An Introduction to
Constrained Objects (COBs) and
X-Analysis Integration (XAI)
Part 1

Guest Lectures for ME 6754 - Spring 2000

Russell S. Peak
Georgia Tech
Engineering Information Systems Lab
eislab.gatech.edu
Outline

- Constrained Objects (COBs)
- Overview of X-Analysis Integration (XAI)
- Example Applications
  - Electronic Packaging Thermomechanical Analysis
  - Aerospace Structural Analysis
COB Structure: Graphical Forms

Spring Primitive

Figure

\[ F(x_1, x_2) = k \Delta L \]

Relations

\[ r_1 : L = x_2 - x_1 \]
\[ r_2 : \Delta L = L - L_0 \]
\[ r_3 : F = k \Delta L \]

Basic Constraint Schematic Notation

Template Structure (Schema)

Constraint Schematic

Subsystem View

(for reuse by other COBs)
COB Structure: Lexical Form

Spring Primitive

Constraint Schematic

Lexical COB Schema Template

```cob
COB spring SUBTYPE_OF abb;
    undeformed_length, L<sub>0</sub> : REAL;
    spring_constant, k : REAL;
    start, x<sub>1</sub> : REAL;
    end, x<sub>2</sub> : REAL;
    length, L : REAL;
    total_elongation, ΔL : REAL;
    force, F : REAL;

RELATIONS
    r1 : "<length> == <end> - <start>";
    r2 : "<total_elongation> == <length> - <undeformed_length>";
    r3 : "<force> == <spring_constant> * <total_elongation>";
END_COB;
```
Example COB Instance
Spring Primitive

Constraint Schematic Instance Views
example 1, state 1

Lexical COB Instances
input:

\begin{verbatim}
INSTANCE_OF spring;
  undeformed_length : 20.0;
  spring_constant : 5.0;
  start : ?;
  end : ?;
  length : ?;
  total_elongation : ?;
  force : 10.0;
END_INSTANCE;
\end{verbatim}

result (reconciled):

\begin{verbatim}
INSTANCE_OF spring;
  undeformed_length : 20.0;
  spring_constant : 5.0;
  start : ?;
  end : ?;
  length : 22.0;
  total_elongation : 2.0;
  force : 10.0;
END_INSTANCE;
\end{verbatim}

Basic Constraint Schematic Notation

- Input $a = 100$ lbs
- Result $b = 30\text{e}6$ psi (output or intermediate variable)
- Result $c = 200$ lbs (result of primary interest)
- Equality relation is suspended
- Relation $rI$ is suspended

Input:

- $5 \text{ N/mm}$
- $20 \text{ mm}$
- $10 \text{ N}$
- $2 \text{ mm}$

Output or intermediate variable:

- $30\text{e}6 \text{ psi}$
- $200 \text{ lbs}$
Multi-Directional I/O (non-causal)
Spring Primitive

Constraint Schematic Instance View

Design check
example 1, state 1

Design synthesis
example 1, state 5

Lexical COB Instance
(state 5)

input:

```
INSTANCE_OF spring;
    undeformed_length : 20.0;
    spring_constant : ?;
    start : 10.0;
    end : ?;
    length : 22.0;
    total_elongation : ?;
    force : 40.0;
END_INSTANCE;
```

result:

```
INSTANCE_OF spring;
    undeformed_length : 20.0;
    spring_constant : 20.0;
    start : 10.0;
    end : 32.0;
    length : 22.0;
    total_elongation : 2.0;
    force : 40.0;
END_INSTANCE;
```
Traditional Mathematical Representation

Two Spring System

System Figure

Free Body Diagrams

Variables and Relations

Kinematic Relations

Constitutive Relations

Boundary Conditions
Constraint Graph
Two Spring System

\[ r_{11} : L_1 = x_{12} - x_{11} \]
\[ r_{12} : \Delta L_1 = L_1 - L_{10} \]
\[ r_{13} : F_1 = k_1 \Delta L_1 \]
\[ r_{21} : L_2 = x_{22} - x_{21} \]
\[ r_{22} : \Delta L_2 = L_2 - L_{20} \]
\[ r_{23} : F_2 = k_2 \Delta L_2 \]

\[ bc_1 : x_{11} = 0 \]
\[ bc_2 : x_{12} = x_{21} \]
\[ bc_3 : F_1 = F_2 \]
\[ bc_4 : F_2 = P \]
\[ bc_5 : u_1 = \Delta L_1 \]
\[ bc_6 : u_2 = \Delta L_2 + u_1 \]
Analysis Primitives

with
Encapsulated Relations

\( r_{11} : L_1 = x_{12} - x_{11} \)
\( r_{12} : \Delta L_1 = L_1 - L_{10} \)
\( r_{13} : F_1 = k_1 \Delta L_1 \)

\( r_{21} : L_2 = x_{22} - x_{21} \)
\( r_{22} : \Delta L_2 = L_2 - L_{20} \)
\( r_{23} : F_2 = k_2 \Delta L_2 \)

System-Level Relations
(Boundary Conditions)

\( bc_1 : x_{11} = 0 \)
\( bc_2 : x_{12} = x_{21} \)
\( bc_3 : F_1 = F_2 \)
\( bc_4 : F_2 = P \)
\( bc_5 : u_1 = \Delta L_1 \)
\( bc_6 : u_2 = \Delta L_2 + u_1 \)
COBs as Building Blocks

Two Spring System

Constraint Schematic

Lexical COB Schema Template

```
COB spring_system SUBTYPE_OF analysis_system;
  spring1 : spring;
  spring2 : spring;
  deformation1, u<sub>1</sub> : REAL;
  deformation2, u<sub>2</sub> : REAL;
  load, P : REAL;

RELATIONS
  bc1 : "<spring1.start> == 0.0";
  bc2 : "<spring1.end> == <spring2.start>";
  bc3 : "<spring1.force> == <spring2.force>";
  bc4 : "<spring2.force> == <load>";
  bc5 : "<deformation1> == <spring1.total_elongation>";
  bc6 : "<deformation2> == <spring2.total_elongation>
    + <deformation1>";

END_COB;
```
input:
INSTANCE_OF spring_system;
spring1.undeformed_length : 8.0;
spring1.spring_constant : 5.5;
spring2.undeformed_length : 8.0;
spring2.spring_constant : 6.0;
load : 10.0;
END_INSTANCE;

result:
INSTANCE_OF spring_system;
spring1.undeformed_length : 8.0;
spring1.spring_constant : 5.5;
spring1.start : 0.0;
spring1.end0 : 9.81818181818182;
spring1.force : 10.0;
spring1.total_elongation : 1.818181818181818;
spring1.length : 9.81818181818182;
spring2.undeformed_length : 8.0;
spring2.spring_constant : 6.0;
spring2.start : 9.81818181818182;
spring2.force : 10.0;
spring2.total_elongation : 1.666666666666666;
spring2.length : 9.66666666666667;
spring2.end0 : 19.484848484848484;
load : 10.0;
deformation1 : 1.818181818181818;
deformation2 : 3.484848484848484;
END_INSTANCE;
Spring Examples Implemented in XaiTools X-Analysis Integration Toolkit
COB Modeling Views

Constraint Schematic

Object Relationship Diagram
Express-G

COB Schema Language
STEP Express

Subsystem Views
I/O Tables
Extended Constraint Graphs

HTML

Constraint Schematic-I

Extended Constraint Graphs-I

COB Instance Language
STEP Part 21

HTML
AA is an entity (class)
Instance of A are objects

A is a simple type
( BOOLEAN, LOGICAL, BINARY,
NUMBER, INTEGER, REAL, STRING)

A has two attribute, a1 and a2, that are both type B

A has an attribute, a1, that is a Set of 1 or ore entities of type B

A is a supertype of B and C. (B and C are subtype of A)

Unofficial extensions:
A has two levels, a1 and a2. a1 is type B. a2 is type C.
COB Object Model View (EXPRESS-G)
Spring Schema

- Real: undeformed_length
- Real: force
- Real: total_ elongation
- Real: length
- Real: end0
- Real: start
- Real: spring_constant
- Real: spring_system
- Real: load
- Real: deformation1
- Real: deformation2

Diagram:
- Springs k1 and k2
- Forces F
- Deformations ΔL
- Lengths L, L0
- Start and end points x1, x2

© GIT
Engineering Information Systems Lab • eislab.gatech.edu
Declarative Knowledge / Derivable Behavior

Two Spring System

\[ \Delta = k_1 \Delta L_1 \]
\[ r_{11} : L_1 = x_{12} - x_{11} \]
\[ r_{12} : \Delta L_1 = L_1 - L_{10} \]
\[ r_{13} : F_1 = k_1 \Delta L_1 \]
\[ r_{21} : L_2 = x_{22} - x_{21} \]
\[ r_{22} : \Delta L_2 = L_2 - L_{20} \]
\[ r_{23} : F_2 = k_2 \Delta L_2 \]

\[ u_{L1} = 0 \]
\[ u_{L2} = \Delta L_1 \]
\[ u_{bc1} = \frac{P}{k_1} \]
\[ u_{bc2} = \Delta L_2 + u_1 \]
\[ u_{bc3} = \frac{P}{k_2} \]
\[ u_{bc4} = \frac{k_1 + k_2}{k_1 k_2} \]

No need to include explicitly (redundant)
Achieving Effective System Properties via Semantically Rich COBs

Derivable System Level Properties

\[ dr_1 : k_{\text{effective}} = \frac{1}{k_1} + \frac{1}{k_2} \]

\[ dr_2 : \Delta L_{\text{effective}} = \Delta L_1 + \Delta L_2 \]

No need to derive
Minimal extra work
Semantically richer
Constrained Object Language (COBs)

- **Capabilities & features:**
  - Various forms: computable lexical form, graphical form, etc.
  - Sub/supertypes, basic aggregates, multi-fidelity objects
  - Multi-directionality (I/O change)
  - Wrapping external programs as white box relations

- **Analysis module/template applications:**
  - Product model idealizations
  - Explicit associativity relations with design models & other analyses
  - White box reuse of existing tools (e.g., FEA, in-house codes)
  - Reusable, adaptable analysis building blocks
  - Synthesis (sizing) and verification (analysis)
Constrained Object Language (cont.)

- **Overall characteristics**
  - Declarative knowledge representation
  - Combining object & constraint graph techniques
  - COBs = (STEP EXPRESS subset) + (constraint concepts & views)
  - Advantages over traditional analysis representations:
    » Greater solution control
    » Richer semantics (e.g., equations wrapped in engineering context)
    » Capture of reusable knowledge

- **Further needs …**
  - Higher order constraints
  - Hybrid declarative/procedural approaches
  - Etc.