CiML: A Consolidated Intentional Modeling Language
©Graphical Modelling Language Development (GMLD @ ECMFA), 2013

Joshua C. Nwokeji, Tony Clark, Balbir Barn

Middlesex University, London

<J.Nwokeji, T.N.Clark, B.Barn>@mdx.ac.uk

July 2, 2013
Definitions IM

- Focuses on motivations rather than behaviours of a System.

Uses of IM

- Requirements Modelling, Analysis, Elaboration, and Specification.
## Intentional Modelling Language (IML)

### Definitions IM
- Focuses on motivations rather than behaviours of a System.
- Explains the *Why* instead of *What* of a System.

### Uses of IM
- Requirements Modelling, Analysis, Elaboration, and Specification.
- Checking the completeness of a Requirements Specification.
Intentional Modelling Language (IML)

Definitions IM
- Focuses on motivations rather than behaviours of a System.
- Explains the Why instead of What of a System.
- Uses intentional concepts such as Goals, and Actors for Requirements Modeling.

Uses of IM
- Requirements Modelling, Analysis, Elaboration, and Specification.
- Checking the completeness of a Requirements Specification
- Provides alternatives to choose from, during Systems design
Intentional Modelling Language (IML)

Definitions IM

- Focuses on motivations rather than behaviours of a System.
- Explains the Why instead of What of a System.
- Uses intentional concepts such as Goals, and Actors for Requirements Modeling.
- Provides traceability for business changes.

Uses of IM

- Requirements Modelling, Analysis, Elaboration, and Specification.
- Checking the completeness of a Requirements Specification
- Provides alternatives to choose from, during Systems design
(a) KAOS

(b) i*
Limitations with Current Approach

- Intentional Elements are fragmented across IML.

Table 1: Comparison of IM Key Concepts with i* and KAOS

<table>
<thead>
<tr>
<th>IM Key Concepts</th>
<th>IM Approaches</th>
<th>i* [7]</th>
<th>KAOS [14]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td></td>
<td>√</td>
<td>?</td>
</tr>
<tr>
<td>Goal</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>HardGoal</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>SoftGoal</td>
<td></td>
<td>√</td>
<td>?</td>
</tr>
<tr>
<td>Task</td>
<td></td>
<td>√</td>
<td>ab</td>
</tr>
<tr>
<td>Constraint</td>
<td></td>
<td>ab</td>
<td>?</td>
</tr>
<tr>
<td>Agent</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Strategic Dependency</td>
<td></td>
<td>√</td>
<td>ab</td>
</tr>
<tr>
<td>Conflict</td>
<td></td>
<td>ab</td>
<td>√</td>
</tr>
<tr>
<td>Obstacle</td>
<td></td>
<td>ab</td>
<td>√</td>
</tr>
</tbody>
</table>

Legend:
- √-included in the Language
- ab-Absent in the Language
- ?-Not explicitly defined in the language
Limitations with Current Approach

- Intentional Elements are fragmented across IML.
- Lack of support for detailed requirement analysis.

Table 1: Comparison of IM Key Concepts with i* and KAOS

<table>
<thead>
<tr>
<th>IM Key Concepts</th>
<th>IM Approaches</th>
<th>i* [7]</th>
<th>KAOS [14]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor</td>
<td></td>
<td>✓</td>
<td>?</td>
</tr>
<tr>
<td>Goal</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HardGoal</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SoftGoal</td>
<td></td>
<td>✓</td>
<td>?</td>
</tr>
<tr>
<td>Task</td>
<td></td>
<td>✓</td>
<td>ab</td>
</tr>
<tr>
<td>Constraint</td>
<td></td>
<td>ab</td>
<td>?</td>
</tr>
<tr>
<td>Agent</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Strategic Dependency</td>
<td></td>
<td>✓</td>
<td>ab</td>
</tr>
<tr>
<td>Conflict</td>
<td></td>
<td>ab</td>
<td>✓</td>
</tr>
<tr>
<td>Obstacle</td>
<td></td>
<td>ab</td>
<td>✓</td>
</tr>
</tbody>
</table>

Legend:
✓ - included in the language
ab - Absent in the language
?-Not explicitly defined in the language
Proposed Solution/Contribution

- CIML

A richer but less cumbersome language
Supports detailed requirement analysis
Provide a means to check completeness with respect to Actors, Goals, and Obstacles.
CIML

A richer but less cumbersome language
Proposed Solution/Contribution

- CIML
- A richer but less cumbersome language
- Supports detailed requirement analysis
Proposed Solution/Contribution

- CIML
- A richer but less cumbersome language
- Supports detailed requirement analysis
- Provide a means to check completeness with respect to Actors, Goals, and Obstacles.
Tooting Process

(a) Tooling
(b) ciml.emf + Java Annotation

Figure: Tooling
Example of a Model Constructed with the CiML Tool
### Table 2: Mapping of key CIML concepts with i*, and KAOS

<table>
<thead>
<tr>
<th>CIML Element/Symbol</th>
<th>Mapped Elements in i*</th>
<th>Mapped Elements in KAOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor - a rectangle without an icon</td>
<td>Actor, Role, and Position</td>
<td>Environmental Agent</td>
</tr>
<tr>
<td>Agent - a rectangle with dashed border and Icon</td>
<td>Agent</td>
<td>Software agent</td>
</tr>
<tr>
<td>Goal - a rounded rectangle with no icon and thick border</td>
<td>Goal</td>
<td>Goal</td>
</tr>
<tr>
<td>Resource - a rectangle with no icon and dotted border</td>
<td>Resource</td>
<td>Entity</td>
</tr>
<tr>
<td>HardGoal - A dashed rounded rectangle without icon</td>
<td>HardGoal</td>
<td>Requirement</td>
</tr>
<tr>
<td>NFGoal - A dotted rounded rectangle without icon</td>
<td>Softgoal</td>
<td>Expectation</td>
</tr>
<tr>
<td>Task - An ellipse without an icon</td>
<td>Task</td>
<td>Events</td>
</tr>
<tr>
<td>Obstacle - Ellipse with an Icon</td>
<td>none</td>
<td>Obstacle</td>
</tr>
<tr>
<td>Conflict - same symbol as Goal</td>
<td>none</td>
<td>Conflict</td>
</tr>
<tr>
<td>Constraint - Ellipse with icon, thick and dotted border</td>
<td>none</td>
<td>Pre, and post conditions</td>
</tr>
<tr>
<td>Activity - ellipse with icon and thick border</td>
<td>none</td>
<td>Operation</td>
</tr>
<tr>
<td>RefinementLink: And-closed arrow with solid ellipse, Or-closed arrow with solid rectangle</td>
<td>DecompositionLink, isa, is-part-of</td>
<td>RefinementLink</td>
</tr>
<tr>
<td>ObstructionLink - square</td>
<td>Hurt</td>
<td>ObstructionLink</td>
</tr>
<tr>
<td>DependencyLink - filled rhomb</td>
<td>DependencyLink</td>
<td>none</td>
</tr>
<tr>
<td>AssignmentLink - arrow</td>
<td>none</td>
<td>Operationalization Link, Responsibility Link, and Assignment Link</td>
</tr>
</tbody>
</table>
Graphical Notation for CIML
Next Steps

Model Driven Enterprise Architecture Alignment. Integrating Intentional Models with Systems Model using Model Driven Engineering (MDE) Techniques
Appendix

G1 Provide excellent POS Services.

SG1 Successful Transaction.
SG2 Secure Transaction.
SG3 Happy Customer.

HG1 Card Readable.
HG2 Daily Transaction Uploaded.
HG3 Smart Card Information Encrypted.
HG4 Daily Transaction Uploaded with Secure Option.
HG5 Process Customer’s request on time.
HG6 Replace Smart Card.

Ag1 POS Terminal.
At1 Vendor.
At2 Repair Engineer.
At3 Customer

O1 Smart card Unreadable.
O2 Faulty POS Terminal.
O3 Faulty Smart Card.

C1 Limit Daily Transaction to 500 pounds.

R1 Smart Card
The End