Knowledge Models to Reduce the Gap between Heterogeneous Models

Application to aircraft systems engineering

ICECCS 2010 UML & AADL Workshop

David SIMON ZAYAS (EADS IW)
Dr. Anne MONCEAUX (EADS IW)
Dr. Yamine AIT AMEUR (LISI-ENSMA)
Context of the problem

Heterogeneous multi-modelling

- Two models describing parts of the aircraft Information System

- CIS: Cockpit Information System model
  - CIS sends messages to "external" systems
  - ClosedWorld System
  - SysML modelling language
  - Object-oriented

- SIS: Shared Information System
  - SIS treats messages from CIS
  - OpenWorld System
  - CORE modelling language
  - Functional-oriented

A message sent from ClosedWorld to OpenWorld must be encrypted
Context of the problem

Implicit knowledge

• To relate the two models, engineers need to agree on the following concepts:
  • ClosedWorld
  • OpenWorld
  • Message
  • Encrypted protocol

• If this is done we can exploit relations: e.g. to check the requirement fulfillment
Current solutions in Systems Engineering

Inter-model Relationships Creation

Point-to-point mapping

M1

M2

MetaModel-based

M1'

M2'

MM

Transformation & constraints

Model evolution impact on relations and constraints

No explicit additional knowledge
Goals

- Manage inter-models heterogeneity in a system engineering setting

- Our proposal
  - To describe, model and verify
    - some inter-model constraints and relationships
    - between pre-existing heterogeneous models (analysis)
    - used in a System Engineering process
  - To make explicit, formalize and exploit additional knowledge
    - usually not expressed by the engineers
    - to express these constraints and relationships
Methodology proposal

Knowledge-based inter-model relationships management

- M1
- M1'
- M1''
- M2
- M2'
- M2''

Relation model
- K model
- Constraints
- Reference
- Constraint
- Annotation
- Traceability
- Export/Transformation

Common modelling universe
Knowledge-based inter-model relationships management

A message sent from ClosedWorld to OpenWorld must be encrypted.

CIS Concept ↔ SIS Concept

Reference

Constraint

Annotated CIS

Annotated SIS

Export/Transformation

Reference

Annotation

Export/Transformation

Encryption protocol

Message

Exported CIS

Exported SIS

CIS

SIS

Export/Transformation

ClosedWorld

OpenWorld
Step by Step

Step 1 (Export to a common modelling formal language)

CORE MM

SIS model as instance of CORE MM

Express implementation
Step 2 (Formal side knowledge model)
Step by Step

Step 3 (Knowledge based model annotation)

Annotation model

Expression implementation

Constraints

Common modelling

Annotation instances
Step by Step

Step 4 (Inter-models relations)

Relation model

---This entity represents a relation between 2 or more elements of different models
ENTITY RELATION
ABSTRACT SUPERTYPE
SUBTYPE OF (CONCEPT);
NAME : STRING;
URI_CONCEPT_ORIGIN: URI_TYPE;
URL_CONCEPT_DESTINATION: SET[1:1] OF URI_TYPE;
ATTRIBUTES: SET[0:1] OF ATTRIBUTE_RELATION;
MULTIPlicity_ORIGIN: SET[0:2] OF MULTIPLICITY;
MULTIPLICITY_DESTINATION: SET[0:2] OF MULTIPLICITY;
CONSTRAINTS: SET[0:?] OF INTER_MODEL_CONSTRAINT;
END_ENTITY;

Express implementation

#33=EQUIVALENCE(#65, 'Interface Equivalence', #39, (#54), (),
(#66), (#67), (#253)); #65=URI_TYPE('EQUIV001'); #66=MULTIPLICITY(1, 1); #67=MULTIPLICITY(1, 1);
Step 5 (Express constraints over the relations)

Constraints model based on PLIB expressions model

"It exists at least one Link which transfers a maintenance message"

Express implementation

---

This entity represents the EXISTS FOL assertion
ENTITY EXISTS_EXPRESSION
SUBTYPE OF (FIRST_LOGICAL_BINARY_EXPRESSION);
DERIVE
SELF BOOLEAN_VALUE_EXPRESSION BOOLEAN_VALUE EXIST_PASS repay_function;
END_ENTITY;
Step by Step

Step 5 (Check of constraint)
For each LINK transferring a maintenance message from a Cockpit INTERFACE (CIS SysML model) to a Shared System INTERFACE (SIS CORE model), the used communication protocol must be secure.

\[
\forall l: \text{Link} \exists \text{Operation UML} \exists \text{Parameter UML} \exists \text{Adt: Annotated Data Type} \exists \text{Adt: Annotated Data Type} \exists \text{Communication Protocol} \equiv \text{cp: Communication Protocol} \land \text{cp: Communication Protocol} \lor \text{cp: Communication Protocol} \equiv \text{cp: Communication Protocol} \lor \text{cp: Communication Protocol} \equiv \text{cp: Communication Protocol}
\]

- For intranet:
  - CIS \rightarrow SIS
  - EX25

- For extranet:
  - CIS \rightarrow SIS
  - X25

Diagram:

![Diagram showing the communication protocol constraint for intranet and extranet connections between CIS and SIS.]
Operational validation with Express modeling language

• Set of entities grouped in a schema
  • Entity = set of typed attributes
  • Derived attributes
  • Inverse attributes
• Types
  • Base Types: INTEGER, REAL, BOOLEAN, STRING
  • Complex Types: Abstract Types LIST, SET, BAG, ARRAY with corresponding operations.
  • Entities used as attribute types.

```express
ENTITY B;
  att_1: REAL;
  att_2: LIST [0:?] OF STRING;
  att_3: A;
END_ENTITY;

ENTITY B2;
  att_1: REAL;
  att_2: LIST[0:?] OF STRING;
  att_3: A;
DERIVE
  att_4: BOOLEAN:=(SELF.att_3\A.att_A =
                  SIZEOF(SELF.att_2));
END_ENTITY;

ENTITY A;
  att_A: INTEGER;
INVERSE
  att_I: B FOR att_3;
END_ENTITY;

SCHEMA Foo1;
ENTITY A;
  att_A: INTEGER;
INVERSE
  att_I: B FOR att_3;
END_ENTITY;

#1=A(3);
#2=B(4.0, ('hello','bye'), #1);
```
Operational Validation

Express
- Schemas and entities can be constrained
- Constraints = First order logical expression.
  - Local Constraint: applied to an entity (WHERE Clause)
  - Global Constraint: applied to entities in a schema (RULE Clause).

Local Constraints

ENTITY A;
  att_A: INTEGER;
WHERE
  WR1 : (SELF.att_A >= 1) AND (SELF.att_A <= 10);
  WR2 : SELF.att_A = f(SELF);
END_ENTITY;

Global Constraint

RULE Card FOR A;
WHERE
  SIZEOF (QUERY (inst <* A | (inst.att_A=1))) = 2;
END_RULE;
Conclusion

• Approach integrating Systems Engineering models
• Specificity of the approach
  • Implicit knowledge is made explicit
  • Side models for relations and constraints
• Operational approach
  • Use of EXPRESS
Perspectives

- Illustrate other cases and kind of relations
- Demonstrate other uses of formal semantic relation
  - To develop services exploiting formal relation
    - Discover relation using K annotation
    - Search into models
    - Check models consistency
- Study the impact of source models evolution
- Scalability
  - More than two models?
  - More than two languages?
Thank you
© EADS France. All rights reserved. Confidential and proprietary document.

This document and all information contained herein is the sole property of EADS France. No intellectual property rights are granted by the delivery of this document or the disclosure of its content. This document shall not be reproduced or disclosed to a third party without the express written consent of EADS France. This document and its content shall not be used for any purpose other than that for which it is supplied.

The statements made herein do not constitute an offer. They are based on the mentioned assumptions and are expressed in good faith. Where the supporting grounds for these statements are not shown, EADS France will be pleased to explain the basis thereof.
Step by Step

Step 1 (Export to a common modeling formal language)

- Need of meta-model of MMi for the relevant concepts involved in heterogeneity reduction
- Models Mi become instances of MMi
Step 2 (Formal side knowledge model)

- Domain knowledge is explicitated
- Each concept is uniquely identified (URI)
Step 3 (Knowledge based model annotation)

- Define semantic links between Models elements (Classes And/Or Instances) and Knowledge concepts

- Each relevant model element is annotated with an URI from K Model (1-1, 1-n, n-1, n-n)
Step by Step

Step 4 (Inter-models relations)

- Need for relationship model
- Structural relations
- Constraint on these relations

Exploit K Model

---

![Diagram showing inter-models relations and annotations](image-url)
Step 5 (express constraints over the relations)

- This is goal oriented
- To guarantee inter-model consistency
- To check inter-model properties