Institute for System Programming
Russian Academy of Sciences

Object-Oriented & Constraint-Based Programming: New Applications for Software Engineering and Product Data Management

Vitaly Semenov

17 November 2011, Moscow

AGENDA

- OO & CB programming paradigms
  - New declarative approach based on popular modelling languages (EXPRESS, UML/OCL, OQL, OWL)
  - Some examples
- STEP (ISO10303) family standards
- Applications for software engineering and data management
  - Checking model-driven data
  - Model verification
  - Test generation
  - Correction of errors
  - Semantic diff/merge of divergent data
- Conclusions

Constraint-Based Programming

- Application areas

  Computer graphics (declarative scenario models and user interfaces)
  Decision making systems (neural propagation networks)
  Economics models (undetermined computations)
  CAD/CAM (parametric systems)
  Information systems (active DBMS, consistency management)
  Data retrieval (descriptive logic)
  Software verification and testing (temporal logic)
  Collaborative environments (poly-syllogistic deduction)

- Problem statements

  Constraint Satisfaction Problem (CSP)
  Constraint Logic Programming (CLP(B), CLP (Q), CLP (R), CLP(FD))
  Concurrent Constraint Programming (CCP)

Constraint-Based Programming

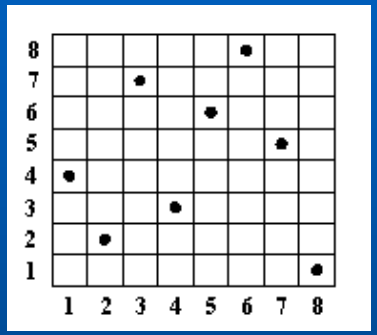
- Some technologies
  - Sketchpad, ALICE, ThingLab, CHIP
  - Lisp, Prolog, ConstraintLisp, Prolog III
  - AKL, Oz, CIAO, TAO
  - Constraint DataBase (CDB)
  - Constraint Based Testing (CBT)
- Underlying methods and approaches
  - Local propagation (Red, Orange, Yellow, Green, DeltaBlue, SkyBlue, UltraViolet, Purple, DeepPurple, Indigo)
  - Global search (ILOG Solver, CHARME)
  - Simplex for linear algebraic systems (2LP)
  - Interval arithmetic (HELIOS, ILOG Solver)
  - Numeric methods for nonlinear systems (NEWTON)
  - Critical path method (ILOG Schedule, CHIP)
  - Groebner basis for polynomial systems (H4-Revise)

Mixed OO&CB Programming

Eight queens puzzle

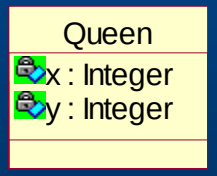
The puzzle is a problem of placing eight chess queens on an 8×8 chessboard so that no two queens attack each other. Thus, a solution requires that no two queens share the same row, column, or diagonal.

The puzzle has 92 **distinct** solutions. If solutions that differ only by symmetry operations (rotations and reflections) of the board are counted as one, the puzzle has 12 **unique** (or fundamental) solutions.



Specification at UML/OCL

```
context Queen
inv: self.AllInstances()->size() = 8
inv: self.x >= 1 and self.x <= 8 and self.y >= 1 and self.y <= 8
inv: self.AllInstances()->forAll(q1, q2 | q1 <> q2 implies  (q1.x <> q2.x and q1.y <> q2.y))
inv: self.allInstances()->forAll(q1, q2 | q1 <> q2 implies  ((q1.x – q1.y) <> (q2.x – q2.y) and (q1.x + q1.y) <> (q2.x + q2.y)))
```



Eight Queens Puzzle

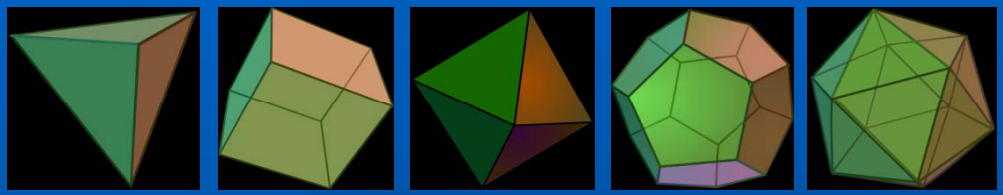
Prolog

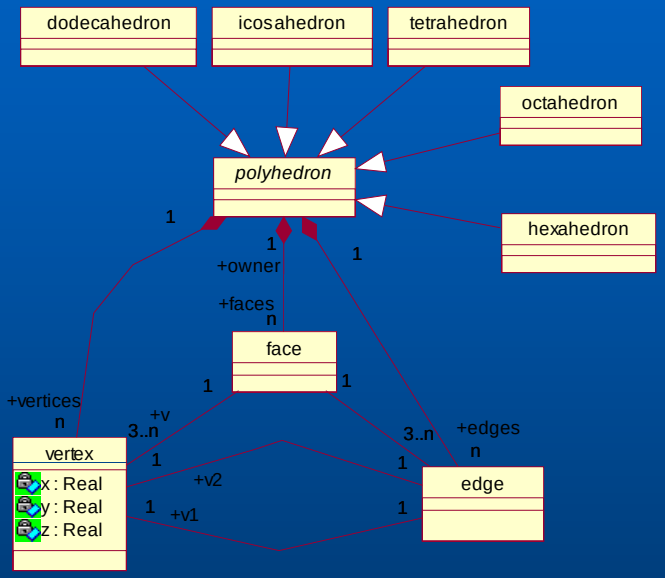
```
?-  шаблон( S), решение( S).
решение( [ ] ).
решение( [x/y | Остальные ] ) :-
                % Первый ферзь на поле x/y,
                % остальные ферзи на полях из списка Остальные
    решение( Остальные ),
    принадлежит( y, [1, 2, 3, 4, 5, 6, 7, 8] ),
    не_бьет( x/y | Остальные ).  % Первый ферзь не бьет остальных
не_бьет( x/y, [ ]).          % Некого бить
не_бьет( x/y, [x1/y1 | Остальные] ) :-
        y =\= y1,           % Разные y-координаты
        y1 - x1 =\= y - x,    % Разные диагонали
        y1 + x1 =\= y + x,
        не_бьет( x/y, Остальные).
принадлежит( x, [x | L] ).
принадлежит( x, [y | L] ) :-
        принадлежит( x, L).
% Шаблон решения
шаблон( [1/y1, 2/y2, 3/y3, 4/y4, 5/y5, 6/y6, 7/y7, 8/y8] ).
```

ConstraintLisp

```
;;; определение класса Queen
(defclass Queen ( )
   (( x :initarg :x :accessor x ) ;;; координата по горизонтали
   ( y :initarg :y :accessor y))) ;;; координата по вертикали
;;; основная процедура поиска решения
(define Queens ( )
(let ((Chessboard ( make-queens)) ) ;;; создаем массив из восьми ферзей
   (cldotimes ( I 7 ) ;;; начальное значение переменной I 0
      (cldotimes ( J (- 7 I )) ;;; начальное значение переменной J 0
         (constraints ( x (aref Chessboard I ))
            ( x (aref Chessboard (+ J I 1 ))) (1+ I) (+ J I 2 ))))
( generateObjs Chessboard ) ;;; генерация решения
( printArrayObjs Chessboard ))) ;;; вывод результатов
;;; ограничения отсутствия вертикальных и диагональных атак
(define constraints (x1 x2 y1 y2)
   (constr (/= x1 x2)) ;;; по вертикали
   (constr (/= (+ x1 y1 ) (+ x2 y2 )))  ;;; по главной диагонали
   (constr (/= (- x1 y1 ) (- x2 y2 )))) ;;; по второй диагонали
;;; создаем массив ферзей
(defun make-queens ( )
   (let ( (queen-array (make-array 8 )))
      (dotimes (I 8 queen-array )
         (setf (aref queen-array I )
            (make-instance 'queen :x (make-cvar-in '((1 ,8 ))) :y (1+ I) )))))
```

Platonic Solids





- Specification at UML/OCL

context polyhedron

inv: self.vertices->size() - self.edges->size() + self.faces->size() = 2
inv: self.faces->forAll(f1, f2 : face | f1.edges->size() = f2.edges->size())
inv: self.vertices->forAll(v1, v2 | v1.numEdges->size() = v1.numEdges->size())

context face
  let secVertices : Sequence(Vertex) = collect(self.v) in
  let first : Vertex = secVertices(1) in
  let sec : Vertex = secVertices(2) in
  let third : Vertex = secVertices(3) in
  let A,B,C,D…
inv: self.owner.vertices->collect(v1 : vertex | not self.vertices->includes(v1)) ->forAll(v2 : vertex | A * v2.x + B * v2.y + C * v2.z + D > 0)

context edge

inv: (self.v2.x – self.v1.x) * (self.v2.x – self.v1.x) + (self.v2.y – self.v1.y) * (self.v2.y – self.v1.y) + (self.v2.z – self.v1.z) * (self.v2.z – self.v1.z) = 1

Product Data Management

WHAT is a PDM?

- “The management of all data related to a product, regardless of its origin, its content, its meaning or its format, over the whole of the product lifecycle.”

Key challenges:

- Multi-disciplinary data and process modelling
- Multi-platform environments (heterogeneous platforms, information standards, languages, legacy applications and data)
- Multi-modal collaboration (alternative transaction models)

Architecture

Landscape design

Virtual mock-up

Structural design

Stress analysis

HVAC

Electric/Lighting

Facility management

Cost estimation

Construction management

STEP (ISO 10303) Standards

General Organization

Description methods:

- EXPRESS, EXPRESS-X, EXPRESS-G

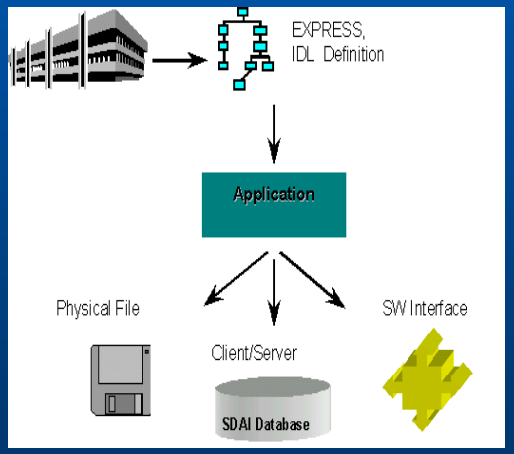
Implementation methods:

- Clear text encoding, XML
- Standard data access interface (SDAI) bindings at C, C++, Java, C#
- IDL, WSDL
- UML/XMI
- SQL, OQL, XML data stores

Conformance testing methodology

- EXPRESS-I

Application parts



STEP Standards

Application domains

- Architecture, engineering, construction
- Airspace, automotive industries
- Ship building
- Electronics, electro-technical equipments
- Geophysics and petroleum production
- Furniture
- Pharmaceutics
- Software engineering

Industry expectations

- Achieving software re-use, portability, cross-platform interoperability
- Reducing time, cost, complexity of software development and integration

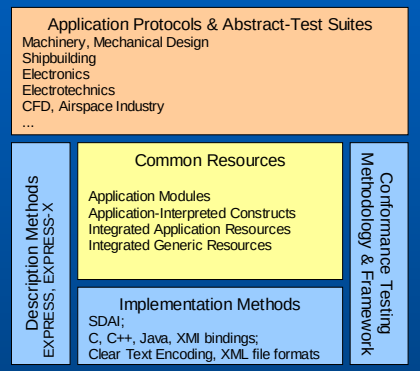
ISO TC184 and OMG

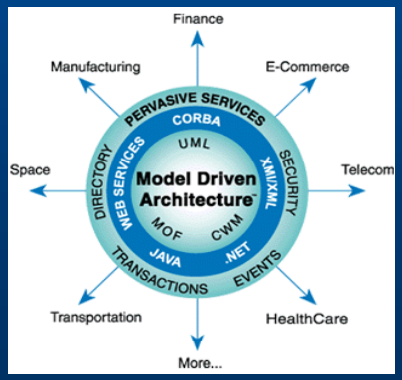
STEP & MDA/MDD harmonization

"Product Data Management (PDM) is an increasingly important technology for engineering and manufacturing enterprises. The information technology that supports Product Data Management requires standards in order to allow interoperability between systems and the sharing of product information between organizations. Two important sources of standards in this area are the Object Management Group (OMG) and the International Organization for Standardization (ISO) STEP community (officially ISO TC184/SC4 Industrial Data)."

"OMG and ISO are organizations that develop and promote standards. The OMG and STEP communities work independently to develop standards for different purposes and that differ in scope, abstraction level, and operational characteristics. The OMG standard defines a standard interface to the services of a PDM system in a distributed, object oriented environment. The STEP standard defines a standard representation of product data including data that is typically managed by a PDM system."

"However, the OMG and STEP communities have worked together to harmonize their respective standards in the area of Product Data Management Standards so that the standards complement each other rather than conflict or overlap."





SemanticSTEP Platform

WHAT doesn't define STEP and WHAT does SemanticSTEP fulfil?

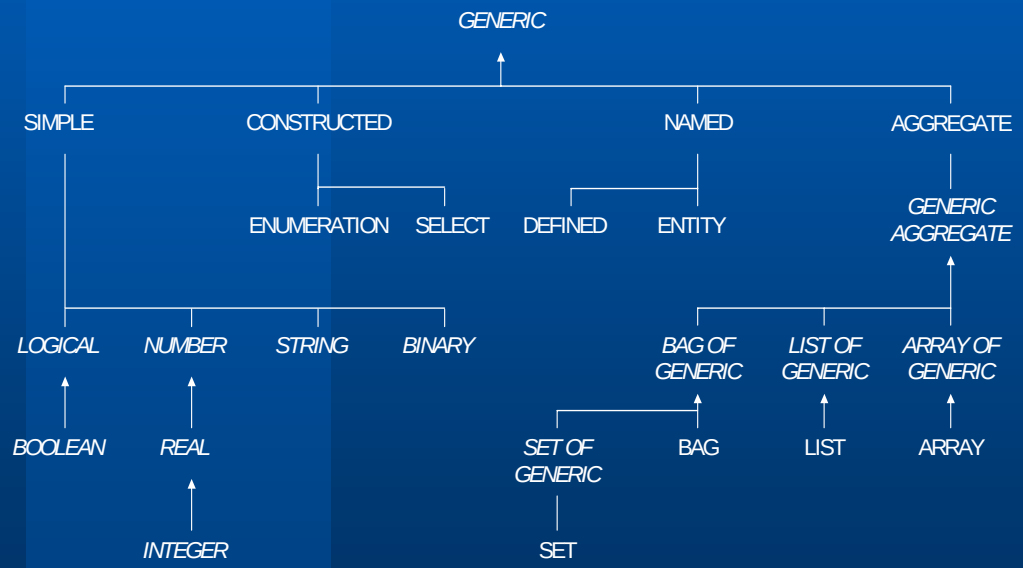
- How to verify data models?
- How to check large-scale data against semantic constraints?
- How to generate test data sets and to certificate interoperable application?
- How to control data consistency and correct errors?
- What are the methods (e.g. long transaction models) to manage product model data?

- What is PDM context for the STEP and its relationships with change, version, configuration, workflow, and document management?
- How to store the product data and to access it using general-purpose databases?
- How to adapt applications to STEP SDAI interfaces?
- How to port legacy STEP-compliant applications into evolving platforms CORBA/IDL, J2EE/Java, XML/SOAP, .NET/C#, WebServices/WSDL?

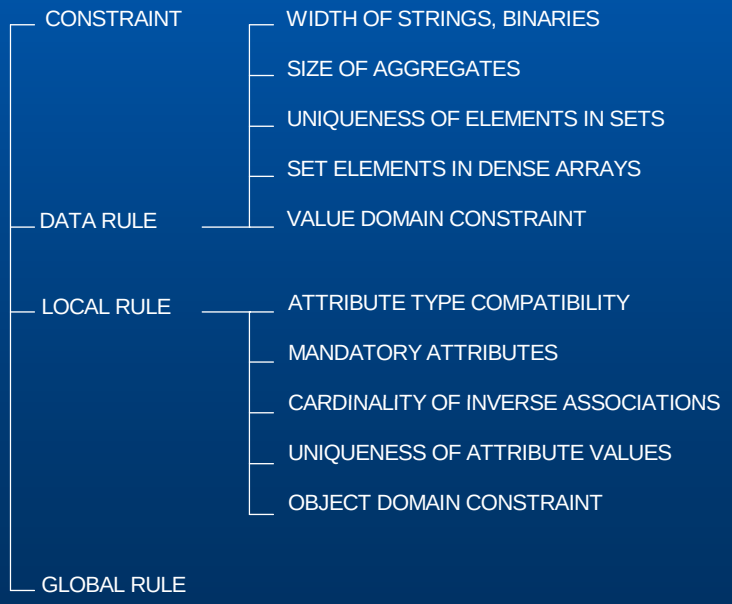
STEP Standards

EXPRESS metamodel

- Data Types

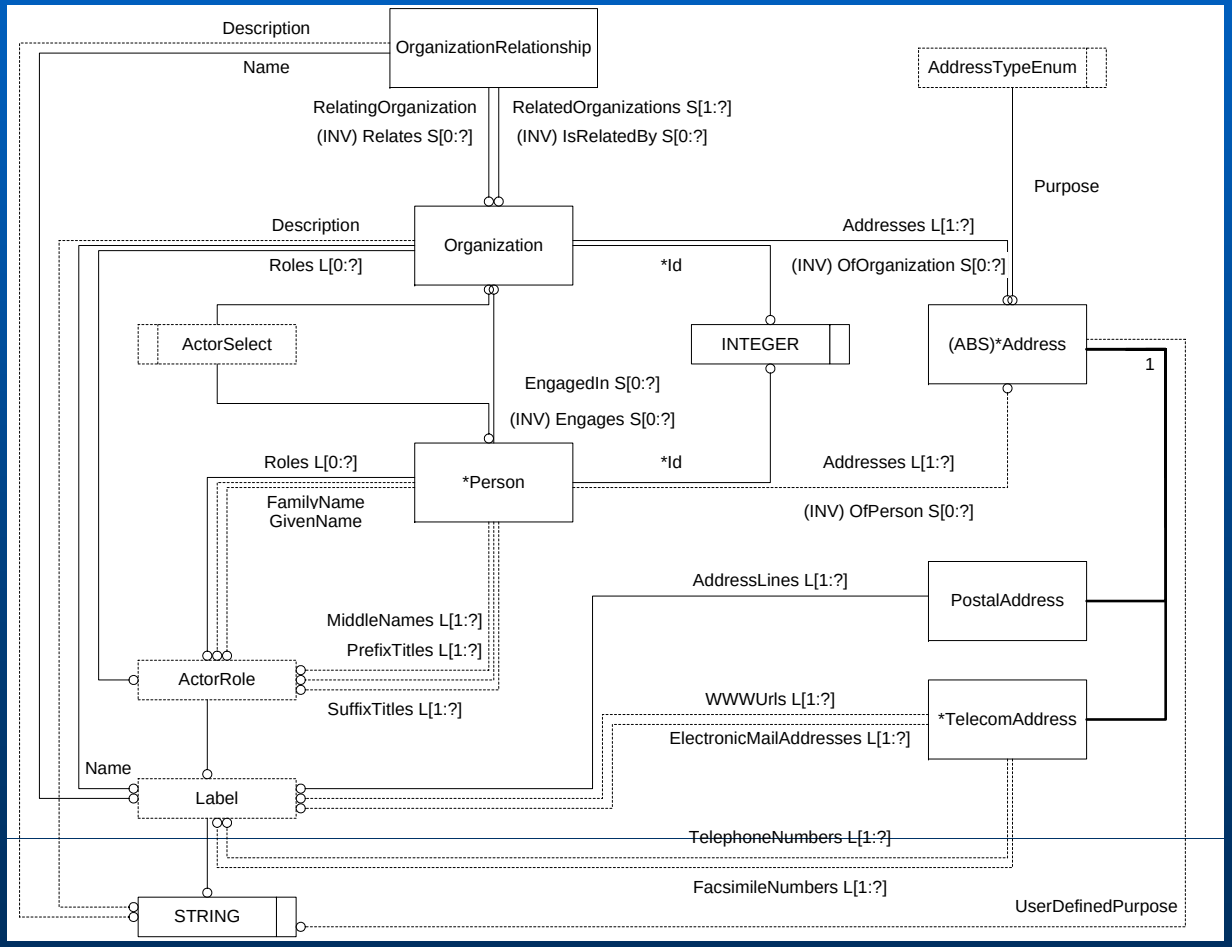


- Constraint Sorts



STEP Standards

EXPRESS-G diagram example



STEP Standards

EXPRESS schema example

```
SCHEMA ActorResource;

 ENTITY Person;
      Id           : INTEGER;
      FamilyName   : STRING(255);
      GivenName    : STRING(255);
      MiddleNames  : OPTIONAL LIST [1:?] OF STRING(255);
      Titles       : OPTIONAL LIST [1:?] OF STRING(255);
      Addresses    : OPTIONAL LIST [1:?] OF UNIQUE Address;
      EngagedIn    : SET OF Organization;
    UNIQUE
      UR1 : Id;
  END_ENTITY;

  TYPE AddressTypeEnum = ENUMERATION OF (OFFICE, HOME, USERDEFINED); END_TYPE;

  ENTITY Address
    ABSTRACT SUPERTYPE OF (ONEOF(PostalAddress, TelecomAddress));
      Purpose            : AddressTypeEnum;
      UserDefinedPurpose : OPTIONAL STRING;
    INVERSE
      OfPerson           : SET OF Person FOR Addresses;
      OfOrganization     : SET OF Organization FOR Addresses;
    WHERE
      WR1 : (Purpose <> AddressTypeEnum.USERDEFINED) OR
            ((Purpose = AddressTypeEnum.USERDEFINED) AND EXISTS(UserDefinedPurpose));
  END_ENTITY;
                                                        …END_SCHEMA;
```

STEP Standards

XML-based representation (ISO10303-28)

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<?IS10744 arch name="iso_10303_28" dtd-system-id="iso_10303_28.dtd" dtd-public-id="ISO 10303-28:2000//DTD"
10303_28_Architectural_DTD//EN form-att="late-bound-element" suppressor-att="late-bound-processing" renamer-
att="late-bound-name" doc-elem-form="iso_10303_28" auto="nArcAuto" ?>
<!DOCTYPE iso_10303_28 SYSTEM "actorresource.dtd">
<iso_10303_28 representation_category="ETEB" version="PDTS">
            <iso_10303_28_header>
                        <document_name>fileName</document_name>
                        <time_stamp>2011-11-11T11:11:11</time_stamp>
                        <author>author</author>
                        <originating_organization>organizationName</originating_organization>
                        <preprocessor_version>engineVersion</preprocessor_version>
                        <originating_system>Windows System</originating_system>
                        <authorization>authorAuthorization</authorization>
                        <documentation>Example</documentation>
            </iso_10303_28_header>
            <express_data id="data1">
                        <Actorresource-schema id="data_sch1">
                                    <Person id="1">
                                                <Person.id> <integer>1</integer> </Person.id>
                                                <Person.familyname> <string>Petrov</string>
</Person.familyname>
                                                <Person.givenname> <string>Petr</string> </Person.givenname>
                                                <Person.middlenames>
                                                            <list-of-string> <string>Petrovitch</string>
</list-of-string>
                                                </Person.middlenames>
```

STEP Standards

Clear text encoding (ISO10303-21)

```
ISO-10303-21;

HEADER;
FILE_DESCRIPTION (('Example'), '2;1');
FILE_NAME ('fileName', '2011-11-11T11:11:11', ('author'), ('organizationName'), 'engineVersion', 'Windows
System', 'authorAuthorization');
FILE_SCHEMA (('ACTORRESOURCE'));
ENDSEC;

DATA;
#1= PERSON(1,'Petrov','Petr',('Petrovitch'),$,(#2,#3),$);
#2= POSTALADDRESS(OFFICE,$,...);
#3= POSTALADDRESS(HOME,$,...);
#4= PERSON(2,'Ivanov‘,'Ivan',('Ivanovitch'),$,(#5,#6,#7),$);
#5= POSTALADDRESS(OFFICE,$,...);
#6= TELECOMADDRESS(OFFICE,$,...);
#7= POSTALADDRESS(HOME,$,...);
ENDSEC;

END-ISO-10303-21;
```

STEP Standards

EXPRESS schema example

```
SCHEMA ActorResource;

 ENTITY Person;
      Id           : INTEGER; (*MANDATORY*)
      FamilyName   : STRING(255);
      GivenName    : STRING(255);
      MiddleNames  : OPTIONAL LIST [1:?] OF STRING(255);
      Titles       : OPTIONAL LIST [1:?] OF STRING(255);
      Addresses    : OPTIONAL LIST [1:?] OF UNIQUE Address;
      EngagedIn    : SET OF Organization;
    UNIQUE
      UR1 : Id;
 END_ENTITY;

 TYPE AddressTypeEnum = ENUMERATION OF (OFFICE, HOME, USERDEFINED); END_TYPE;

 ENTITY Address
   ABSTRACT SUPERTYPE OF (ONEOF(PostalAddress, TelecomAddress));
     Purpose            : AddressTypeEnum; (*MANDATORY*)
     UserDefinedPurpose : OPTIONAL STRING;
   INVERSE
     OfPerson           : SET OF Person FOR Addresses;
     OfOrganization     : SET OF Organization FOR Addresses;
   WHERE
     WR1 : (Purpose <> AddressTypeEnum.USERDEFINED) OR
           ((Purpose = AddressTypeEnum.USERDEFINED) AND EXISTS(UserDefinedPurpose));
 END_ENTITY;
                                                    …END_SCHEMA;
```

SemanticSTEP

Checking model-driven data against constraints

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (('Example'), '2;1');
FILE_NAME ('fileName', '2011-11-11T11:11:11', ('author'), ('organizationName'), 'engineVersion', 'Windows
System', 'authorAuthorization');
FILE_SCHEMA (('ACTORRESOURCE'));
ENDSEC;
DATA;
#1=
PERSON(1,'Petrovvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv','Petr',('Petrovitc
h'),$,(#2,#3),$);
#2= ADDRESS(OFFICE,$,...);
#3= POSTALADDRESS(USERDEFINED,$,...);
#4= PERSON(1,'Ivanov', 'Ivan' ,('Ivanovitch'),$,(#5,#5,#7),$);
#5= POSTALADDRESS(OFFICE,$,...);
#6= TELECOMADDRESS(OFFICE,$,...);
#7= POSTALADDRESS($, $, ...);
ENDSEC;
END-ISO-10303-21;
```

SemanticSTEP

Model verification

- Overdetermined schema (Are there any data sets satisfying the specified constraints?)
  Instantiation of every concrete data type
  Instantiation of every mandatory (and/or optional) attribute
- Underdetermined schema (Are there any data sets violating the specified constraints?)

```
ISO-10303-21;

HEADER;
FILE_DESCRIPTION (('Example'), '2;1');
FILE_NAME ('fileName', '2011-11-11T11:11:11', ('author'), ('organizationName'), 'engineVersion', 'Windows
System', 'authorAuthorization');
FILE_SCHEMA (('ACTORRESOURCE'));
ENDSEC;

DATA;
#1= PERSON(1,'Petrov','Petr',('Petrovitch'),$,(#2,#3),$);
#2= POSTALADDRESS(OFFICE,$,...);
#3= POSTALADDRESS(HOME,$,...);
#4= PERSON(2,'Ivanov‘,'Ivan',('Ivanovitch'),$,(#5,#6,#7),$);
#5= POSTALADDRESS(OFFICE,$,...);
#6= TELECOMADDRESS(OFFICE,$,...);
#7= POSTALADDRESS(HOME,$,...);
ENDSEC;

END-ISO-10303-21;
```

SemanticSTEP

Test generation

- Representative tests (large combinations of typed data belonging to the underlying schema or ist given subschemas)
- Positive/negative tests (small data sets generated individually for every specified constraint)

```
DATA;
#1= PERSON(1,'Petrov','Petr',('Petrovitch'),$,(#2,#3),$);
#2= POSTALADDRESS(OFFICE,$,...);
#3= POSTALADDRESS(HOME,$,...);
#4= PERSON(2,'Ivanov‘,'Ivan',('Ivanovitch'),$,(#5,#6,#7),$);
#5= POSTALADDRESS(OFFICE,$,...);
#6= TELECOMADDRESS(OFFICE,$,...);
#7= POSTALADDRESS(HOME,$,...);
ENDSEC;
```

```
DATA;
#1=
PERSON(1,'Petrovvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv','Petr',('Petrovitch')
,$,(#2,#3),$);
#2= POSTALADDRESS(OFFICE,$,...);
#3= POSTALADDRESS(HOME,$,...);
#4= PERSON(2,'Ivanov‘,'Ivan',('Ivanovitch'),$,(#5,#6,#7),$);
#5= POSTALADDRESS(OFFICE,$,...);
#6= TELECOMADDRESS(OFFICE,$,...);
#7= POSTALADDRESS(HOME,$,...);
ENDSEC;
```

SemanticSTEP

Semantic diff/merge of divergent replicas

- Original version

```
DATA;
#1=
PERSON(1,'Petrov','Petr',('Petrovitch'),$,(#2,#3),$);
#2= POSTALADDRESS(HOME,$,...);
#3= POSTALADDRESS(OFFICE,$,...);
ENDSEC;
```

- Version 1

```
DATA;
#1=
PERSON(1,'Petrov','Petr',('Petrovitch'),$,(#2,#3,#4),
$);
#2= POSTALADDRESS(HOME,$,...);
#3= POSTALADDRESS(OFFICE,$,...);
#4= TELECOMADDRESS(HOME,$,...);
ENDSEC;
```

- Version 2

```
DATA;
#1=
PERSON(1,'Petrov','Petr',('Petrovitch'),$,(#2,#3),$);
#2= POSTALADDRESS(USERDEFINED, 'Partial Home
Working',...);
```

- Final version semantically concordant with the schema

```
DATA;
#1= PERSON(1,'Petrov','Petr',('Petrovitch'),$,(#2,#4),$);
#2= POSTALADDRESS(USERDEFINED, 'Partial Home Working',...);
#4= TELECOMADDRESS(HOME,$,...);
ENDSEC;
```

Product Data Management

Concurrency versus consistency

ACID principles

- Atomicity, Consistency, Isolation, Durability

Transaction anomalies

- Dirty read, write, fuzzy read
- Phantom, lost update, skew read, write

Advanced transaction models

- Nested transactions
- Sagas
- Specific access patterns
- Altruistic & Constrained shared locks
- Multi-version transactions

Product Data Management

Optimistic replication

Long-lived transactions

- Atomicity degradation
- Isolation disturbance

Replication advantages

- High availability and concurrency
- Small costs on multi-access

Replication drawback

- Problem to reconcile divergent replicas in consistent and meaningful way

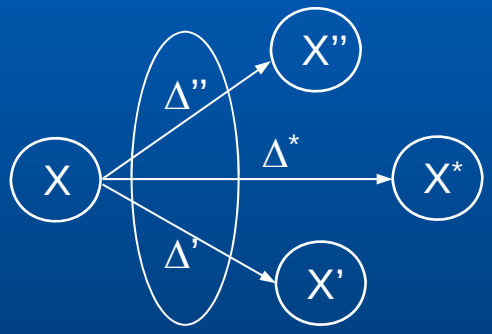
Successful applications

- UseNET, PDA, Bayou, CVS, Synergy, ClearCase, Perforce

SemanticSTEP

Semantic reconciliation method

- Model verification
- Test generation
- Data correction
- Semantic diff/merge



$X, X', X'' \in Model$, $\Delta' = delta(X', X)$, $\Delta'' = delta(X'', X)$

$\Delta^* = merge(\Delta', \Delta'')$, $X^* = apply(X, \Delta^*)$,

where $\Delta', \Delta'', \Delta^* = \{ \delta \}$

δ := new(obj) | del(obj) | move(obj) | mod(obj.attr) | ins(obj.coll,i) | rem(obj.coll,i) | prm(obj.coll,i,j) | trans(obj.coll,i)

Formal statements

$\max(card(\Delta^*)) \mid \Delta^* \subseteq \Delta' \cup \Delta''; X, X', X'' \in C(Model) \rightarrow X^* \in C(Model)$

$\min(card(\Delta^*)) \mid X \in Model \rightarrow X^* \in C(Model)$

STEP Standards

IFC by International Alliance for Interoperability

9 geographical chapters world-wide,
membership of over 400 companies from over 20 countries

Architecture & Engineering & Construction & Facility Management

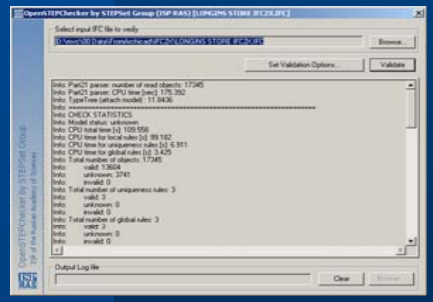
INCREASING MODEL COMPLEXITY

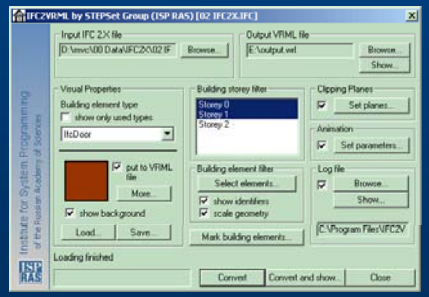
Statistics / EXPRESS schemas	PLCS	CIS 2	IFC 1.5	IFC 2.0	IFC 2x	IFC 2x2	IFC 2x3	IFC 2x4
Object types	459	731	186	290	370	623	653	775
Constructed & defined types	103	175	95	157	228	312	343	407
Derived attributes	65	198	44	45	41	55	96	106
Functions/procedures	2	73	25	27	25	37	38	42
Where rules in object types	55	511	107	168	196	271	337	638
Where rules in defined types	173	20	12	12	13	16	24	24
Uniqueness rules	8	50	1	3	14	14	17	4
Global rules	4	1	0	0	3	3	2	2

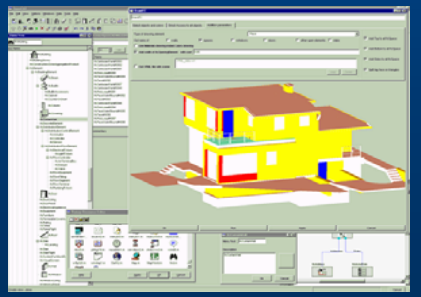
SemanticSTEP

IFC utilities

- **Semantic Checker**
  - Control of exchanged IFC files
- **Test Generator**
  - Development and certification of IFC-compliant applications
- **Merger**
  - Effective collaborative solution for both client-server and stand-alone applications
- **VRML Converter**
  - Web-publishing of IFC-driven data as visual scenes





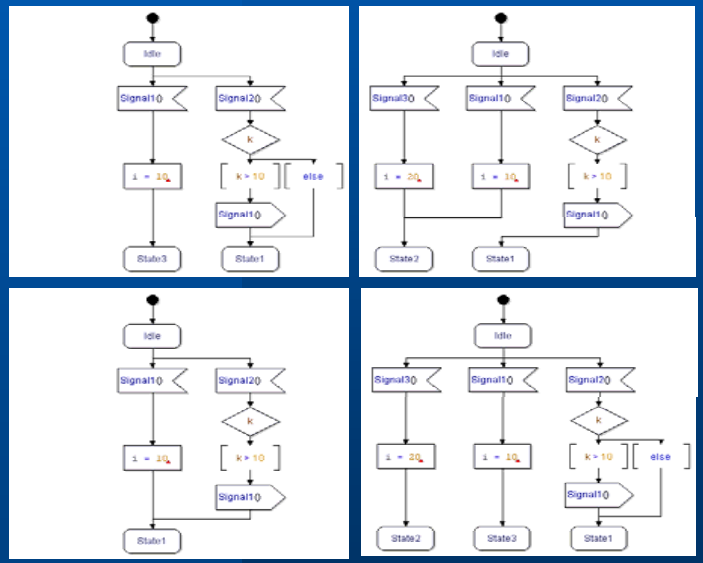


SemanticSTEP

Telelogic TAU Studio

- Semantic Diff/Merge of UML diagrams

Statechart1 Statechart2 Sequence diagram1 Sequence diagram2

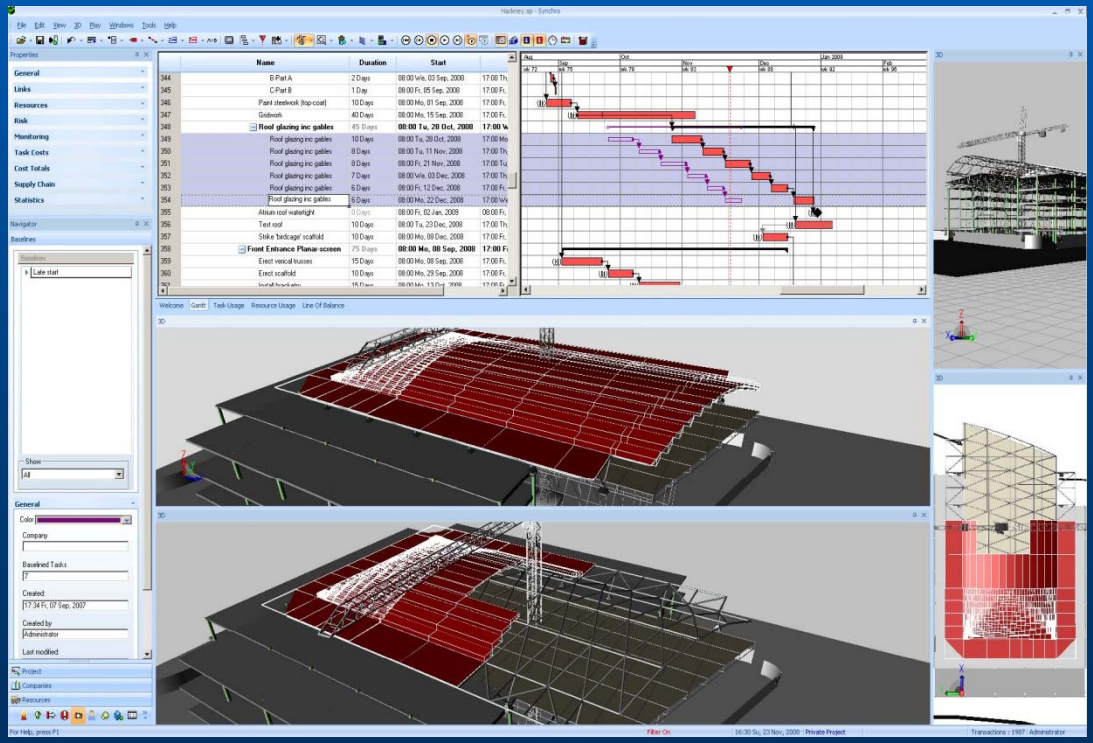


Original statechart Merged result Original sequence diagram Result

SemanticSTEP

Synchro (5D modelling & planning system)

- Integration with CAD and PM
  - Export
  - Import
  - SynchroniseFrom
  - SynchroniseTo



SemanticSTEP

THANKS for attention

PLEASE, any questions

PLEASE, more questions..