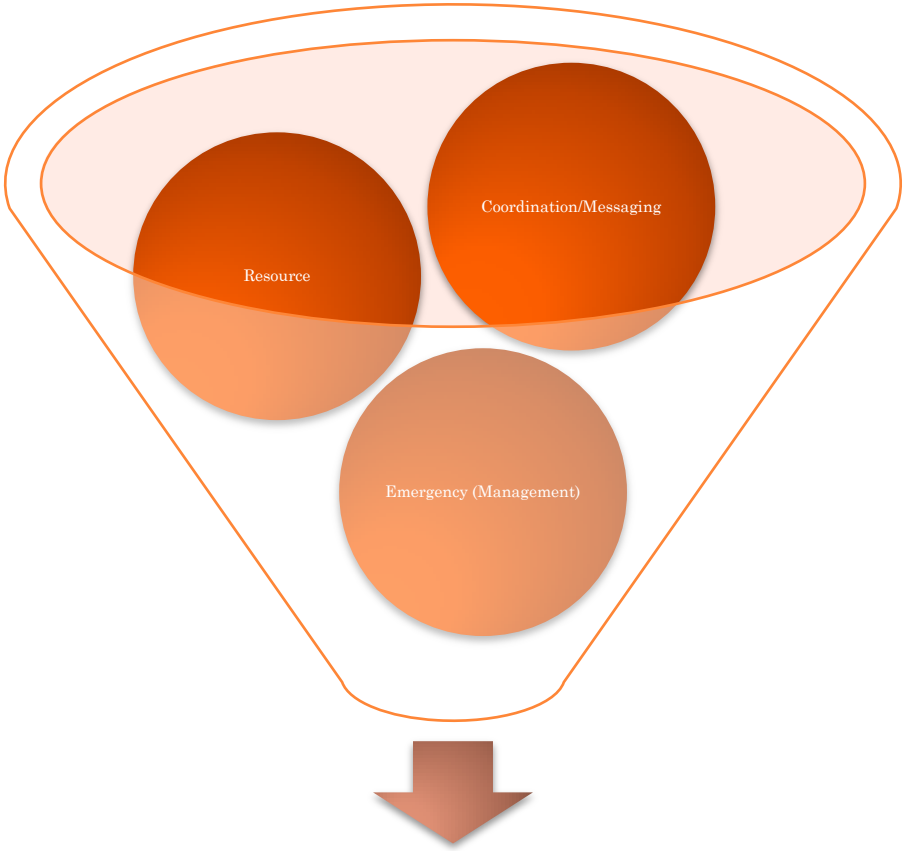
EDXL-RM - MESSAGE TYPES (CONT.'ED)

Message Type	Description	Message Sender
ResponseToRequestReturn	Message used as the response to a "RequestReturn" indicating whether the resource may be released, with relevant time-line information.	Resource Consumer
RequestQuote	Message used to request a price quote from a seller or supplier.	Resource Consumer
ResponseToRequestQuote	Message used as the response to a "RequestQuote". Allows sender to list resource(s) which they feel represent suitable match with the request, with pricing information.	Resource Supplier
RequestResourceDeploymentStatus	Message used to request current "status" of resource.	Resource Consumer, Resource Supplier
ReportResourceDeploymentStatus	Message used to report on the current "status" of any resource.	Resource Consumer, Resource Supplier
RequestExtendedDeploymentDuration	A request initiated by the requester / receiver of resource, "I want to extend how long I need to keep this resource"	Resource Consumer
ResponseToRequestExtendedDeployment Duration	Message used as the response to "RequestExtendedDeploymentDuration".	Resource Supplier

RM COORDINATION USING EDXL-RM



DOMAIN ANALYSIS: RESULT



Metamodel

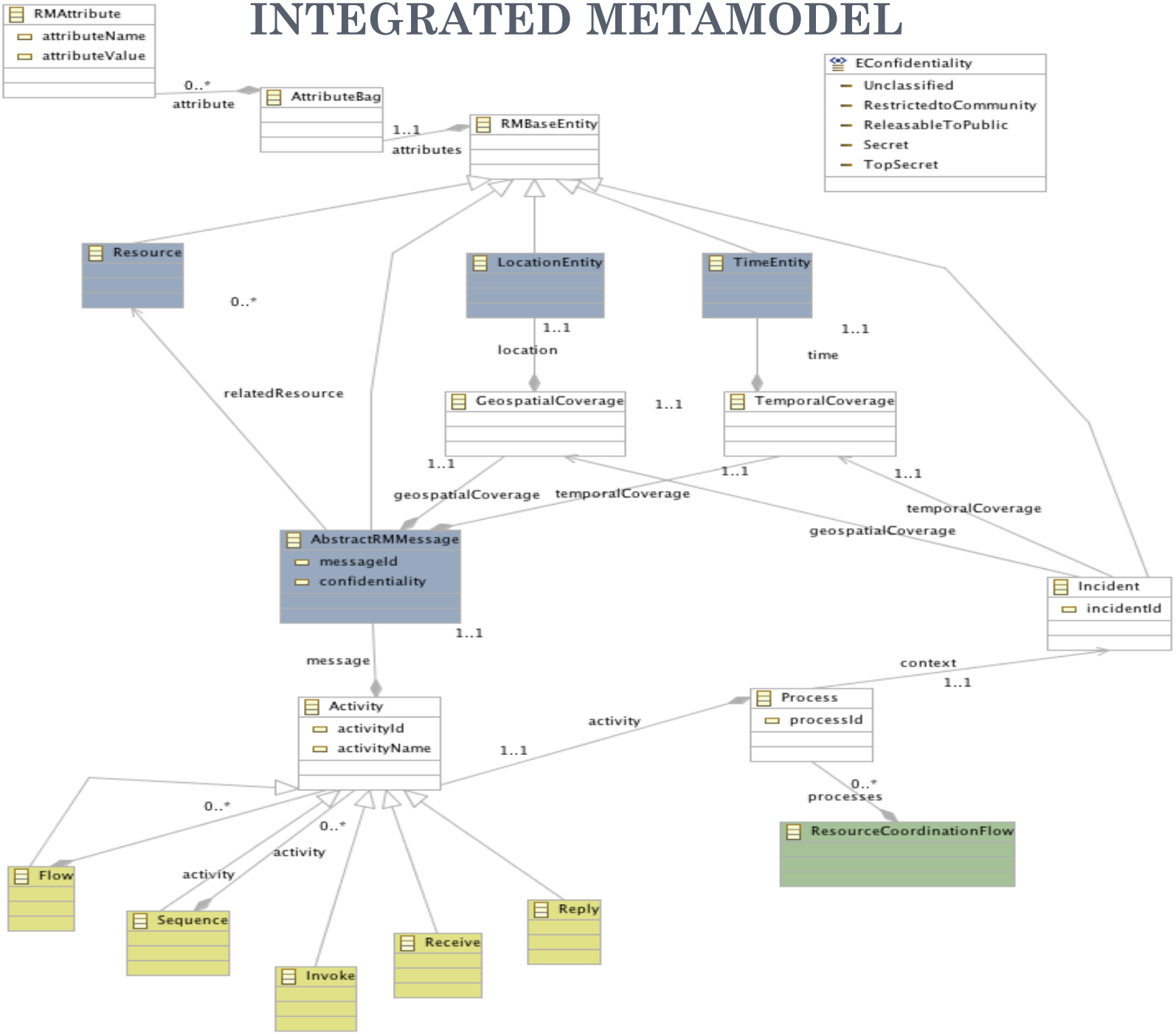
DOMAIN GLOSSARY

Concept	Description
ResourceCoordinationFlow	Is the 'entry' meta-concept. Comprises a collection of resource coordination processes.
Flow	Is a specialization of 'Activity'. Conceptually maps to the BPEL's Flow meta-model entity, therefore carries the very same semantics as defined in BPEL metamodel. Is a container for a number of 'parallel' activities.
Sequence	Is a specialization of 'Activity'. Conceptually maps to the BPEL's Sequence meta-model entity, therefore carries the very same semantics as defined in BPEL metamodel. Is a container for a number of 'sequential' activities.
Process	A conceptualization for a checklist or a workflow. Contains an activity instance, where the steps of checklist or workflow are defined. Process is a logical wrapper around the Activity meta-concept. At the same time, conceptually maps to the BPEL's Process meta-model entity, and shares the very same semantics as defined in BPEL metamodel.
Activity	Base meta-level concept for concept including Flow, Sequence, Invoke, Receive and Reply. Imported from the BPEL meta-model. This meta entitiy allows for definition of compositions of activities. A single activity instance is wrapped by a process instance
Incident	A meta incident concept. To be specialized by domain models; e.g. an 'earthquake', 'flood', 'manmade disaster', etc. Conceptually, Incident is a happening that has temporal and geospatial projections.
TemporalCoverage	Used to assign temporal coverage (i.e. time point or duration) information to either actual incidents, or to the resource management messages
GeospatialCoverage	Used to assign geospatial coverage (i.e. location) information to either actual incidents, or to the resource management messages
AbstractRMMessage	The meta message concept. Conceptually, the model level specializations of this concept shall allow expressing recource coordination messages between emergency management entities. For EDXL-RM, this meta concept could be used in generalization of the 15 different EDXL-RM messages

DOMAIN GLOSSARY (CONT.'ED)

Concept	Description
TimeEntity	Comprises the temporal coverage concept. Could be modelled at the M1 level as a point, duration, etc
LocationEntity	Comprises the geospatial coverage concept. Could be modelled at the M1 level as a geospatial point (i.e. a location that has lat/lon), a polygon, etc.
Resource	Resource is conceptually the base entity/notion. Anything that can be exchanged between providers and consumer at the time of an incident/emergency is a resource. Examples might include Search and Rescue (SAR) Teams, Mobile Camps, Construction Equipment, Blood Units, Vehicles, Tents, etc.
RMBaseEntity	The base meta-concept for some RM concepts including Incident, TimeEntity, LocationEntity, Resource.
AttributeBag	A generic attribute storage mechanism for being able to attach RMAttributes to the RMBaseEntity.
RMAttribute	Meta-concept that corresponds to attributes to be appended to the AttributeBag.
Receive	Is a specialization of 'Activity'. Conceptually maps to the BPEL's Receive meta-model entity, therefore carries the very same semantics as defined in BPEL metamodel. Is used to model reception of a (coordination) message from a sender.
Invoke	Is a specialization of 'Activity'. Conceptually maps to the BPEL's Invoke meta-model entity, therefore carries the very same semantics as defined in BPEL metamodel. Is used to model an 'invocation', which essentially means passign a message between 2 entities.
Reply	Is a specialization of 'Activity'. Conceptually maps to the BPEL's Reply meta-model entity, therefore carries the very same semantics as defined in BPEL metamodel. Is used to express sending a reply to a message.
EConfidentiality	The confidentiality level that belong to an AbstractRMMessage. Is actually an enumeration comprising the standards confidentiality levels (unclassified, restricted, … etc.).

INTEGRATED METAMODEL



DSL GRAMMAR

ResourceCoordinationFlow ::= (Process)*

Process ::= ProcessName Activity IncidentRef ProcessID

ProcessName ::= Identifier

IncidentRef ::= Identifier

ProcessID ::= Identifier

Flow ::= FlowName (Activity)*

FlowName ::= Identifier

Activity ::= ActivityName Flow Sequence (AbstractRMMessage)? Reply Invoke Receive

ActivityName ::= Identifier

Sequence ::= SequenceName (Activity)*

SequenceName ::= Identifier

AbstractRMMessage ::= AbstractRMMessageName (Activity)* TemporalCoverage GeospatialCoverage AbstractRMMessageID (ResourceRef)* Confidentiality

Reply ::= Target Source

Invoke ::= Target Source

Receive ::= Target Source

Target ::= Identifier

Source ::= Identifier

TemporalCoverage ::= TimeEntityName TimeEntity

DSL GRAMMAR (CONT.'ED)

GeospatialCoverage ::= LocationEntity

AbstractRMMessageID ::= Identifier

ResourceRef ::= Identifier

Confidentiality ::= ConfidentialityType

ConfidentialityType ::= Identifier

RMBaseEntity ::= Incident TimeEntity Resource LocationEntity AttributeBag

Incident ::= TemporalCoverage GeospatialCoverage

TimeEntity ::= Time

Resource ::= ResourceType

LocationEntity ::= Location

AttributeBag ::= (IncidentAttribute)*

IncidentAttribute ::= IncidentAttributeName

Time ::= Identifier

ResourceType ::= Identifier

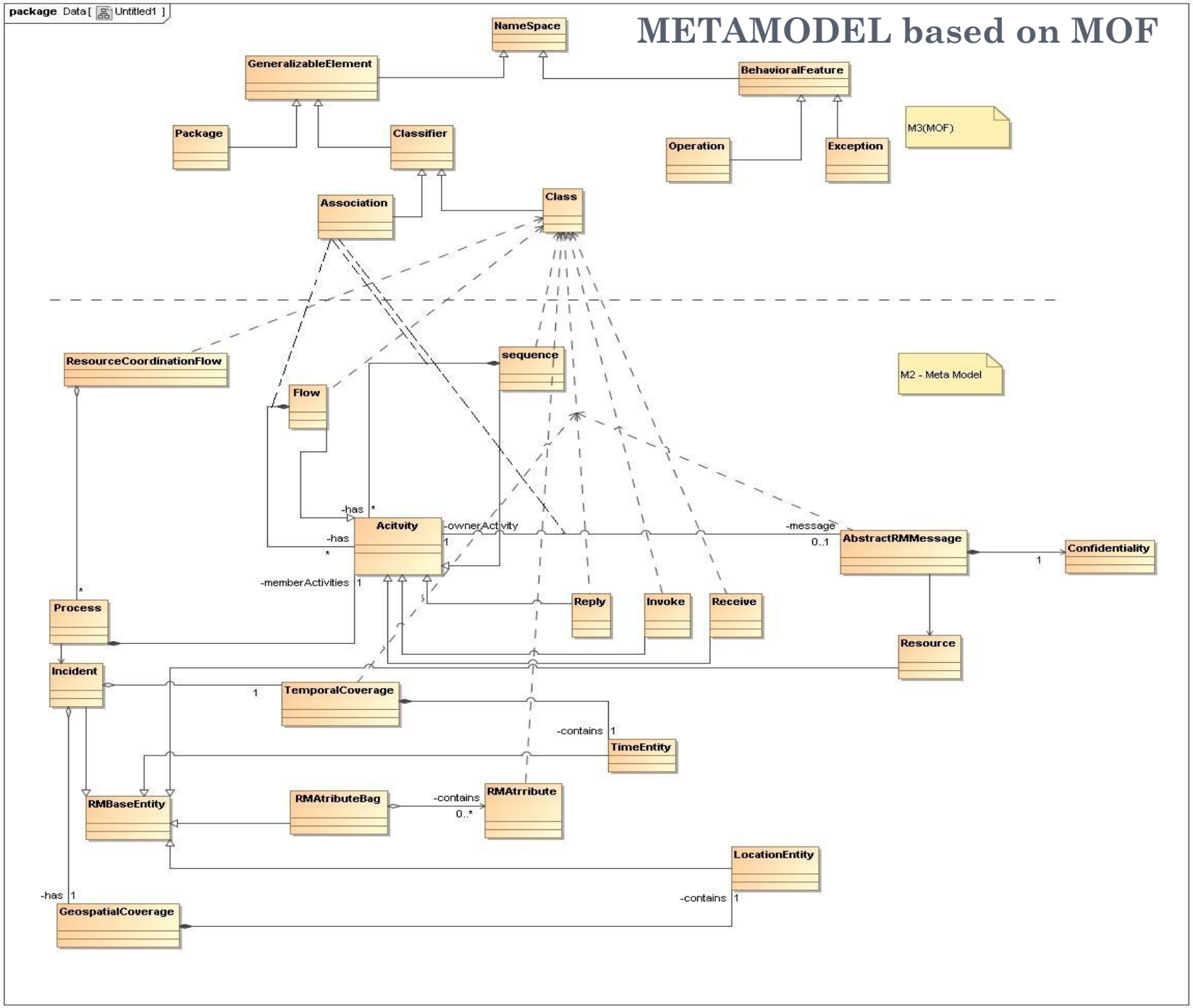
Location ::= Identifier

IncidentAttributeName ::= Identifier

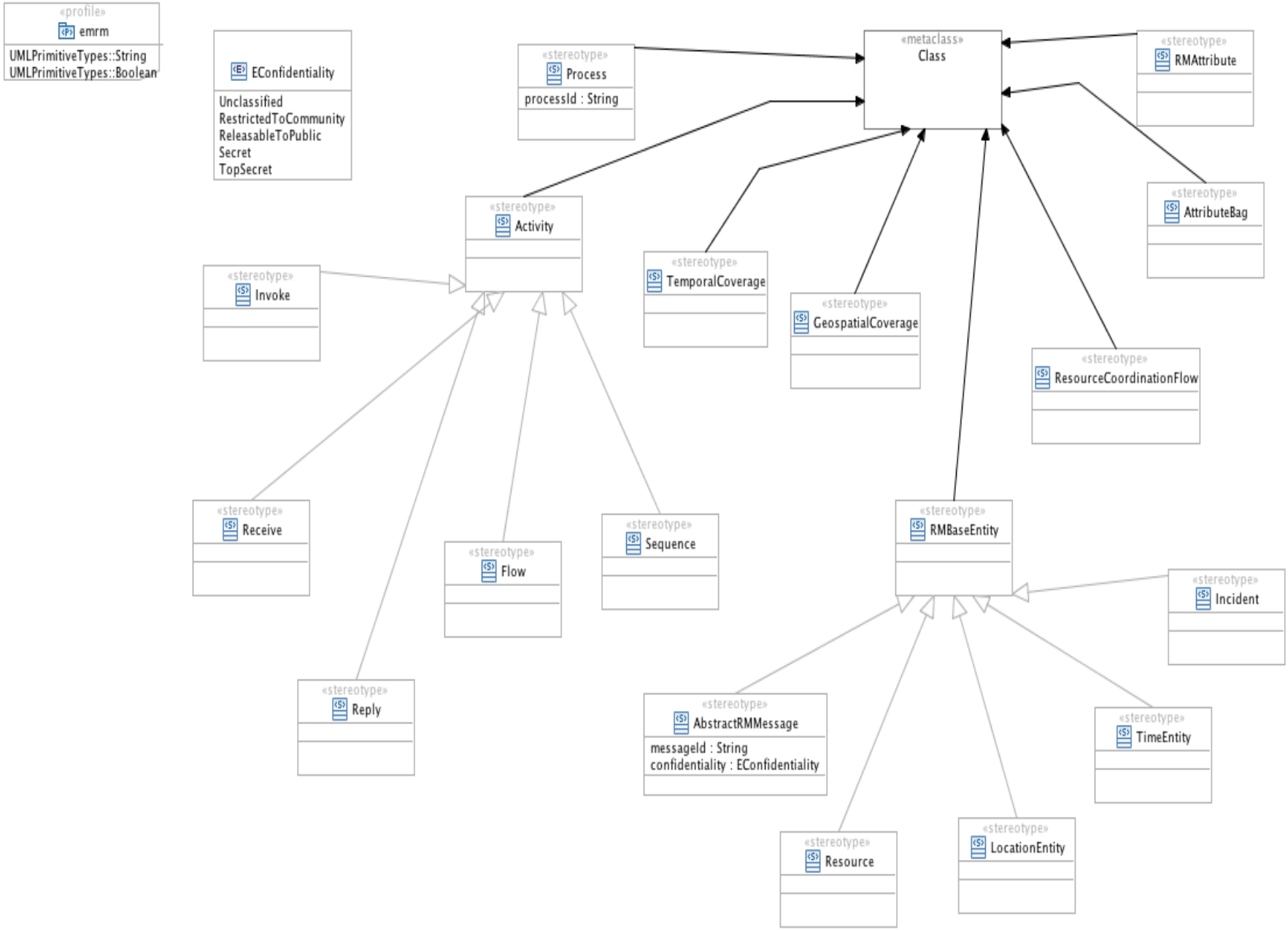
TimeEntityName ::= Identifier

AbstractRMMessageName ::= Identifier

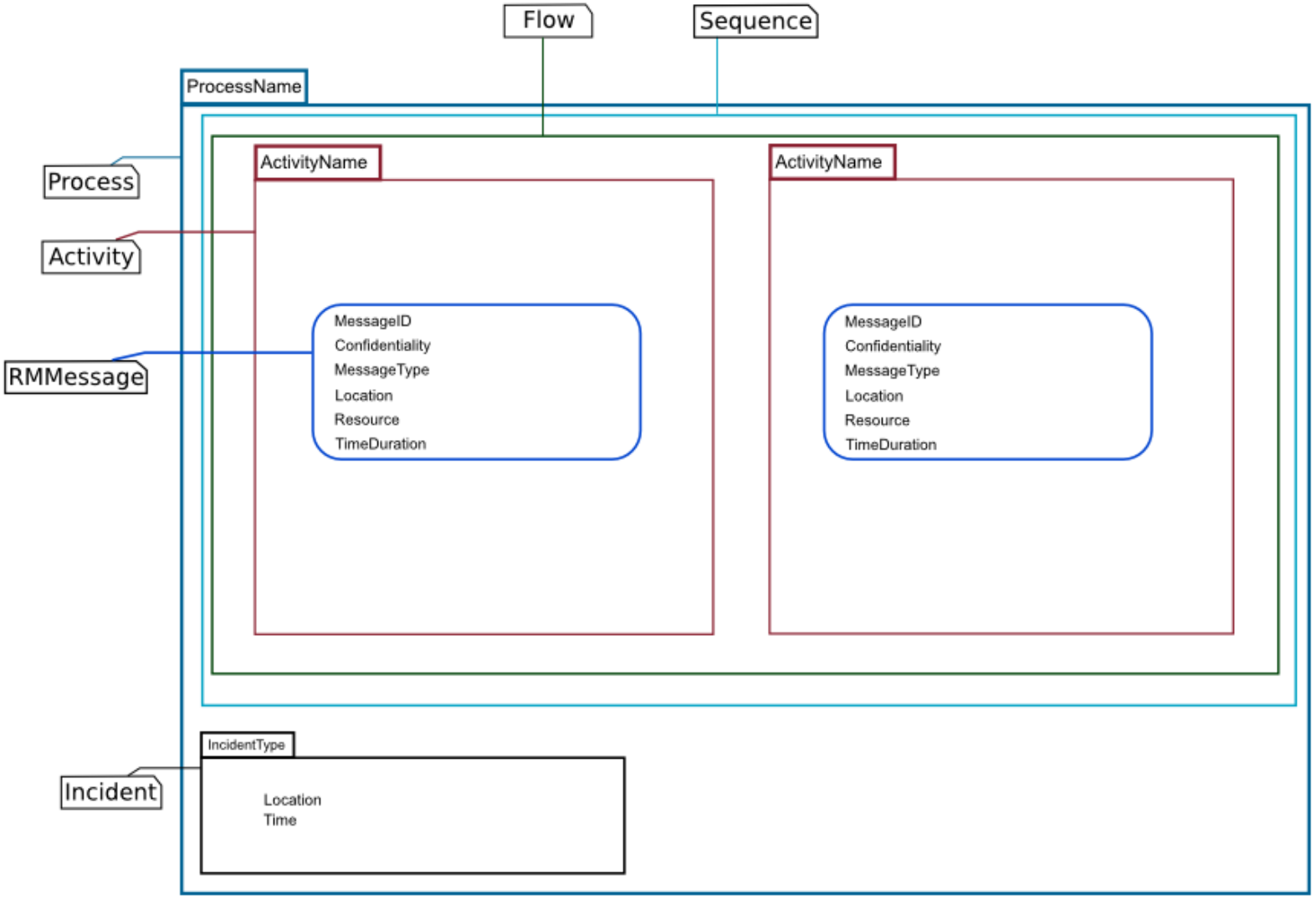
METAMODEL based on MOF



METAMODEL based on UML Profiling



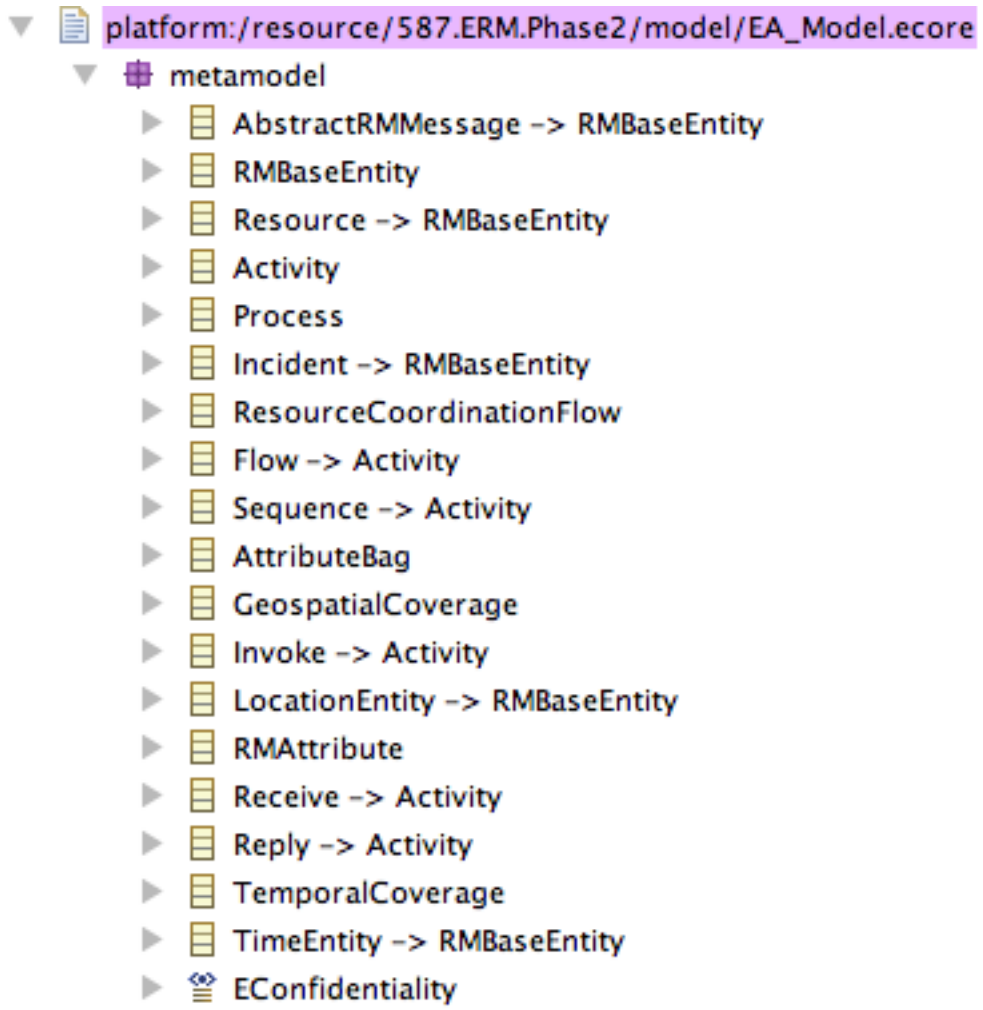
CONCRETE SYNTAX



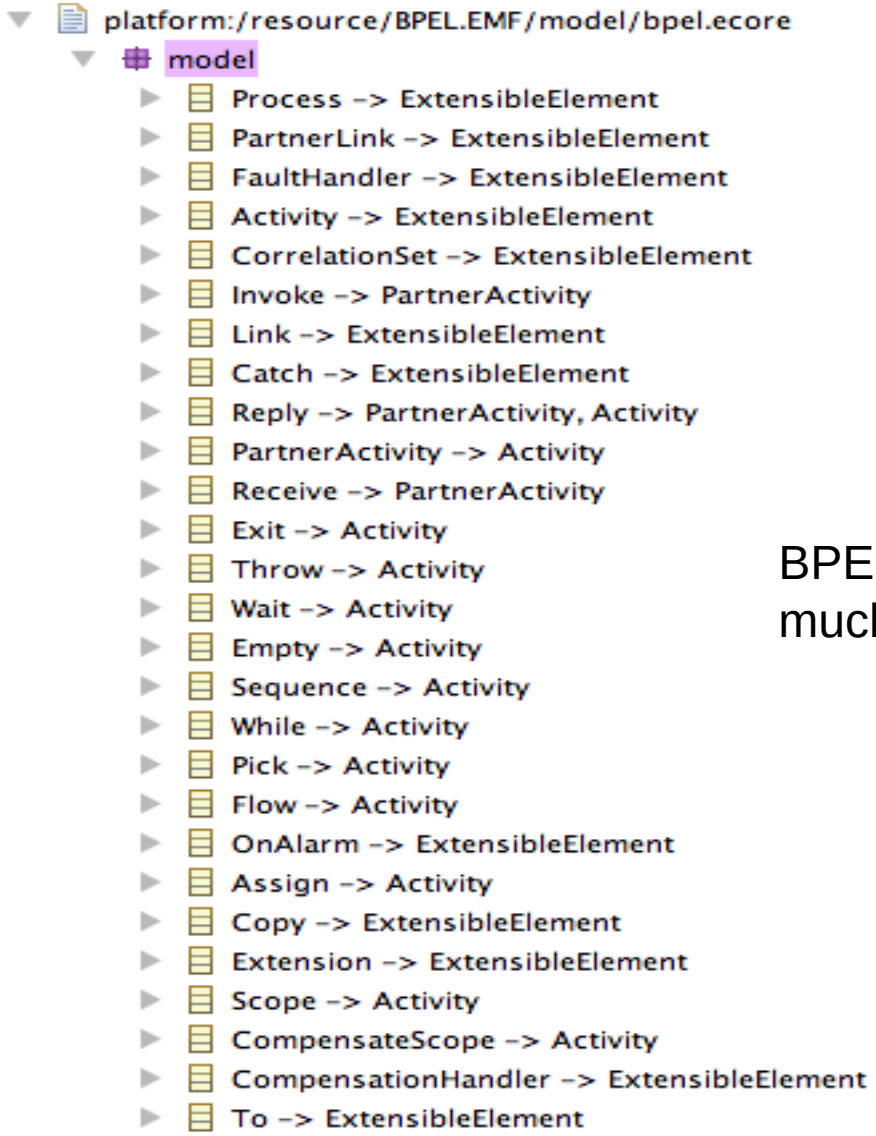
M2M TRANSFORMATION

- ATL
- Source metamodel ecore
- Target metamodel ecore
- ATL transformation rules
- Input model xpi (Emergency management)
- Output model xpi (BPEL)

M2M TRANSFORMATION (Source MM)



M2M TRANSFORMATION (Part of Target MM)



BPEL MM is actually much longer than this!

M2M TRANSFORMATION: Sample ATL Rules

```
module AltDeneme1; -- Module Template
create Out : Bpel from IN : Emergency;

rule ProcessMapping
{
	from
		a : Emergency!Process
	to
		p : Bpel!Process (
				name <- a.processId,
				activity <- a.activity
		)
}
rule FlowMapping
{
	from
		a : Emergency!Flow
	to
		p : Bpel!Flow
		(
				name <- a.activityId,
				activities <- a.activity
		)
}
```

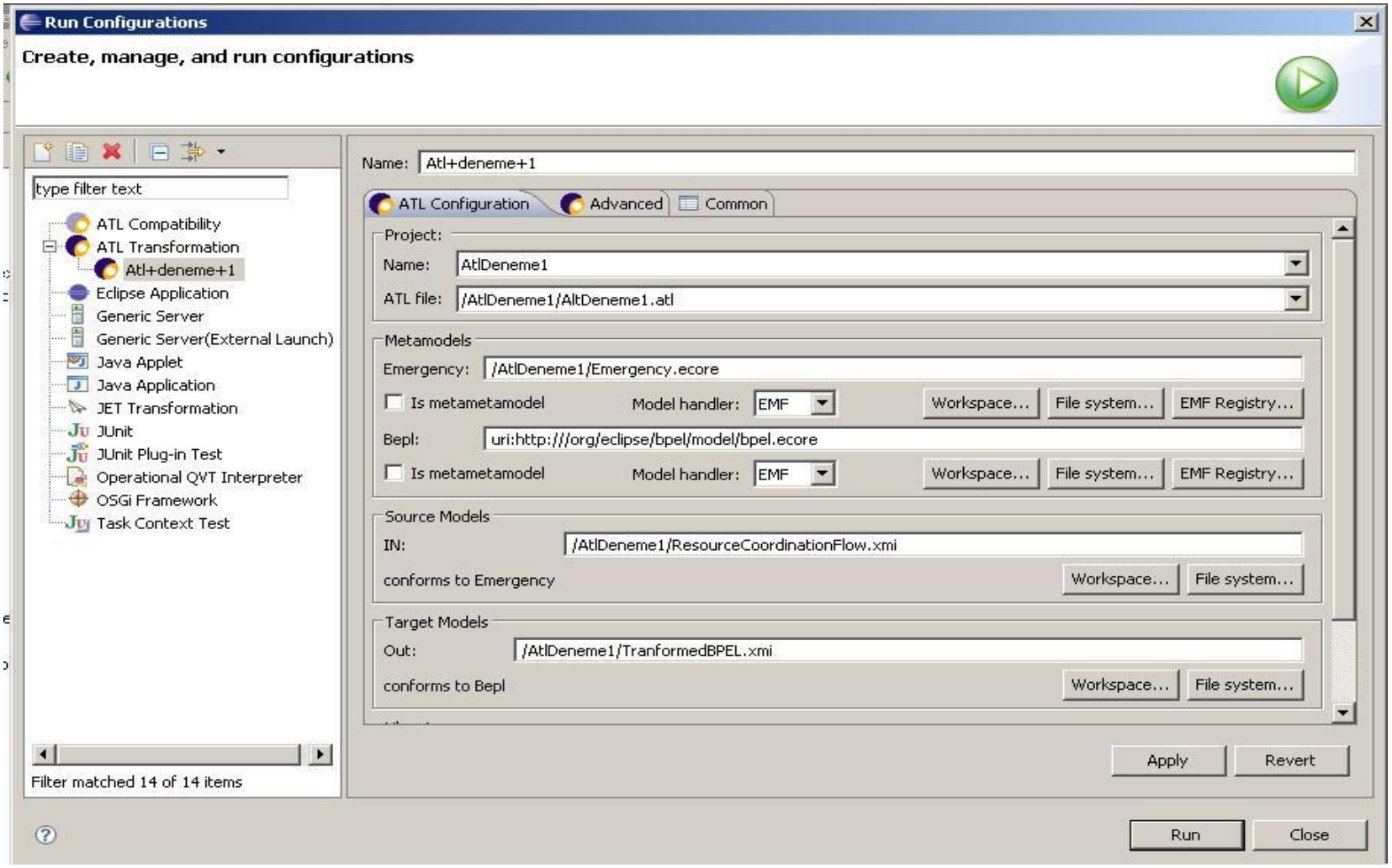
M2M TRANSFORMATION: Sample ATL Rules (Cont.'ed)

```
rule InvokeMapping
{
	from
		a : Emergency!Invoke
	to
		p : Bpel!Invoke
		(
			name <- a.activityId,
			inputVariable <- a.message
		)
}


rule ReceiveMapping
{
	from
		a : Emergency!Receive
	to
		p : Bpel!Receive
		(
			name <- a.activityId,
			variable <- a.message
		)
}
```

M2M TRANSFORMATION: Sample ATL Rules (Cont.'ed)

```
rule ReplyMapping
{
	from
		a : Emergency!Reply
	to
		p : Bpel!Reply
		(
			name <- a.activityId,
			variable <- a.message
		)
}

rule AbstractRMMessageMapping
{
	from
		a : Emergency!AbstractRMMessage
	to
		p : Bpel!Variable
		(
			name <- a.messageId,
			messageType <- a.messageType
		)
}
```

M2M TRANSFORMATION: ATL using Eclipse



M2T TRANSFORMATION

- openArchitectureWare
  - Template language for code generation Xpand
- Java code from ecore

- Example:

```
import org.eclipse.emf.ecore.EObject;

public interface Activity extends EObject {
    String getActivityId();
    void setActivityId(String value);
    String getActivityName();
    AbstractRMMessage getMessage();
    void setMessage(AbstractRMMessage value);

} // Activity
```

LESSONS LEARNED AND CONCLUSIONS

- 'DSL' is actually something we have been using for years. The course made us to think again about its meaning and uses.
- We have found Meta-Modeling conceptually not too easy to grasp. It takes a lot of time to start thinking in 'meta', while it is quite difficult to resist the temptation of doing modeling on the wrong level! (i.e. M1 in place of M2)
- Placing yourself in place of a tool developer helps a lot with meta-modeling
  - How would a model based on my metamodel be of use to a tool developer?
  - What would be the added value

LESSONS LEARNED AND CONCLUSIONS (CONT.'ED)

- Tooling seems problematic for Meta-Modeling.
  - We have started out with a commercial tool called Enterprise Architect (EA). We also had a trial version of MagicDraw (MD).
  - EA did not fulfill all the requirements of the assignments (especially transformation related). We started using a variety of tools (mostly Eclipse plug-ins like EMF, OAW, ATL).
  - We had to perform export/import between these tools; but there were inconsistencies between EA export, MD export and OAW although they all support XMI.

LESSONS LEARNED AND CONCLUSIONS (CONT.'ED)

- If MDSD will be a success, serious tool support is an absolute need. UML diagram editing tooling that we have is not enough.

- But in the end, we can say Eclipse Modeling Framework , seems to work OK in itself; although there is plenty of bugs. But, we did not come across any showstopper bugs after we switched to EMF.

LESSONS LEARNED AND CONCLUSIONS (CONT.'ED)

- Due complexity of our selected target meta-model (i.e. meta-model of WS-BPEL), we had difficulty devising how it would be best to import meta-concepts from it into our meta-model. Thus, in the end, our meta-model could best be described as 'influenced by' BPEL metamodel.
  - This has led to difficulties during model to model transformation.
- For model to model generation, because WS-BPEL is targeted for 'web-service' based compositions, we had to go back to the meta-model and make changes.

LESSONS LEARNED AND CONCLUSIONS (CONT.'ED)

- For model to code generation, we were planning to actually generate BPEL code, but instead we chose to generate java code based on EMF Ecore.
- In the end, we believe that a meta-model's success can best be verified by building a tool that supports it. Better, the more tooling and accepted models are available based on that meta-model, the more trust can have about it.

REFERENCES (Not exhaustive)

- Rui Chen, Raj Sharman, H Raghav Rao, and Shambhu J Upadhyaya. Coordination in Emergency Response Management. Communications Of The Acm, 51(5):8, Apr 2008 vol. 51
- OASIS WS-BPEL Technical Committee. Web services business process execution language version 2.0. pages 1–66, May 2007.
- OASIS Emergency Management TC. Emergency data exchange language resource messaging (edxl-rm) version 1.0. pages 1–174, Nov 2009.
- Jim Steel, Renato Iannella, and Ho-Pun Lam. Using ontologies for decision support in resource messaging. Proceedings of the 5 International ISCRAM Conference, page 9, Apr 2008
- David Aylward, Jim Bound, Bonnie Gorsic, Steve Gross, Walter LeGrand, Paul Mangione, Harrison Miles, Nelson Santini, Amin Soleimani, and John Yanosy. Findings and recommendations for mobile emergency communications interoperability (meci). NCOIC, page 152, Jan 2007.
- Eclipse ATL Project Model To Model Transformation, http://www.eclipse.org/m2m/atl
- OpenArchitectureWare, http://www.openarchitectureware.org/
- Markus Völter, Bernd Kolb, Best Practices for Model-To-Text Transformations, Sep 2006