LANCASTER UNIVERSITY

The Open University

Aspect-Oriented Requirements Engineering: An Introduction

Awais Rashid and Yijun Yu

AOSD-EUROPE
eu network of excellence

Acknowledgements

- Americo Sampaio, Ruzanna Chitchyan, Nathan Weston, Vander Alves, Paul Rayson, Alessandro Garcia and Peter Sawyer *(Lancaster University, UK)*
- Ana Moreira and Joao Araujo *(New University of Lisbon, Portugal)*
- Elisa Baniassad *(Chinese University of Hong Kong, China)*
- Shmuel Katz *(Technion-Israel Institute of Technology, Israel)*
- Monica Pinto and Lidia Fuentes *(University of Malaga, Spain)*
- Bashar Nuseibeh, Robin Laney (The Open University)
- Nan Niu, John Mylopoulos, Steve Easterbrook, Neil Ernst (University of Toronto)
- Julio Cesar Leite (PUC-Rio)
- Bruno Gonzales-Baixauli (Univ. of Valladolid)

Tutorial Outline

- Why aspect-oriented requirements engineering?
- A real-world case study
- AORE in action on the case study
  - Lancaster AORE tools
  - OU AORE tools
- Discussion with audience and wrap up

LANCASTER UNIVERSITY

The Open University

Concerns in Requirements

AOSD-EUROPE
eu network of excellence

“The first step towards the management of disease was replacement of demon theories and humours theories by the germ theory. That very step, the beginning of hope, in itself dashed all hopes of magical solutions. It told workers that progress would be made stepwise, at great effort, and that a persistent, unremitting care would have to be paid to a discipline of cleanliness. So it is with software engineering today.”

Fred Brookes, No Silver Bullet

Why do We Wish to Separate Concerns?

- **Modular Reasoning**
  - Keep concerns separated regardless of how they affect or influence various other concerns in the system, so that we can reason about each concern in isolation
- **Compositional Reasoning**
  - Understand the dependencies and interactions amongst concerns to reason about the global or emergent properties of the system

Typical Separation of Concerns Modes

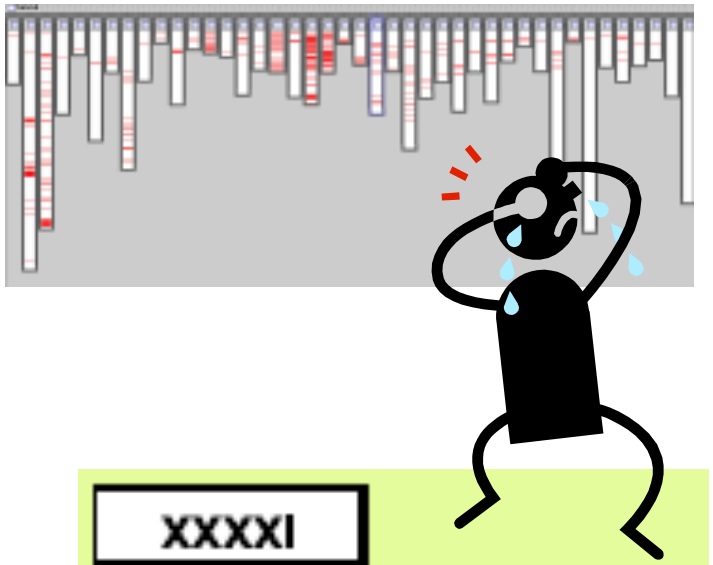
- Use cases
  - A step-by-step *usage path* through a system
- Viewpoints
  - A slice of each concern relevant to the particular stakeholder
- Goals
  - Hierarchical decomposition and operationalisation based on goals and softgoals
- Your favourite technique

The Problem of Crosscutting Concerns

- The dominant decomposition
  - Slice requirements on the lines of viewpoints, use cases and goals
- A number of concerns do not naturally fit these dominant decomposition boundaries

Scattering and Tangling

- **Scattering**
  - The specification of one property is **not encapsulated** in a single requirements unit, e.g., a viewpoint, a use case.
- **Tangling**
  - Each requirements unit contains descriptions of **several properties** or different functionalities



**“When a wise man, established well in virtue,
Develops consciousness and understanding,
Then as a bhikku ardent and sagacious,
He succeeds in disentangling this tangle.”**

Buddhaghosha (Visuddhimagga)

Aspect-Oriented Requirements Engineering

- Improved support for separation of crosscutting functional and non-functional properties during requirements engineering
  - **Improved understanding of the problem and ability to reason about it**
- Ide
  as
  - derived
- Trace aspectual requirements and their trade-offs to architecture and subsequently all the way to implementation

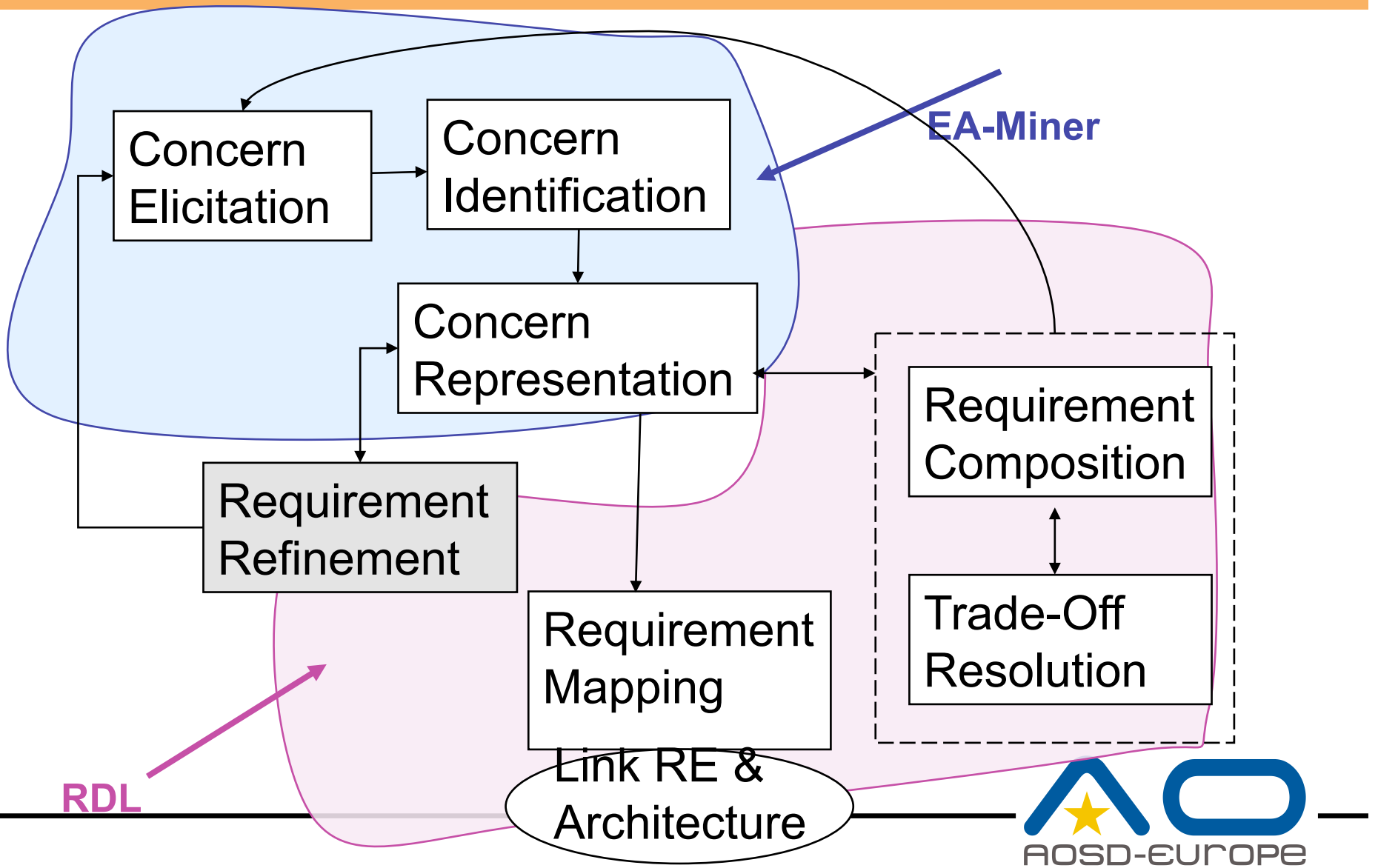
Case Study: HealthWatcher

A testbed for AOSD

- http://www.comp.lancs.ac.uk/~greenwop/tao
- An information system for online health complaints handling
- It was implemented
- Requirements documented as use cases
  - http://www.comp.lancs.ac.uk/~greenwop/tao/aore_modelv2.pdf
- How do we apply AORE?

Lancaster AORE Toolset

Lancaster AORE Toolset



Identifying Aspects

Requirements Sources

- Requirements documents tend to be unstructured
- Requirements can take a variety of forms
  - Natural Language
  - Interviews
  - Sketches
  - Standardisation documents
  - Organisational procedures
  - ……

Aspect Identification Challenges

- Time consuming and error prone activities
  - Requirements elicitation and analysis
    - How to identify the concerns (or aspects, base abstractions, etc.)?
    - How the concepts are related?
    - How to structure them into a model?
- Why?
  - Replication of requirements (crosscutting) and unrelated responsibilities (tangling) are difficult to find and separate
  - Documents can be of varied structure and nature (e.g., user manuals, legacy specifications, e-mails, etc.)
  - Size matters (e.g., 5,000 words document takes 20 minutes just to read)
- Therefore
  - Tool support is vital for identifying concerns and structuring them into a suitable model for requirements analysis

The EA-Miner Tool

- Provides automation support for concern identification
- Uses Natural Language Processing (NLP) to reason about the properties and semantics of requirements expressed in natural language
- Can be used with a range of AORE models (currently viewpoints and multi-dimensional models are supported)
- Reduce the complexity (effort) of concern identification
  - Not aimed at automating 100% of the tasks

How EA-Miner Identifies Viewpoints and Aspects?

- Based on Heuristics and NLP properties
  - For the Viewpoint-based AORE model, EA-miner identifies:
    - Viewpoints (based on POS tag): Nouns are the Viewpoint candidates
    - Non-functional Aspects: (Based on lexicon of NFRs and semantic tags): candidates are words found in the lexicon whose semantic tag is of interest
    - Functional Aspects: candidates are action verbs who are referenced in the requirements of several viewpoints.

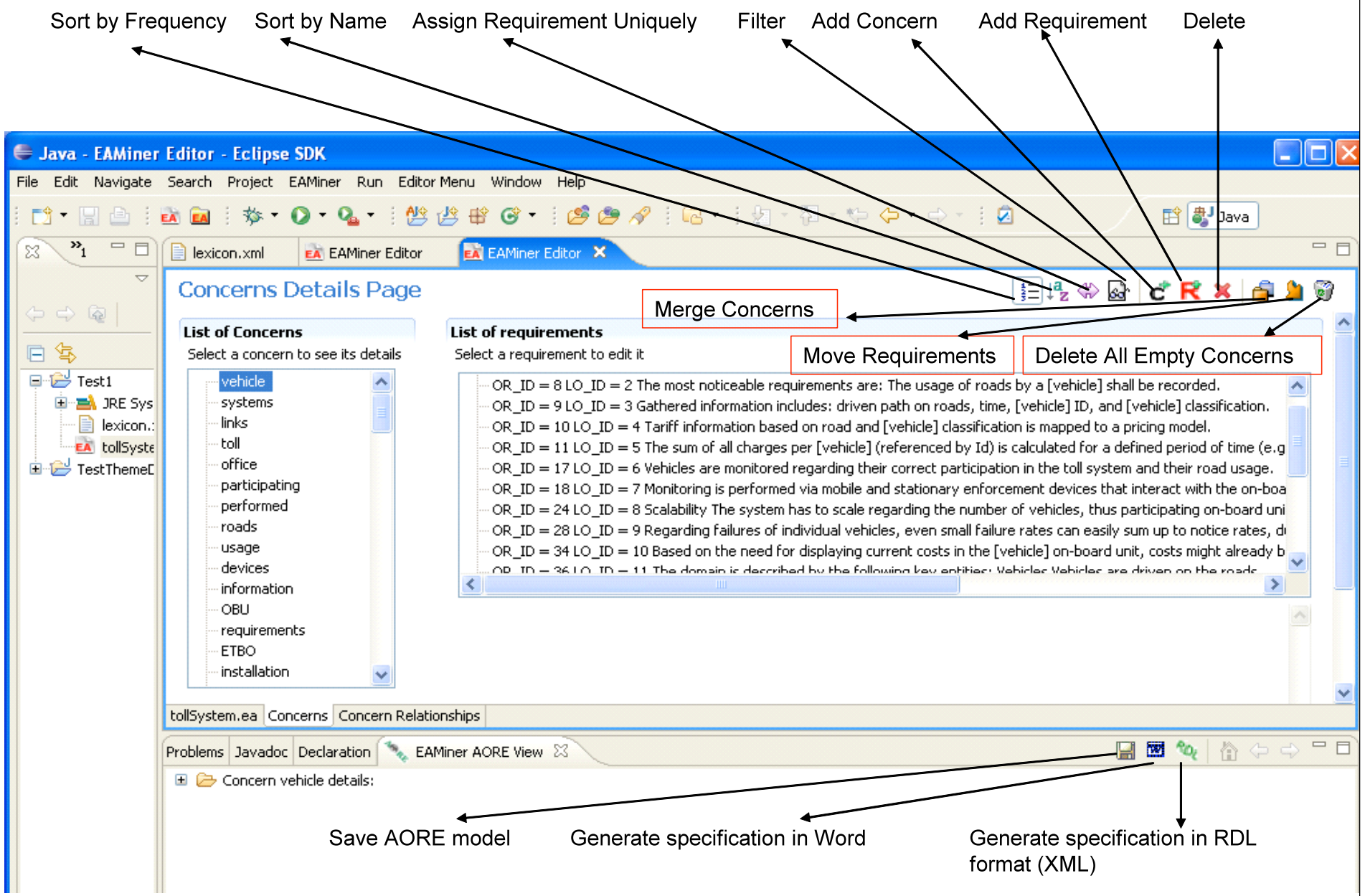
The NFRs lexicon

- Used for the identification of non-functional aspects
- Semantic tags are important for analysing the context
  - Performance:
    - Meaning1: How well a football player performed
    - Meaning2: A distributed system property
- The lexicon was initially built re-using existing knowledge (NFR catalogues, domain experts)
  - Lexicon entries:

<word content="authorise" sem="S7.4+" nfr="security"> </word>
<word content="performance" sem="K4" nfr="performance"> </word>

  - Tool will suggest automatic updates in the future

WMATRIX SEM class that groups words related to permission


Sort by Frequency
Sort by Name
Assign Requirement Uniquely
Filter
Add Concern
Add Requirement
Delete
Java - EAMiner Editor - Eclipse SDK
File Edit Navigate Search Project EAMiner Run Editor Menu Window Help
lexicon.xml
EAMiner Editor
EAMiner Editor
Concerns Details Page
Merge Concerns
Move Requirements
Delete All Empty Concerns
List of Concerns
Select a concern to see its details
List of requirements
Select a requirement to edit it
vehicle
systems
links
toll
office
participating
performed
roads
usage
devices
information
OBU
requirements
ETBO
installation
OR_ID = 8 LO_ID = 2 The most noticeable requirements are: The usage of roads by a [vehicle] shall be recorded.
OR_ID = 9 LO_ID = 3 Gathered information includes: driven path on roads, time, [vehicle] ID, and [vehicle] classification.
OR_ID = 10 LO_ID = 4 Tariff information based on road and [vehicle] classification is mapped to a pricing model.
OR_ID = 11 LO_ID = 5 The sum of all charges per [vehicle] (referenced by Id) is calculated for a defined period of time (e.g
OR_ID = 17 LO_ID = 6 Vehicles are monitored regarding their correct participation in the toll system and their road usage.
OR_ID = 18 LO_ID = 7 Monitoring is performed via mobile and stationary enforcement devices that interact with the on-boa
OR_ID = 24 LO_ID = 8 Scalability The system has to scale regarding the number of vehicles, thus participating on-board uni
OR_ID = 28 LO_ID = 9 Regarding failures of individual vehicles, even small failure rates can easily sum up to notice rates, d
OR_ID = 34 LO_ID = 10 Based on the need for displaying current costs in the [vehicle] on-board unit, costs might already b
tollSystem.ea
Concerns
Concern Relationships
Problems
Javadoc
Declaration
EAMiner AORE View
Concern vehicle details:
Save AORE model
Generate specification in Word
Generate specification in RDL format (XML)

EA-Miner in Industrial Setting

- A number of empirical studies have been conducted
- Applied to parts of a real-world Toll Collection at Siemens
- Uncovered key architectural drivers missed by developers of the AO system

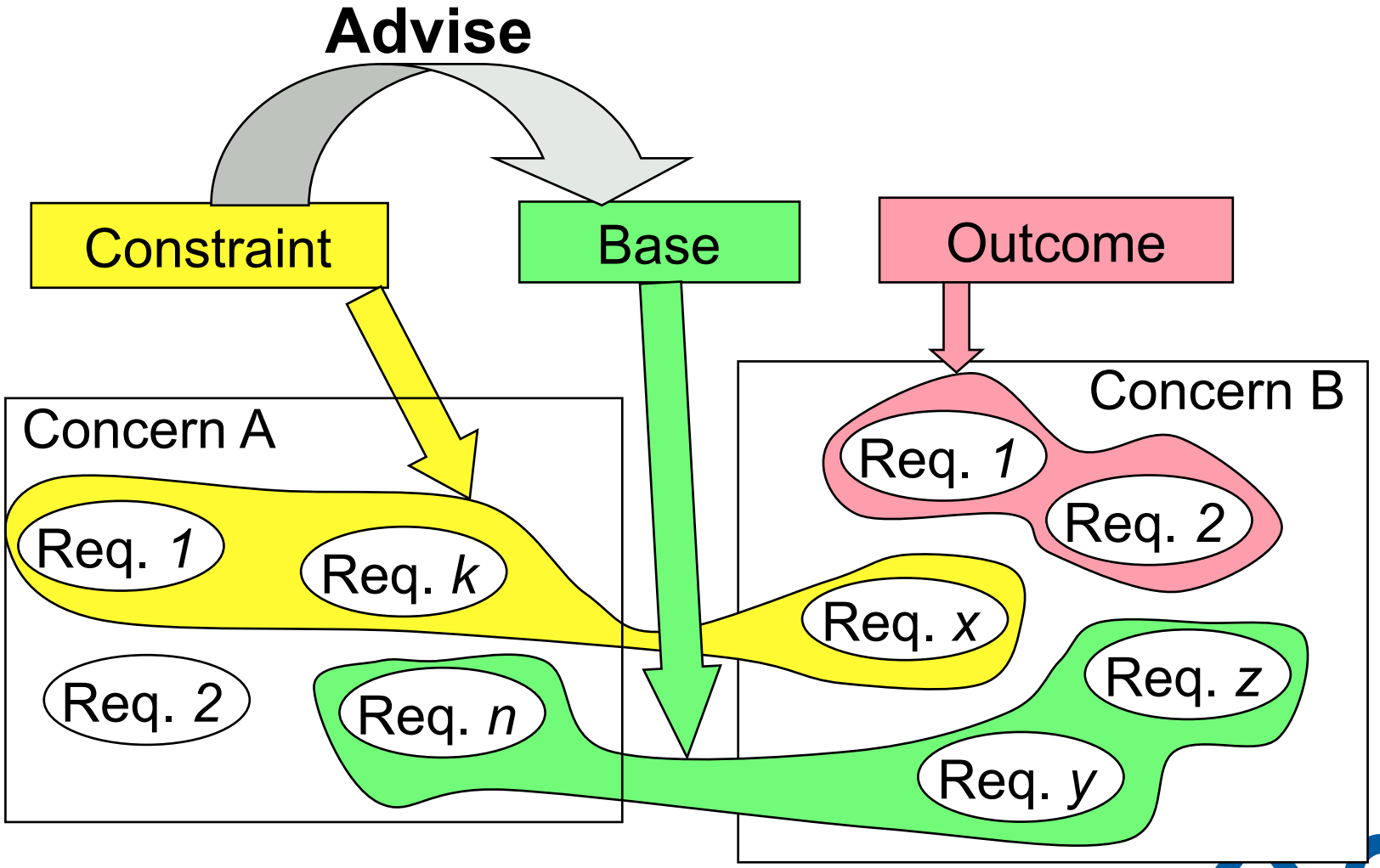
From EA-Miner to RDL

- Requirements Description Language (RDL)
- EA-Miner generates concerns in RDL and hints for compositions
- Requirements engineers specifies the compositions in the RDL

RDL

- A symmetric model for concern analysis
  - All concerns (non-crosscutting and crosscutting) specified as concerns
  - Compositions specify and capture the crosscutting relationships

RDL Composition Elements



Beyond Syntactic Compositions

LANCASTER UNIVERSITY

The Open University

- Composition support is generally lacking in RE techniques
- Where available compositions are specified based on labels
  - E.g., use case names, use case step numbers
- Must define each point which may participate in composition explicitly in some way

Example: HealthWatcher

- Use case Register Tables specifies the *Health Unit* table
- *Health unit* is referred to by a number of use cases, e.g., *Update Health Unit*

Problems of Syntactic Compositions

- Map developer's intentionality onto a syntax governed structural model
- Infer semantic influences and interferences from syntactic compositions
- Composition fragility

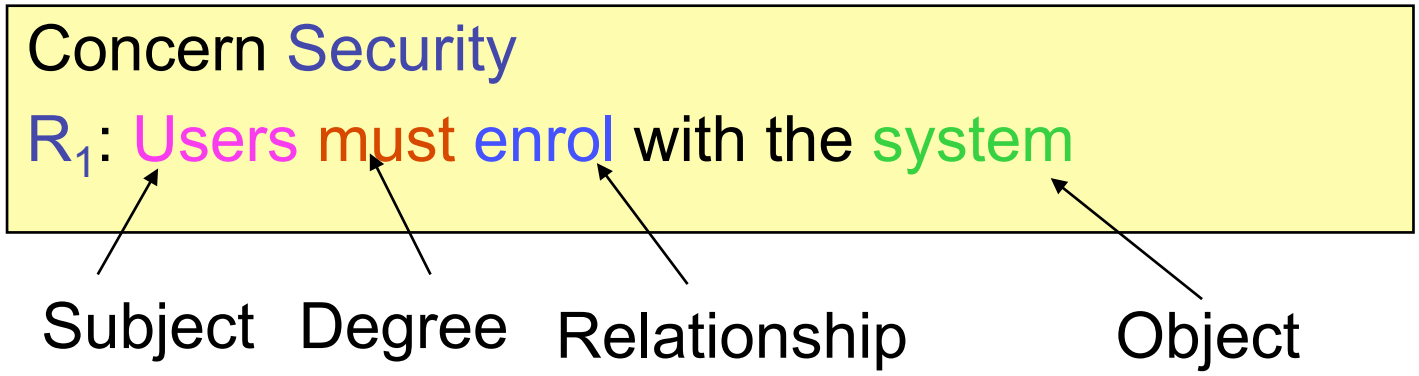
Semantics-based Composition with the RDL

LANCASTER UNIVERSITY

The Open University

- Make natural language requirement semantics more explicit and base composition on these semantics.
- How?
  - Use grammatical information
  - Use semantic grouping from linguistics
  - Use natural language processing to help automation

RDL Concern Example




```
<Concern name="Security">
  <Requirement id="1">
    <Subject> Users </Subject>
    <Degree type="Modal"  semantics="Modal" level="high"> must </Degree>
    <Relationship type="Move" semantics="Group">enrol </Relationship>
     with the
    <Object> system </Object>
    </Requirement>
</Concern>
```

Semantic Queries

Query:

Select the requirements where user enrols or a system is enrolled to.

RDL Query

all requirements where (Subject="users" and

Relationship="enrol") or (Object="system" and

Relationship="enrol")

Also used in Queries for matching:
Synonyms: register, sign up, join…
Lemmas: enrol, enrolled, enrolling..

RDL Composition Example

Composition:

Apply the requirements where user enrols to a system before any citizen registers a complaint.

```
<Composition name="AuthenticatedComplaint">
  <Constraint operator="apply"> subject= "user" and relationship=
  "enrol" and object="system" </Constraint>
  <Base operator="before"> subject="citizen" and
relationship="register" and object="complaint" </Base>
  <Outcome operator="satisfied"/>
</Composition>
```

Other Types of Queries

Based on Verb type or degree of importance, etc.

Composition:

Constraint requirement *“Include requirements into 1st release”* should be applied for

all requirements which have Modal verbs of high importance.

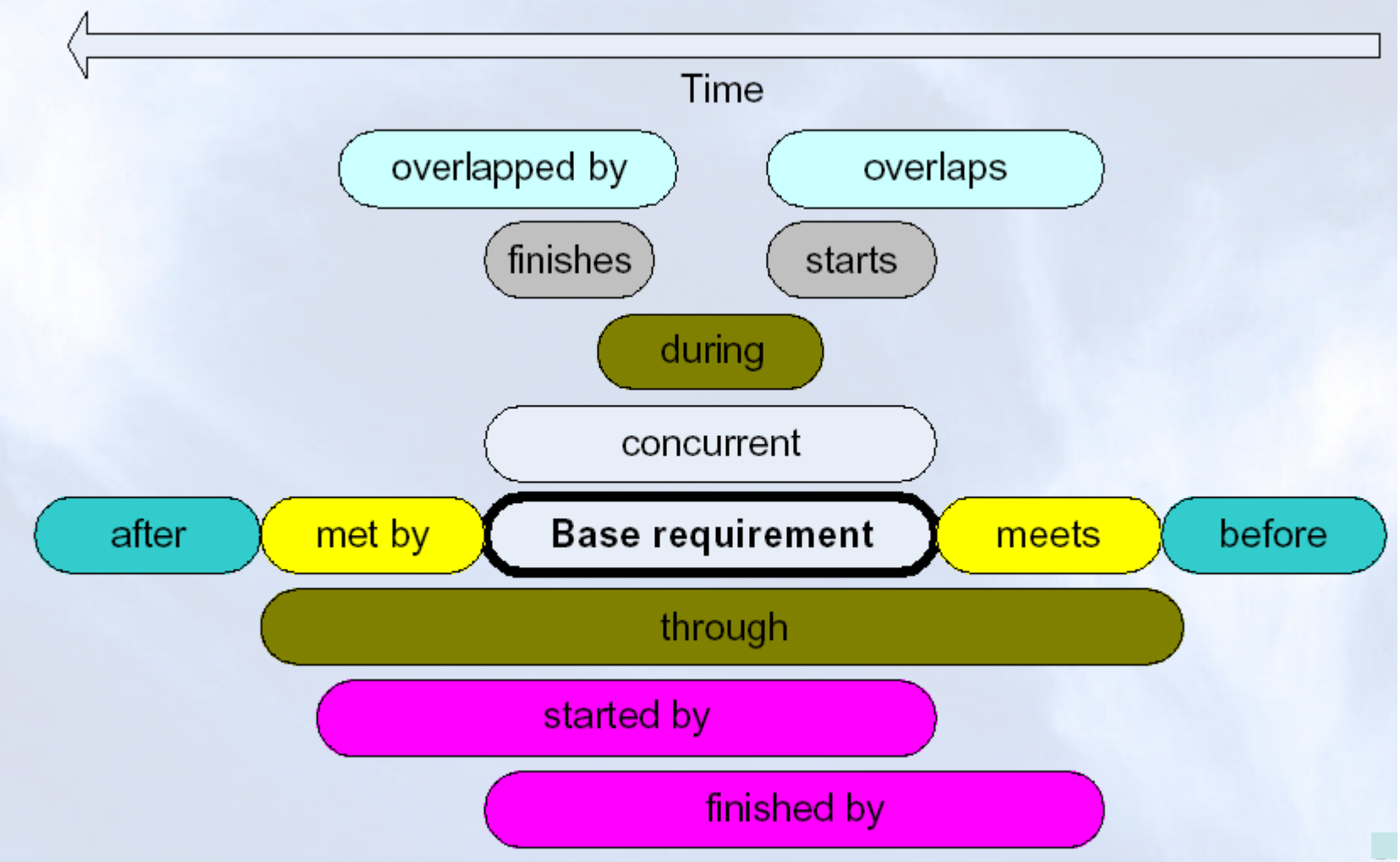
<Composition name=“CompFirstRelase”>

<Constraint operator=“include”> *“Include requirements into 1st release”* </Constraint>

<Base operator=“if”> degree.semantics=“Modal” and degree.level=“high”</Base>

</Concern>

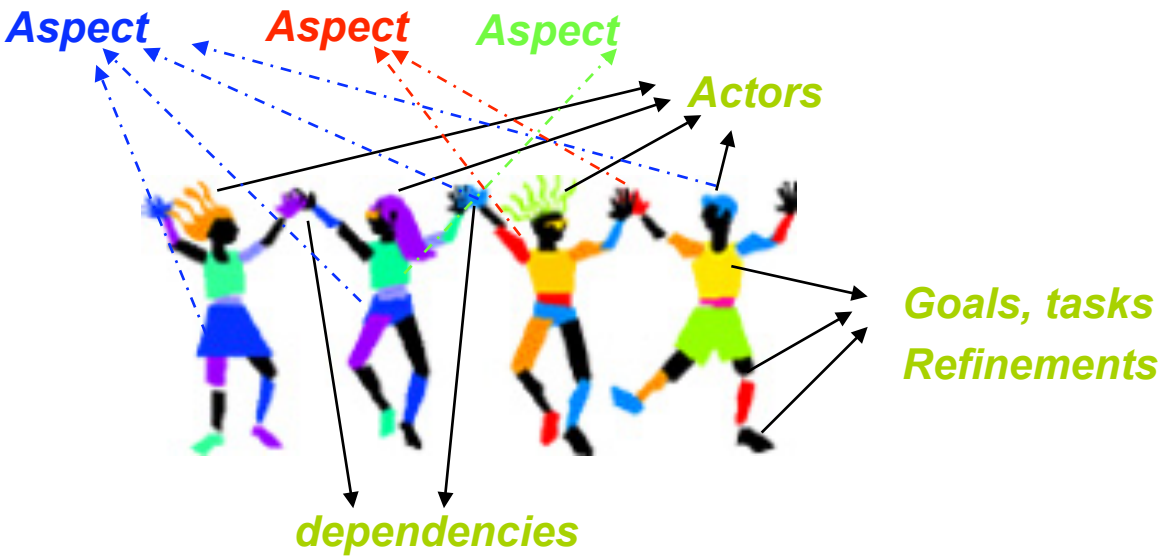
Analysing Dependencies



Automation Support: Wmatrix

- Wmatrix Natural Language Processor
  - Semantic Category Annotation
  - Part of Speech (POS) Annotation
  - Using the POS tags form rules for Subject, Relationship, and Object identification (continuously improving)
  - Map RDL verb classes onto Wmatrix native classification (continuously improving)
- Used as the backend of EA-Miner, which generates the annotated files for analysis tools

OU AORE toolset



Tools

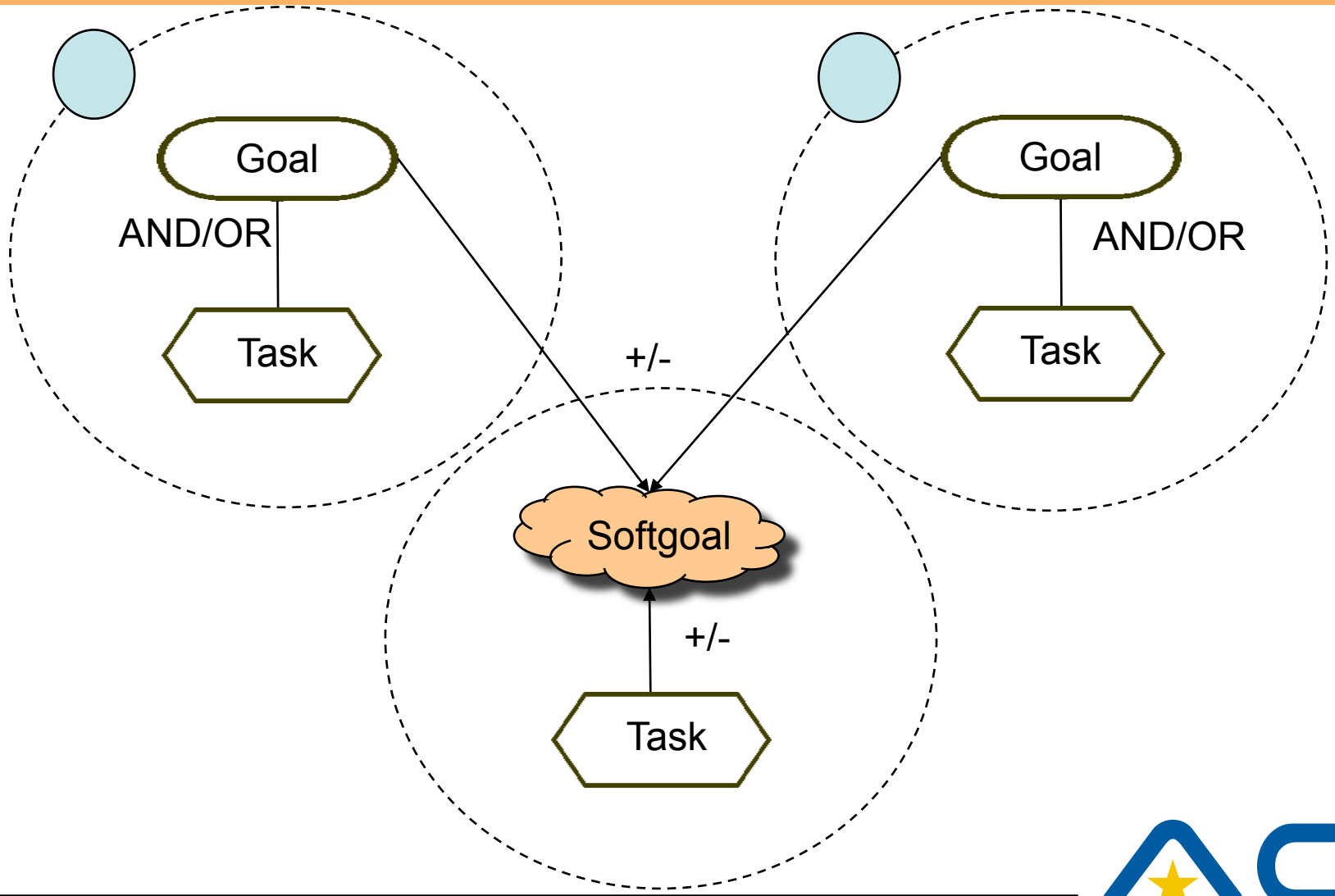
- GORE (Goal-oriented) RE tools
  - OpenOME: an i* modelling tool (with Univ. of Toronto)
    https://se.cs.toronto.edu/trac/ome/wiki
    http://mcs.cs.toronto.edu/yy66/tool/java/istar
- PORE (Problem-oriented) RE tools
  - Problem frames modelling tool
    http://mcs.cs.toronto.edu/yy66/tool/java/pf
- AORE tool support
  - V-graph and Q7

Y. Yu, N. Niu, B. Gonzales-Baixauli, W. Candillon, J. Mylopoulos, S.M. Easterbrook, J.C.S.P. Leite and G. van Wormhoudt. Tracing and Validating Goal Aspects In: *the 15th International Conference on Requirements Engineering (RE'07)*. pp. 53-56, 2007

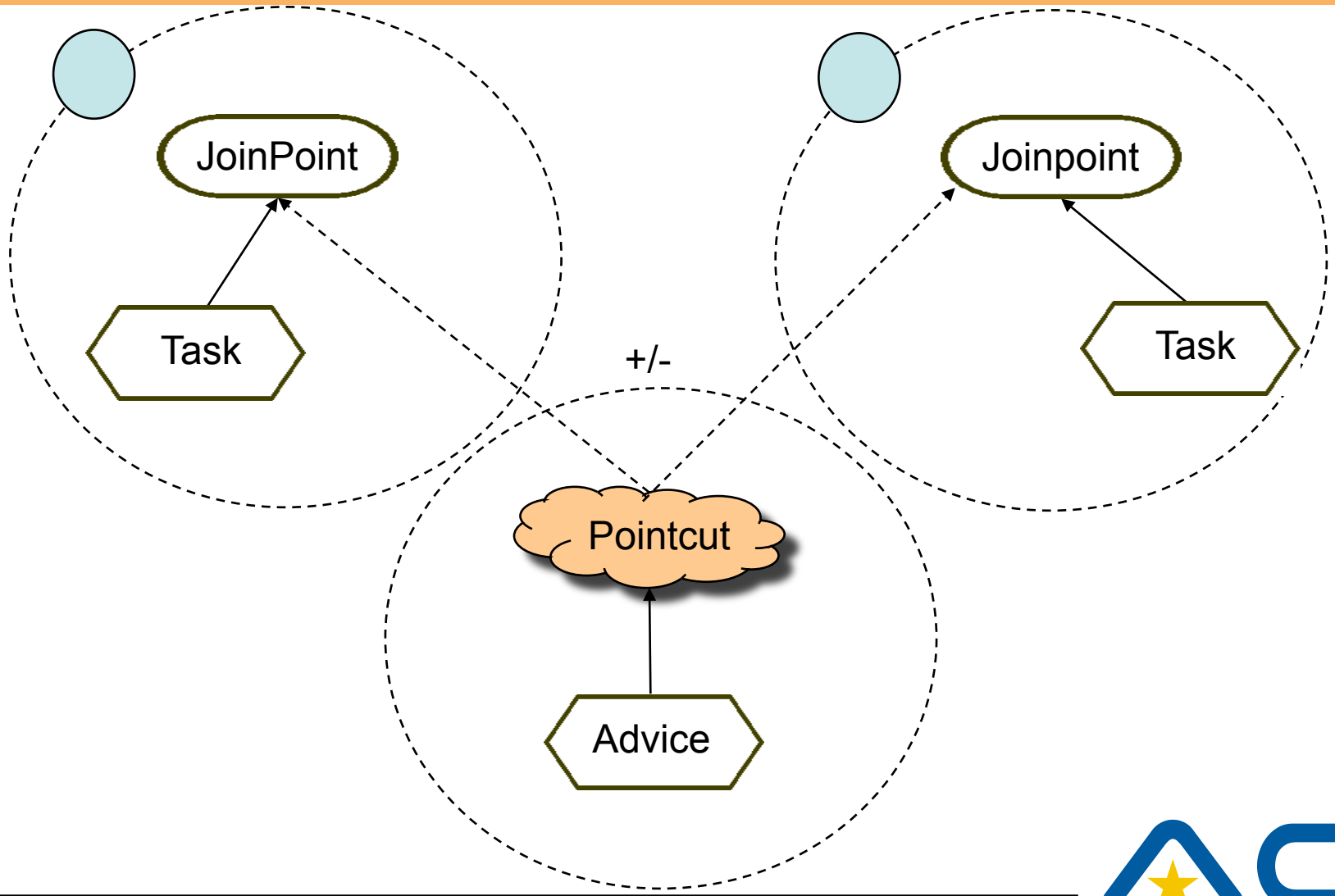
J.C.S.P. Leite, Y. Yu, L. Liu, E.S.K. Yu and J. Mylopoulos, Quality-based Software Reuse, In *Proceedings of CAiSE'05*. pp. 535-550. 2005

Y. Yu, J.C.S.P. Leite, J. Mylopoulos. From goals to aspects: discovering aspects from requirements goal models. In: *Proceedings of the 12th IEEE International Requirements Engineering Conference (RE'04). Sep. 6-10, pp. 38-47, 2004.*

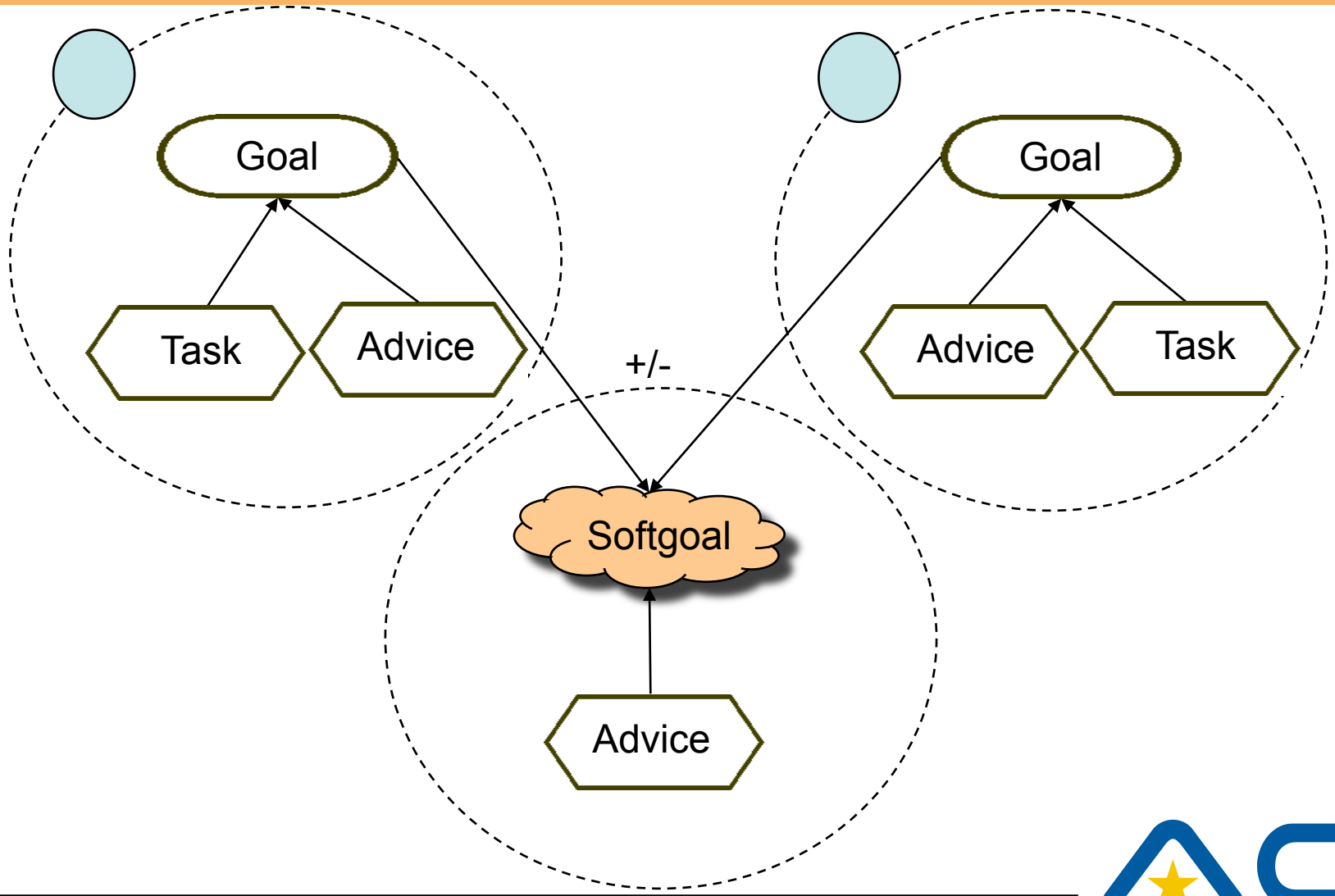
V-graph in i*



Weaving



Weaving



Q7 syntax/semantics

model := advice *
advice:= when () => <who>::why [what] { how } => how_much
how := contribution advice *
how_much := correlation <who>::why [what]
contribution := AND | OR
correlation := + | - | ++ | - - | ~ | AND | OR

- Why-How (recursive refinements, requirement/machine)
- Who (stakeholders)
- What (subject matters, world and machine domains)
- How-Much (satisfaction/validation criteria, logical justifications)
- When (context-aware conditions at runtime, context)

Charles B. Haley, Robin C. Laney, Bashar Nuseibeh: Deriving security requirements from crosscutting threat descriptions. AOSD 2004: 112-121

Q7 syntax/semantics (cont.)

advice :=

when() => <who>::why [what] <= where { how } => how_much

where := (contribution| correlation <who>:: why [what]) *

– *where* expresses point-cuts to match join-points in the requirements model
– wildcards can be used in such expressions
– supports all operators in i*

HealthWatcher

- How do we apply requirements aspect approach?

1. List stakeholders
2. List concerns of stakeholders
3. Refine and relate concerns
4. Define crosscutting concerns
5. Separate crosscutting concerns
6. Weave crosscutting concerns

- Any new findings from the application?

1. List stakeholders

- <Citizen>, e.g., potential patients
  - Any person who wishes to interact with the system.
- < Employee>
  - Health System employee, e.g., doctors, nurses
- <SystemOwner>, e.g., data enterers
  - Health System maintainer

2. List concerns

- E.g. Responsiveness
  - The response time must not exceed 5 seconds to log the system.

    <SystemDeveloper>::Log [system] => -- “time longer than 5s”
  - The response time must not exceed 5 seconds to submit a complaint specification.

    <Citizen>::Submit [complaints] => -- “time longer than 5s”
  - The response time must not exceed 5 seconds to submit a query information.

    <SystemOwner>::Submit [query information] => -- “time longer than 5s”

3. Refinement and correlation

Example of goal refinement and correlations

- Ask "why" about the tasks

```
<Citizen>::Specify [complaints] { AND
    Enter [complaints]
    Submit [complaints] => -- "time longer than 5s"
}
```

- Ask "how" and "how much" about the goals

```
<SystemDeveloper>::Log [system] { AND
    Log [query information] => -- "time longer than 5s"
    Log [submit complaints] => -- "time longer than 5s"
} => -- "time longer than 5s"
```

- Ask "when" about the evaluation of goal/task satisfaction (validation)

```
"query information is correctly specified" ()
    <SystemOwner>::Submit [query information] => - - "time longer than 5s"
! "query information is correctly specified" ()
    <SystemOwner>::Specify [query information]  {
        <SystemOwner>::ReportError [ query information] => - - "time longer than 5s"
        <Citizen>::Specify [query information]
        <SystemOwner>::Submit [query information] => - - "time longer than 5s"
    }
```

4. Crosscutting concerns

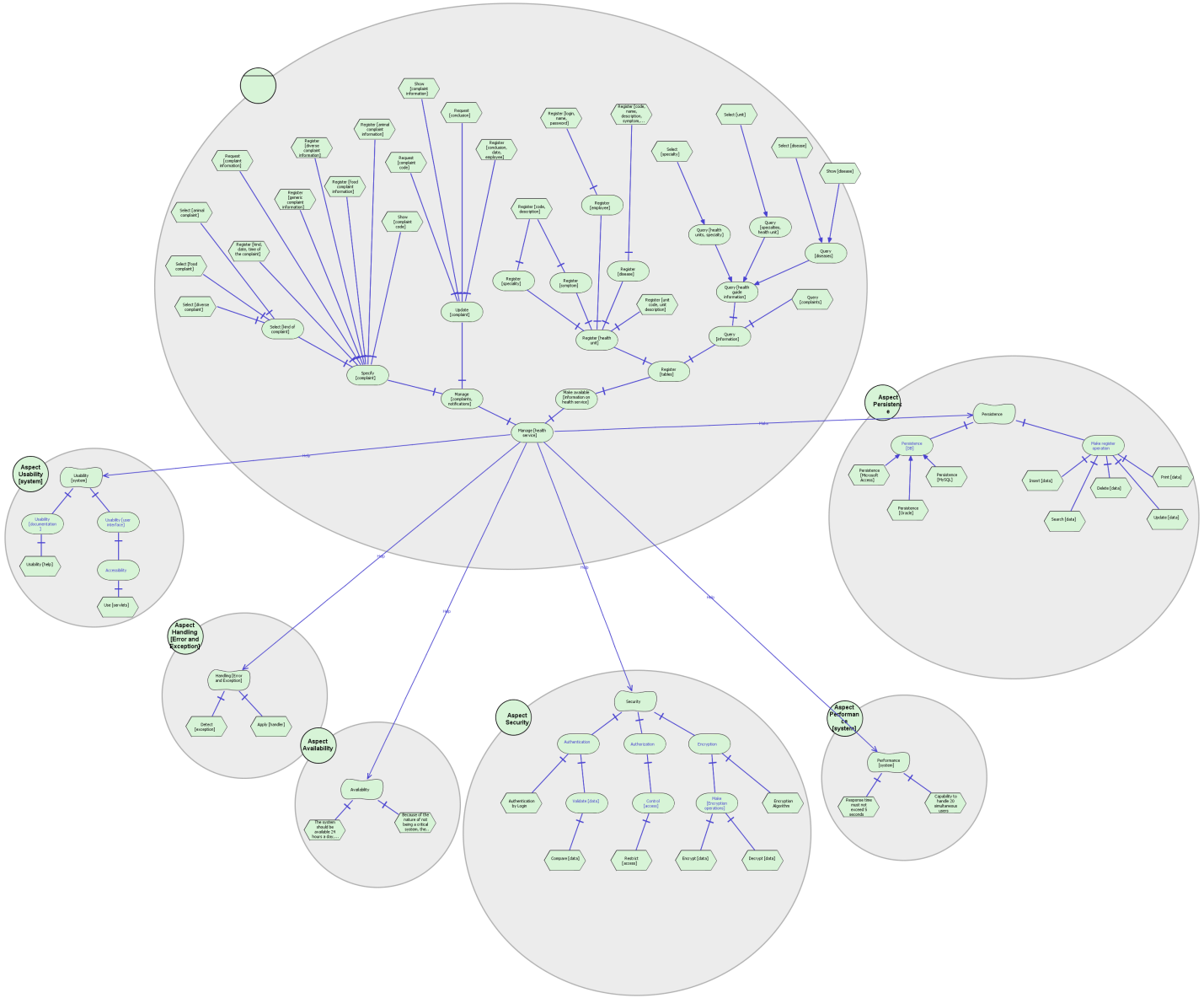
- Enumerate joinpoints: any place where "time longer than 5s" is to be avoided

  "time longer than 5s" <= -- <Citizen>::Submit [complaints],
  -- <SystemDeveloper>::Log [system],
  -- <SystemOwner>::Submit [query information]

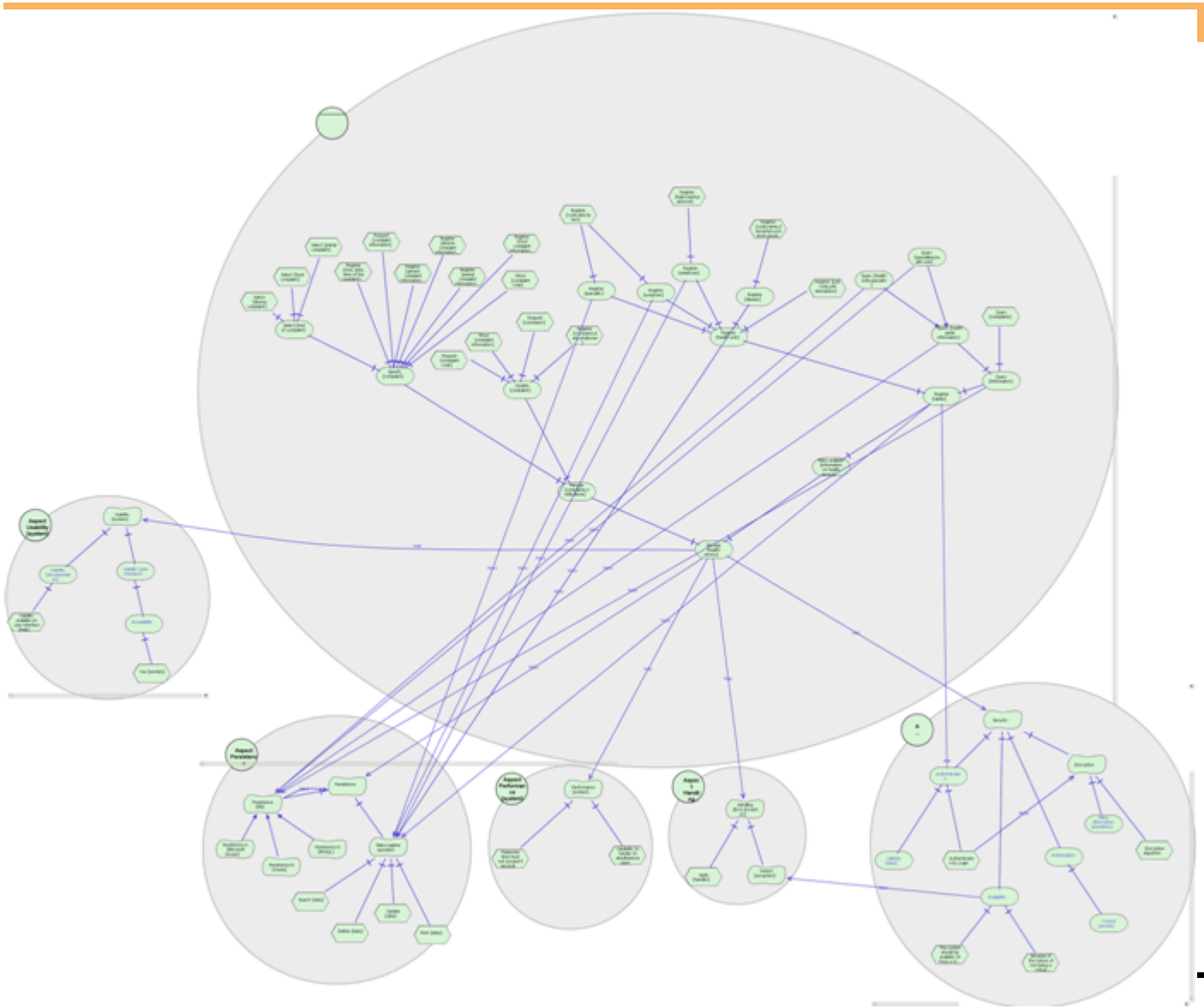
- Identify pointcuts

  *<Citizen>::AutomatedTasks [*]*

5. Separating into aspects



6. Weaved model



LANCASTER UNIVERSITY

The Open University

AOSD-europe
eu network of excellence

Findings about requirement aspects in HealthWatcher

- Functional requirements can be aspects as well
- Aspects can also crosscut other aspects

```
<AspectP>::Persistence {
  Persistence [DB] <= ++ Register [*] { MSAccess MySQL PostgreSQL }
}

<AspectS>::Security {
  Authentication [Login] <= ++ Persistence [DB]
  …
  Encryption <=  & Authentication [Login]
}
```

Summary

- Aspect-orientation helps modularise crosscutting concerns in requirements
- The key contribution of AORE is the composition specifications
  - Aid to specify requirements dependencies
  - Analyse dependencies for conflicts
  - To gain insights into architectural drivers

Further Information

Lancaster AORE Toolset is available from:

http://www.aosd-europe.net (via the Atelier workbench)

OU AORE Toolset is available from:

http://computing-research.open.ac.uk/sead/aore