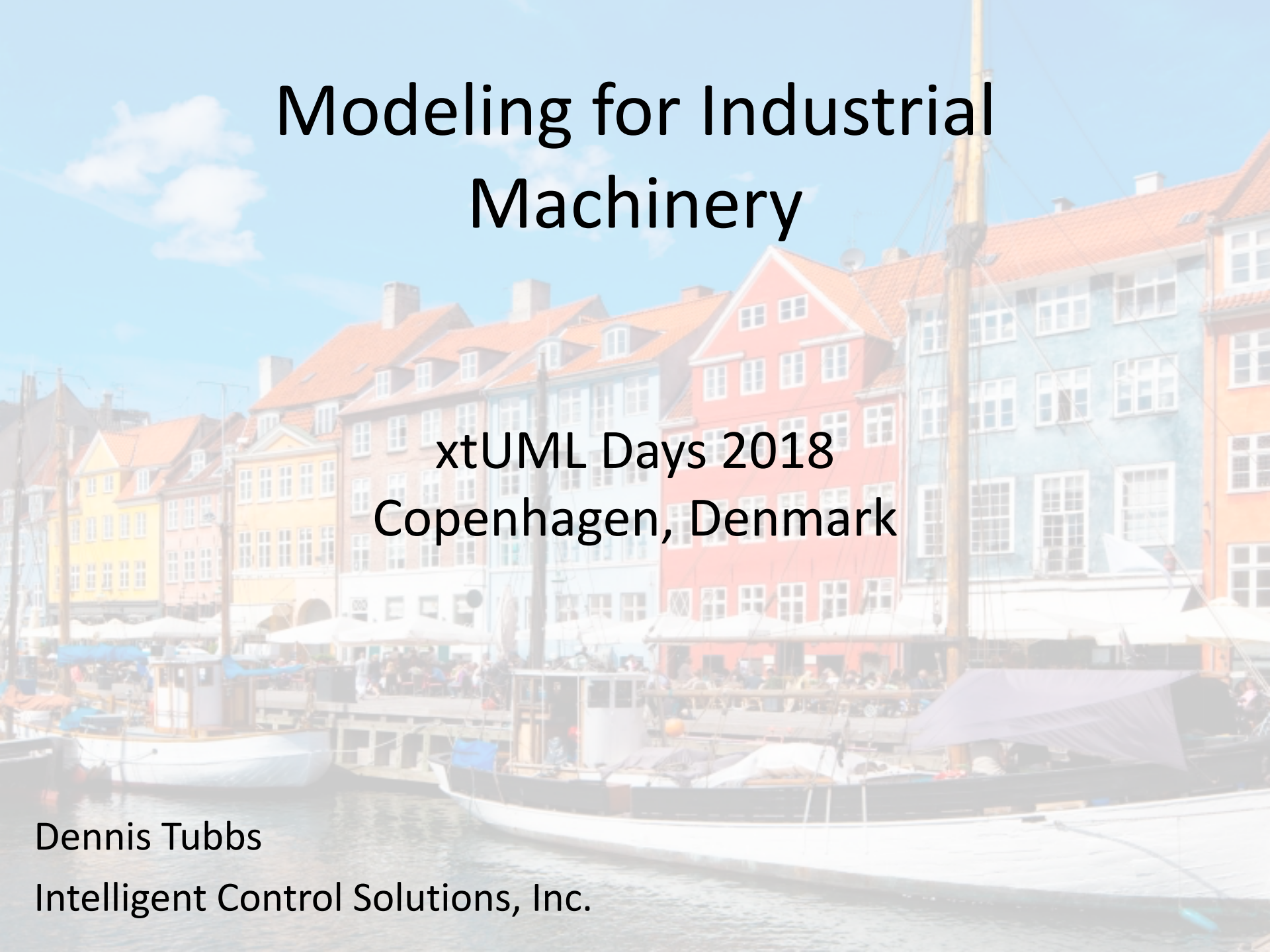


Modeling for Industrial Machinery

xtUML Days 2018
Copenhagen, Denmark

Dennis Tubbs

Intelligent Control Solutions, Inc.

Synopsis

- Introduction
- Corrugated Slitters
- Project Goals
- Model
- Translation (Structured Text & C#)

Introduction

- Started post college career at Marquip, Inc.
- Introduced to S/M in 1994 at the Embedded Systems Conference in Atlanta where I attended one of Steve Mellor's presentations.
- 1 week course at Project Technology
- Read every book and paper on Shlaer-Mellor.
- Contractor / Consultant
- Worked in Industrial Equipment and Medical Device Industries

BW Papersystems

MarquipWardUnited

Designs, manufactures and services full corrugators, including VortX Starch Kitchens



Complete line of corrugator equipment from starch kitchens to stackers with speeds up to 460 MPM (1500 FPM) and machine widths up to 2.85 m (112 in.).

Locations

Phillips, WI (USA)

Milton, PA (USA)

Budapest, Hungary

Equipment

Full Corrugators:

Splicers

Singlefacers

Doublefacers

Slitter/Scorers

Knives

Stackers

BW Papersystems

MarquipWardUnited

Designs, manufactures and services Flexo Folder Gluers & Rotary Die Cutters. Curioni specializes in mini & jumbo sizes





The modern ServoGrafix Rotary Die Cutter, a true-Servo machine offers superior graphics and registration which consistently delivers exceptional quality, we maintain the highest standards of safety, versatility, and durability in the industry.

Locations

Baltimore, MD (USA)

Milan, Italy

Equipment

Rotary Die Cutters

Flexo Folder Gluers

BW Papersystems

Curioni

BW Papersystems

MarquipWardUnited

Mid-size sheeter equipment for paper mills, folding carton plants, and converters



Locations

Phillips, WI (USA)

Equipment

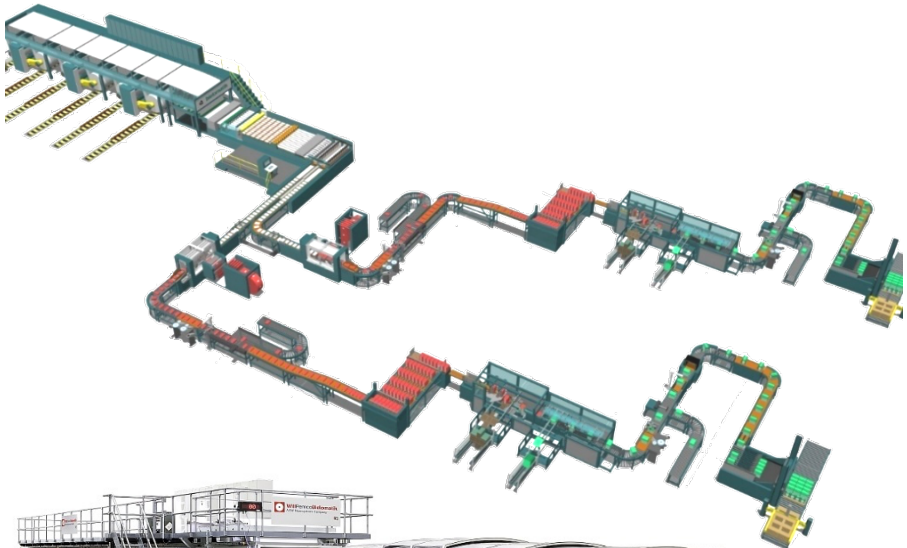
Folio Sheeters



BW Papersystems

WillPemcoBielomatik

Manufactures machinery for the paper converting, manufacturing and packaging industries







Locations

Hamburg, Germany

Sheboygan, WI (USA)

Equipment

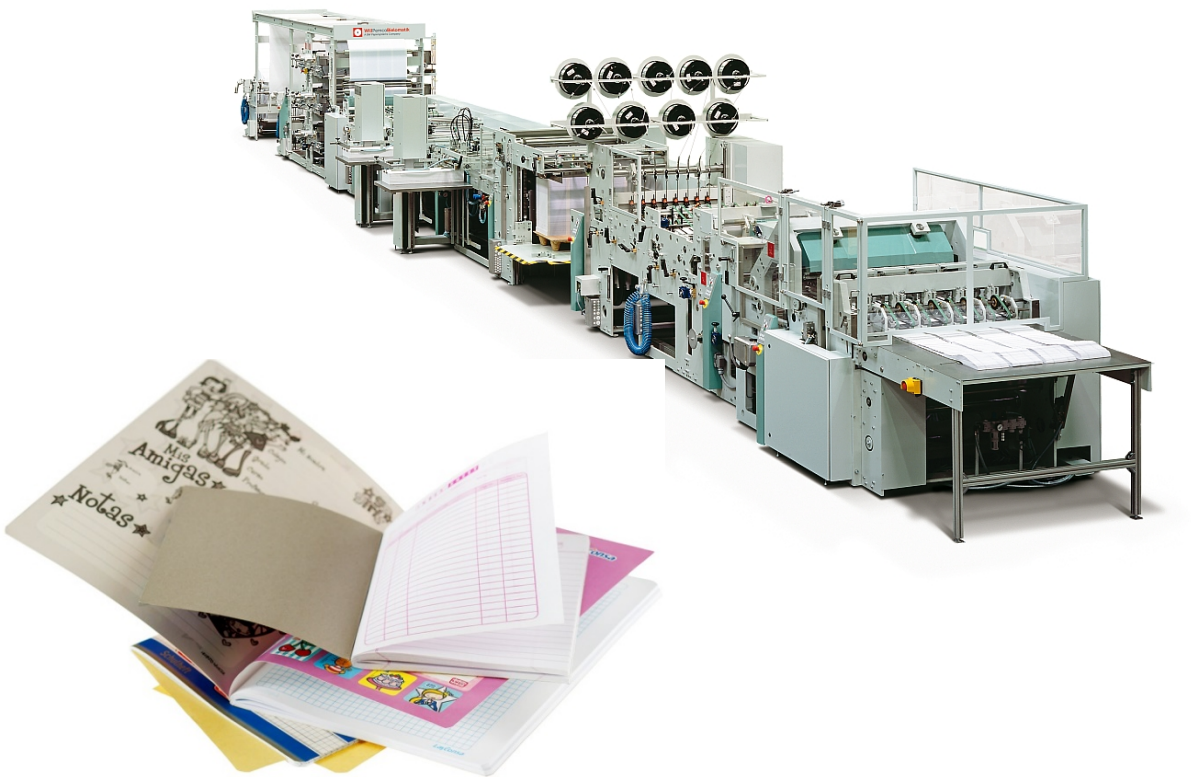
Cut-Size Sheeters

Cut-size Ream Wrappers

BW Papersystems

WillPemcoBielomatik

Equipment for the stationery industry



Locations

Hamburg, Germany

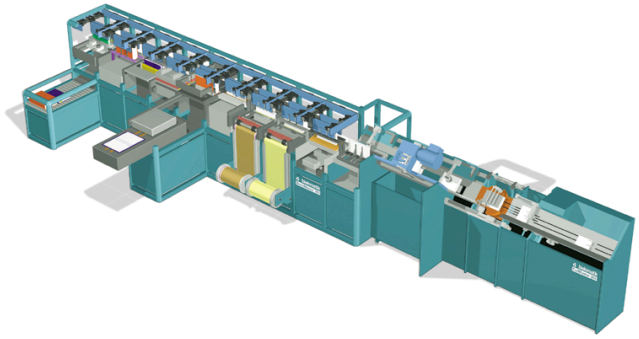
Equipment

Exercise book production machines

BW Papersystems

BW Bielomatik

Applications for the Digital Printing







Locations

Stuttgart, Germany

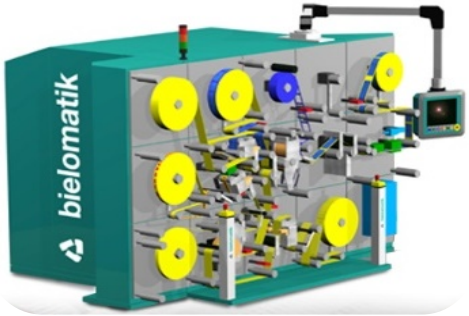
Equipment

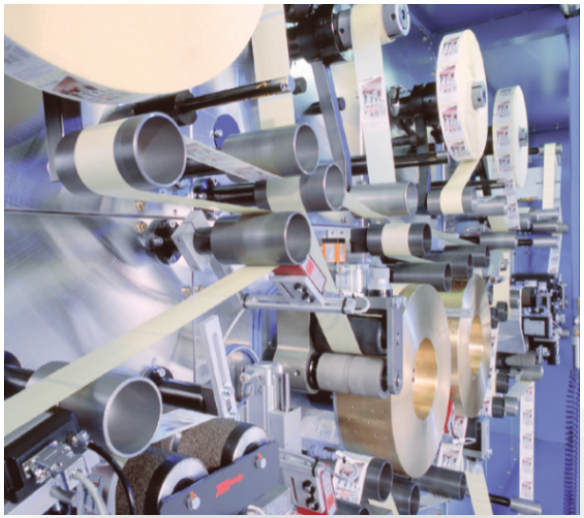
Print-Finishing-on-Demand for Book-on-Demand binding machines

BW Papersystems

BW Bielomatik

RFID/Transponder processing Industry







Locations

Stuttgart, Germany

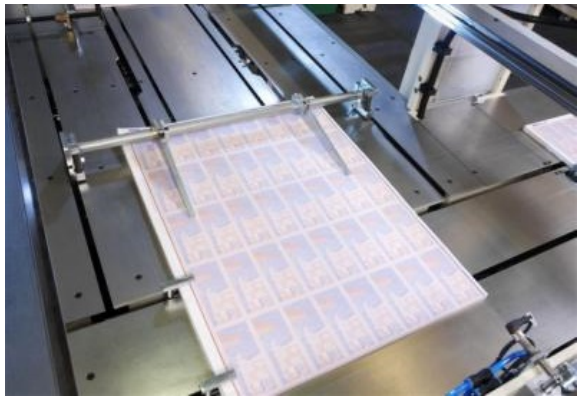
Equipment

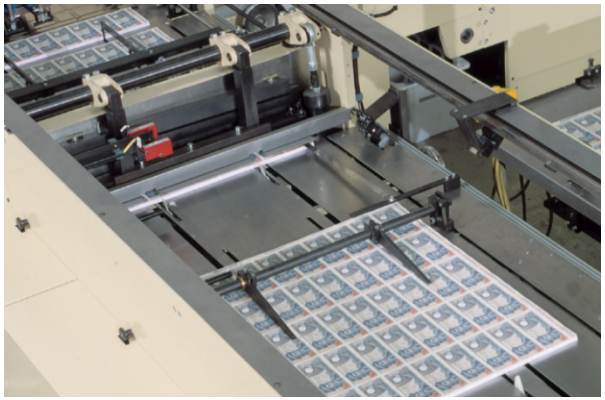
RFID Transponder for laminating and finishing technologies

BW Papersystems

BW Bielomatik

Paper Finishing









Locations

Stuttgart, Germany

Equipment

Register-Cut sheeters for security and banknote papers.

Machines for cutting, collecting, counting and banding of banknotes

BW Papersystems

Kugler-Womako

Security Documents



Locations

Stuttgart, Germany

Equipment

World market leader in the manufacturing of passport machines

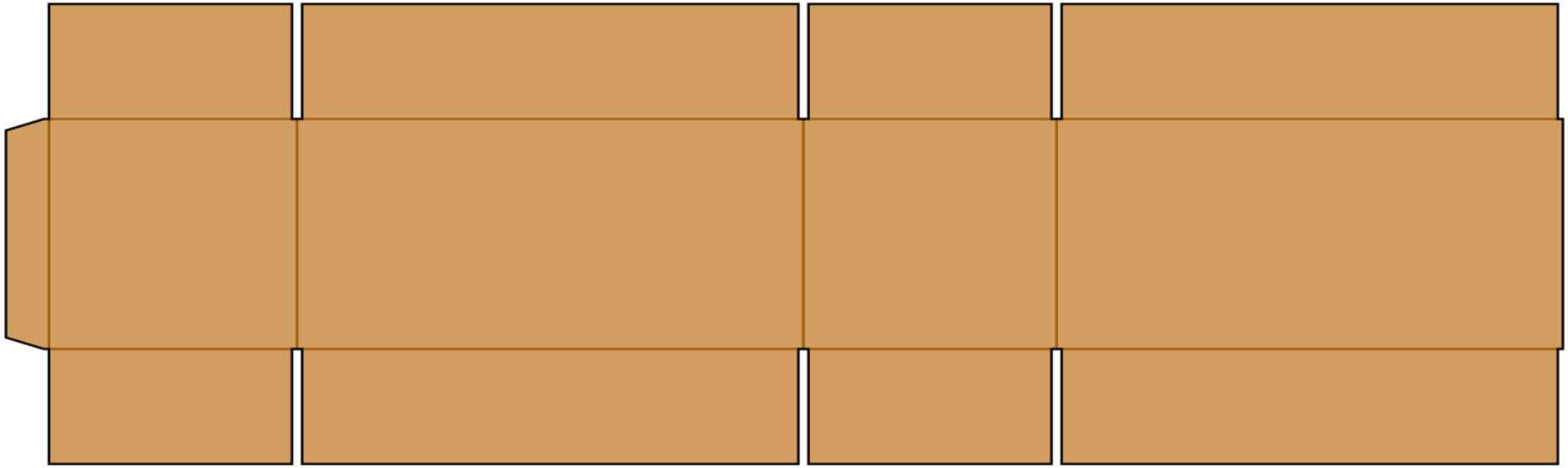
Corrugator

- What is a corrugator?
  - A combination of machines that convert rolls of paper into a blank sheet for making a box.
  - https://www.youtube.com/watch?v=C5nNUPNvWAw
  - Show video

Slitter / Scorer

- What is the role of a slitter/scorer in a corrugator?
  - Score the web so the box will fold.
  - Slit the web to the width required for each box.

Making a Box – Slits & Scores



Quickset Slitters

- Model IV Quickset Slitter/Scorer – 1998
  - Servo motor for every tool
  - Requires a gap in the web at setup change
  - Show Video

Model IV Quickset Slitter/Scorer

- Modeled with BridgePoint
- C++ Architecture
  - Used for coordination of servo controllers
  - Out of the box
  - Large event queue
- Object Dump

Model IV Quickset Slitter/Scorer

- C Architecture
  - Servo control
  - Customized
  - Event queue was replaced by inline functions
- Target an embedded 68k based SBC

Model IV Quickset Slitter/Scorer

- Success
  - It Works!!! Very High Reliability & Efficiency
  - Fail early in development lifecycle
  - Collaboration among team members
- Areas for Improvement
  - Difficult to troubleshoot hardware failures
  - Colleagues didn’t understand modeling
  - OOA received bad rap by association

Robotic Slitters

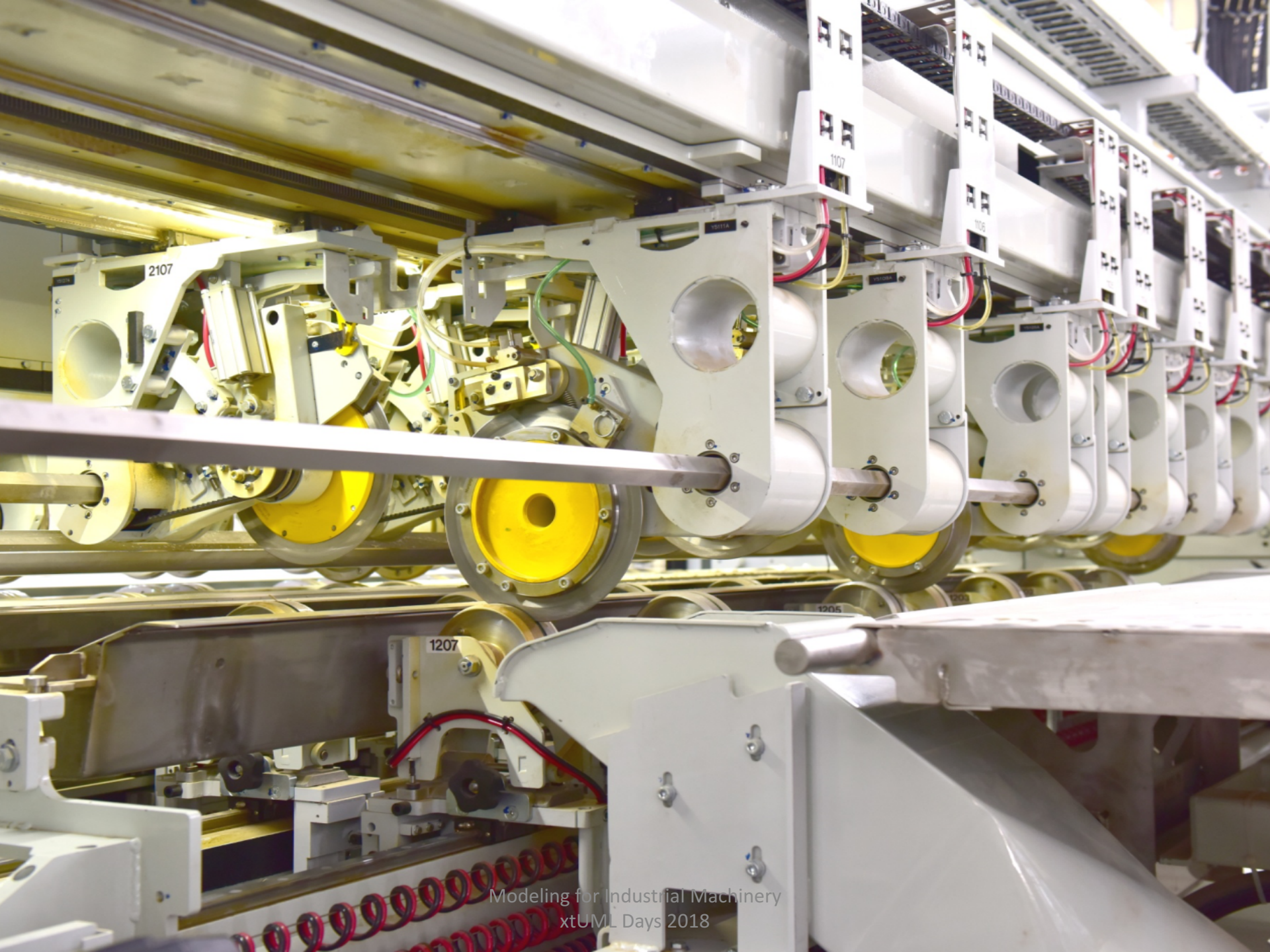
- RPSE Slitter/Scorer
  - Robots position tools
  - Duplicate sections plunge in/out of the web
  - Software was not modeled
  - Show Video



Modeling for Industrial Machinery
xtUML Days 2018


SAFETY INFORMATION
4101
4102
4105
4106
4107
4108
4109


2107
2201
2202
1207
1206
1205



Modeling for Industrial Machinery
xtUML Days 2018

Current Project

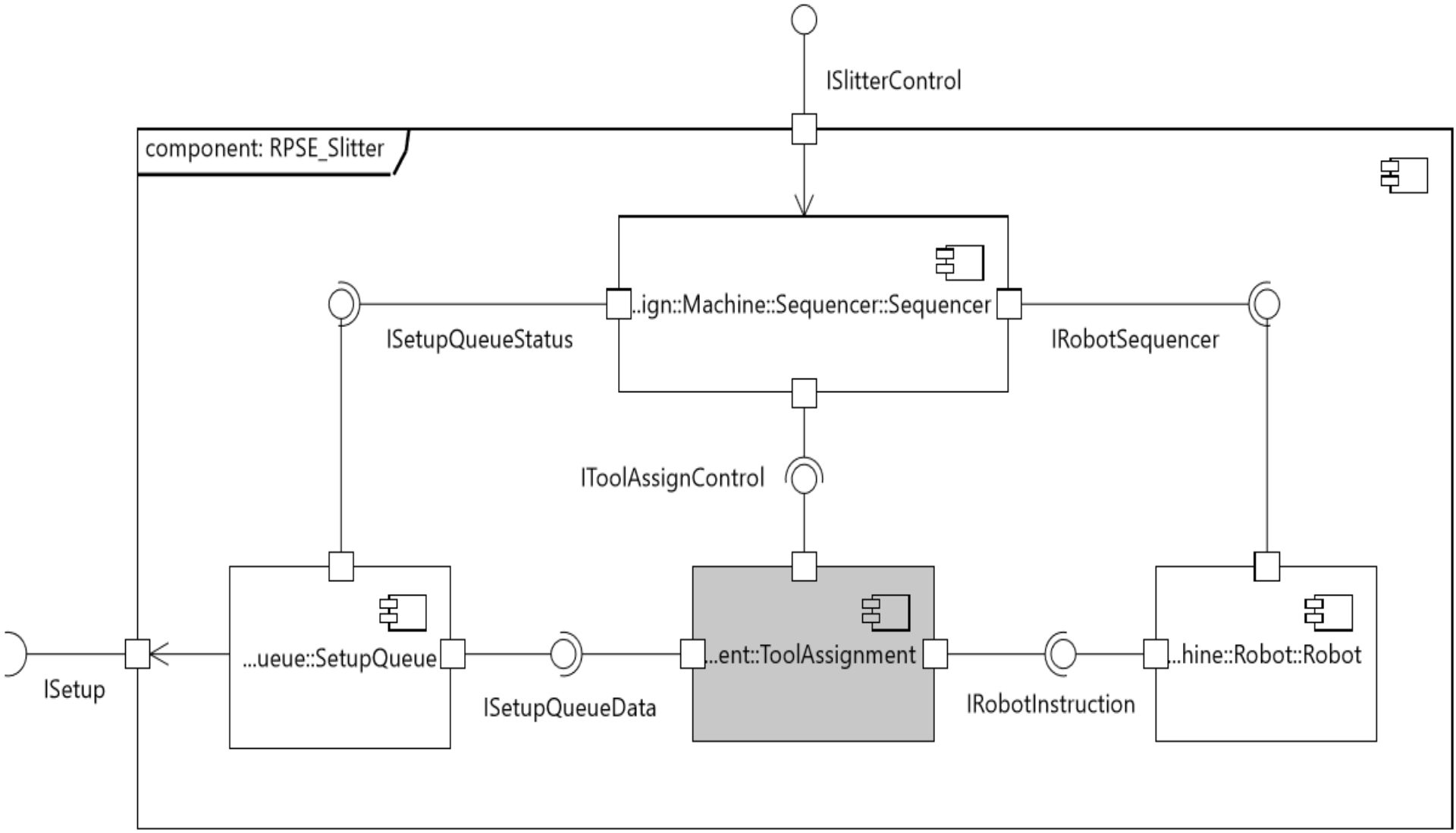
- Require – Faster recovery time on RPSE
- Problem – Tool positions are calculated remotely
  - Large amount of data flowing back and forth
  - Interface communication slow (OPC)

Current Project

- Solution – Move tool position calculations
  - Create lists of Robot Instructions
  - Simplify interface
  - Reduce affects of time with Look-Ahead

- What makes this solution possible now?
  - Target platform is now capable, hardware and programming tool improvements.

Component Model

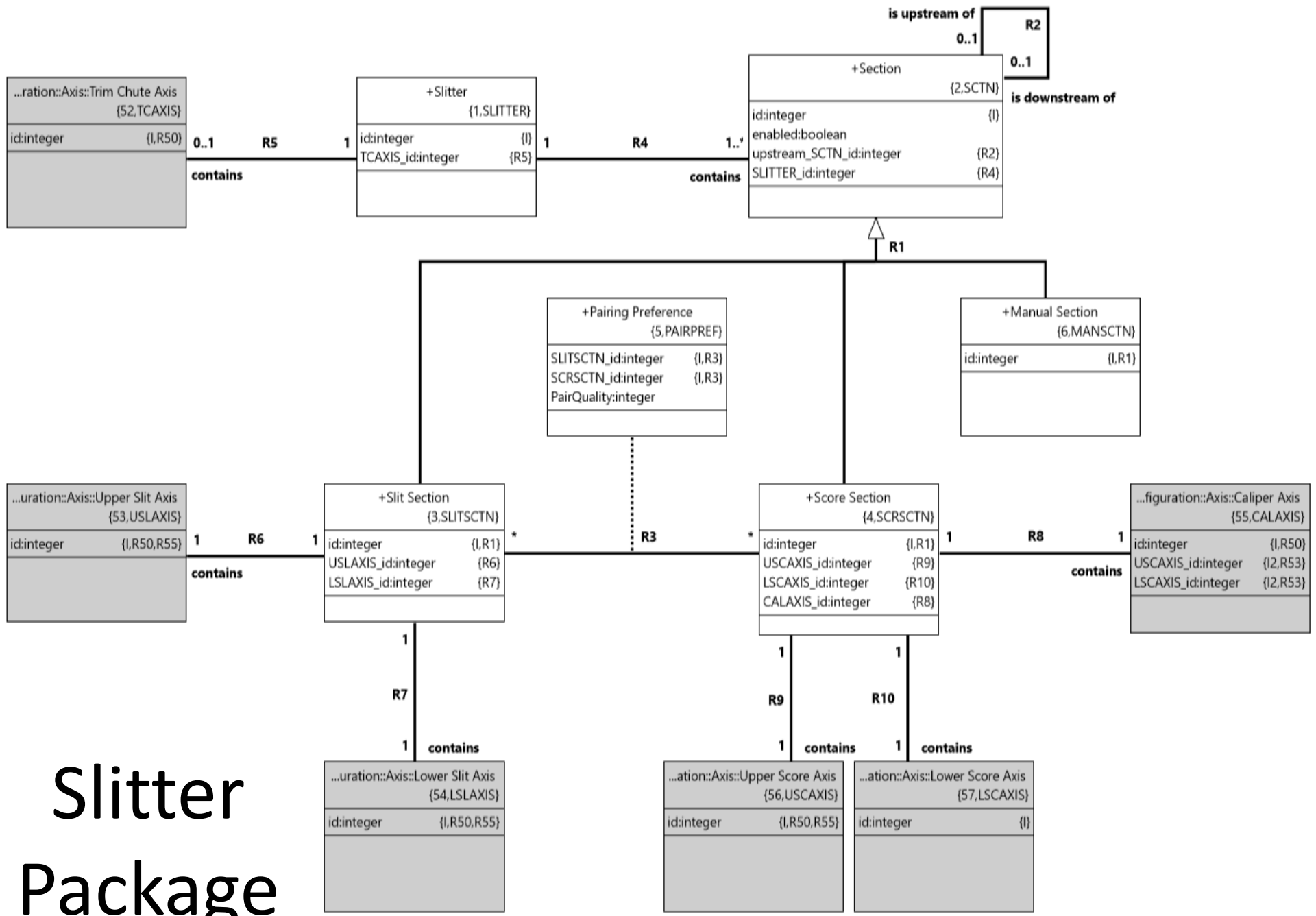


Project Goals

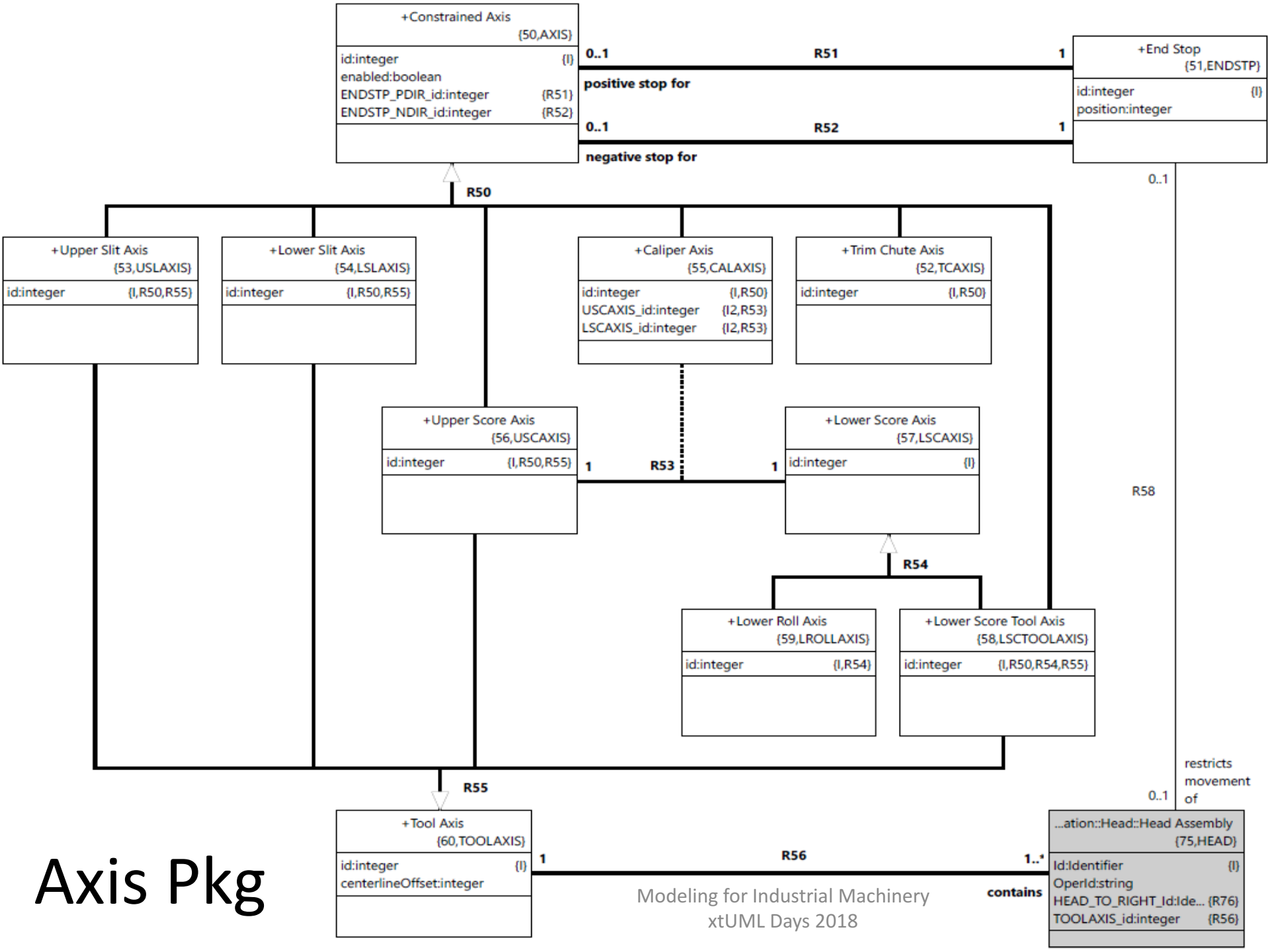
- Implement Tool Assignment Algorithms in PLC
- Improve the Algorithms
- Implement Tool Assignment Algorithms in C#
- Partial Translation
- Use Model for documentation
- Code
  - Structured Text, C#
  - Human Readable!

Model – Class Packages

- Configuration Packages
  - Slitter
  - Axis
  - Score Profile
  - Robot

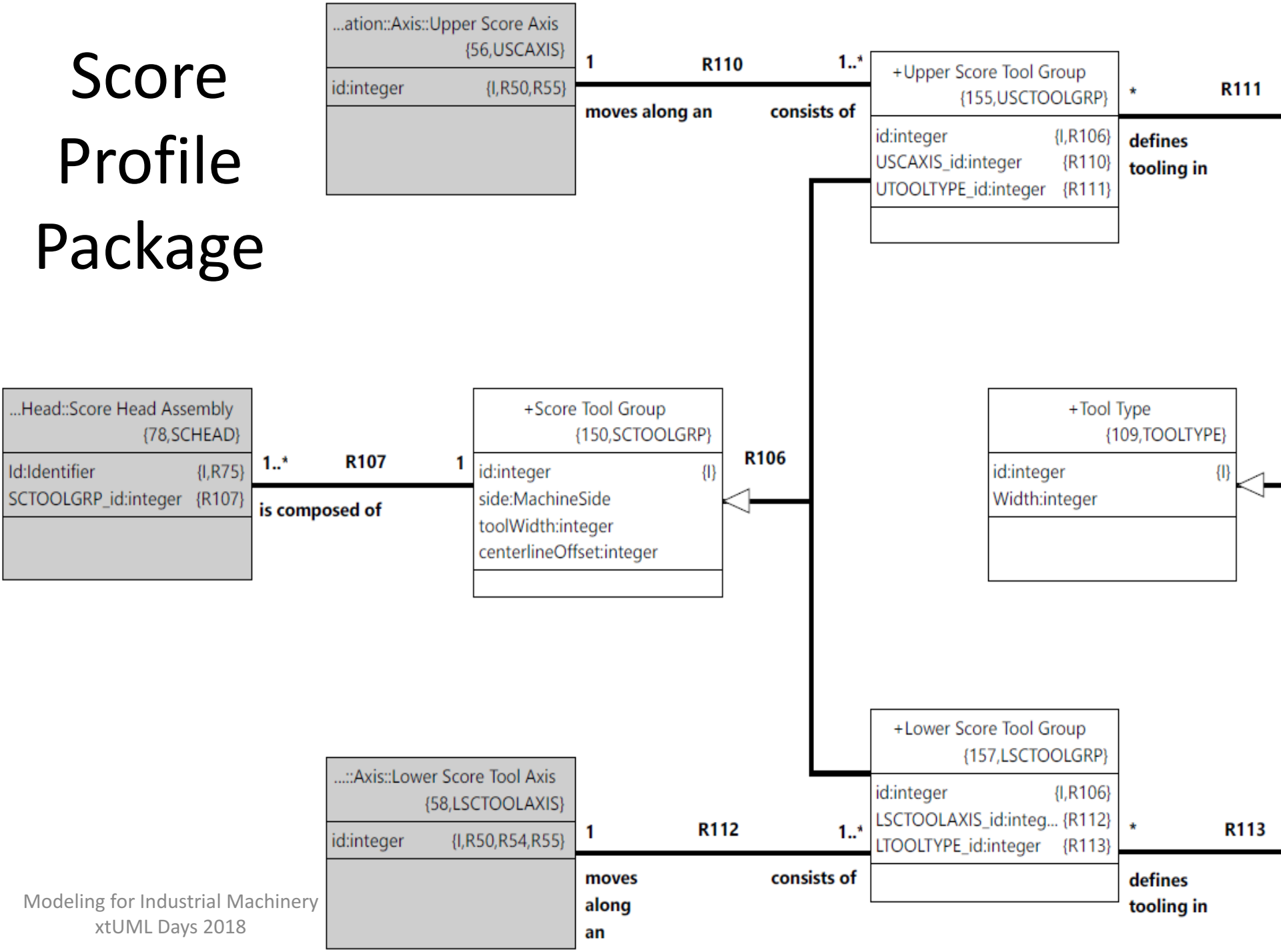


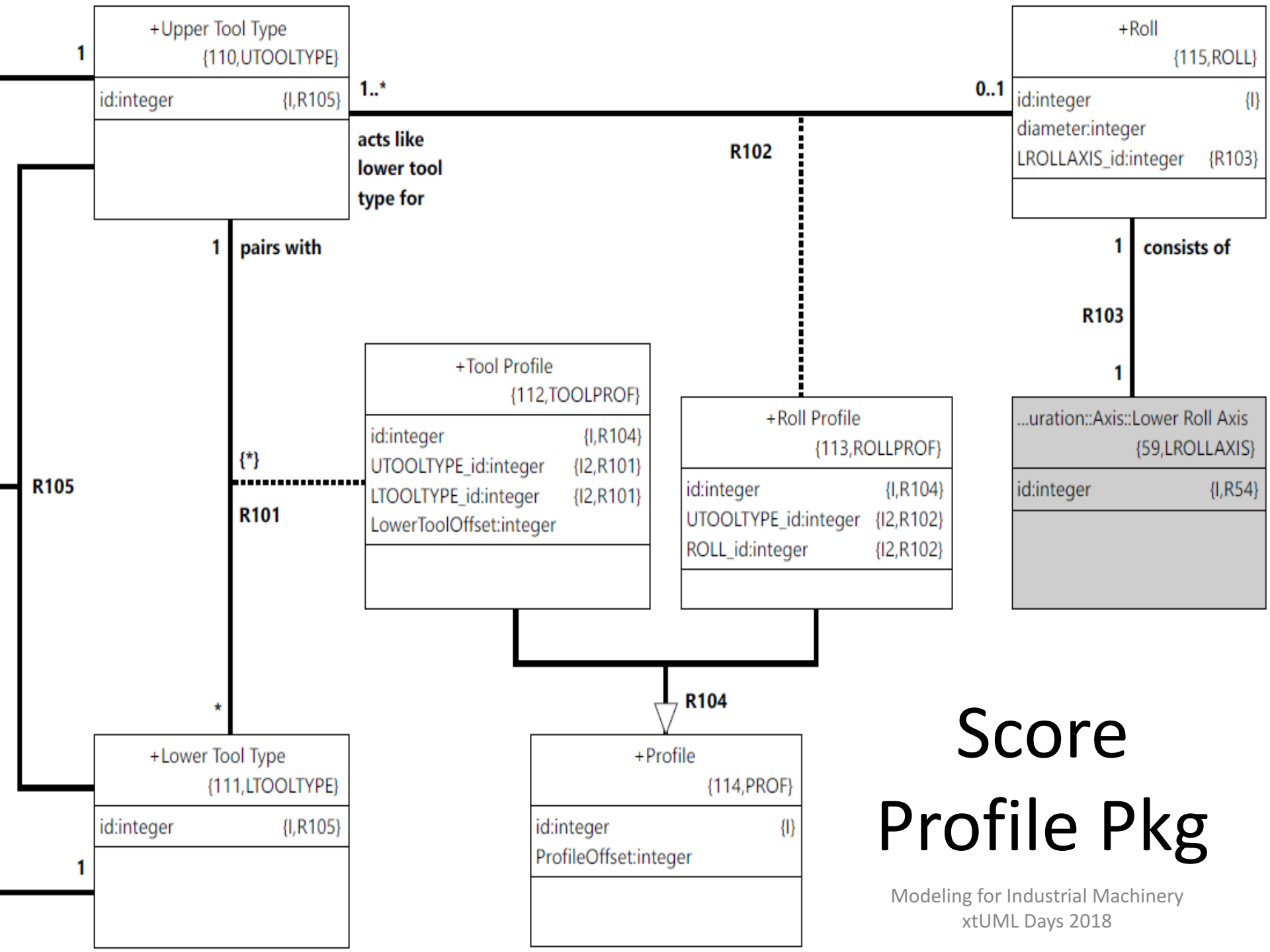
Slitter Package


+Constrained Axis
{50,AXIS}
id:integer {I}
enabled:boolean
ENDSTP_PDIR_id:integer {R51}
ENDSTP_NDIR_id:integer {R52}
0..1
R51
1
positive stop for
0..1
R52
1
negative stop for
+End Stop
{51,ENDSTP}
id:integer {I}
position:integer
0..1
R50
+Upper Slit Axis
{53,USLAXIS}
id:integer {I,R50,R55}
+Lower Slit Axis
{54,LSLAXIS}
id:integer {I,R50,R55}
+Caliper Axis
{55,CALAXIS}
id:integer {I,R50}
USCAXIS_id:integer {I2,R53}
LSCAXIS_id:integer {I2,R53}
+Trim Chute Axis
{52,TCAXIS}
id:integer {I,R50}
+Upper Score Axis
{56,USCAXIS}
id:integer {I,R50,R55}
1
R53
1
+Lower Score Axis
{57,LSCAXIS}
id:integer {I}
R58
R54
+Lower Roll Axis
{59,LROLLAXIS}
id:integer {I,R54}
+Lower Score Tool Axis
{58,LSCTOOLAXIS}
id:integer {I,R50,R54,R55}
R55
restricts
movement
of
0..1
+Tool Axis
{60,TOOLAXIS}
id:integer {I}
centerlineOffset:integer
1
R56
1..*
contains
...ation::Head::Head Assembly
{75,HEAD}
Id:Identifier {I}
OperId:string
HEAD_TO_RIGHT_Id:Ide... {R76}
TOOLAXIS_id:integer {R56}

Axis Pkg

Score Profile Package

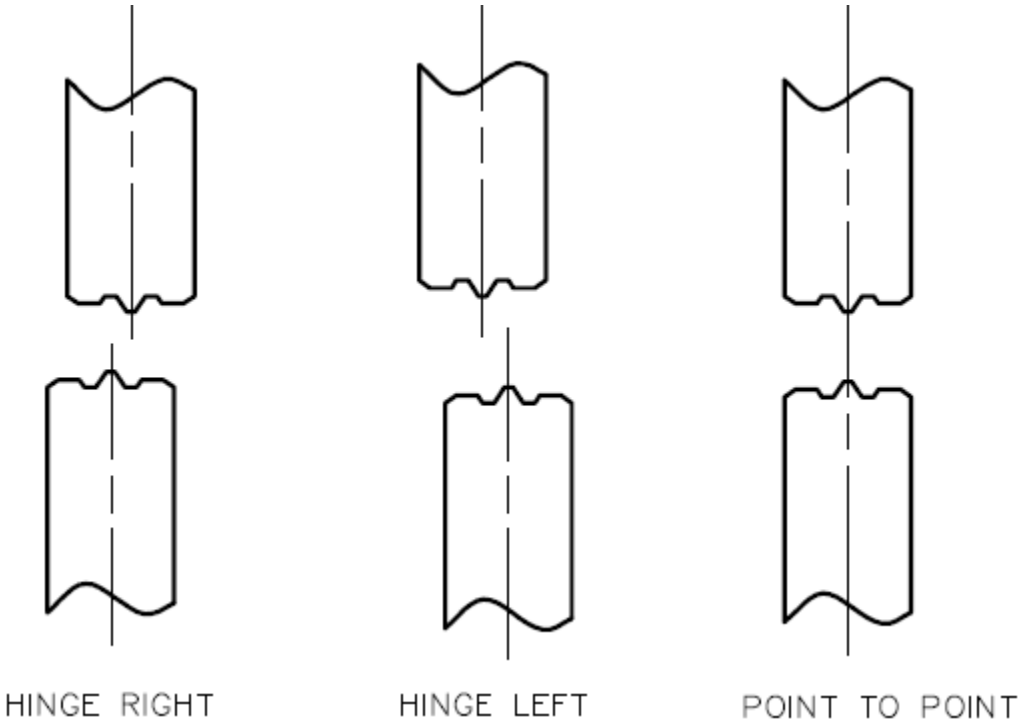




Score Profile Pkg

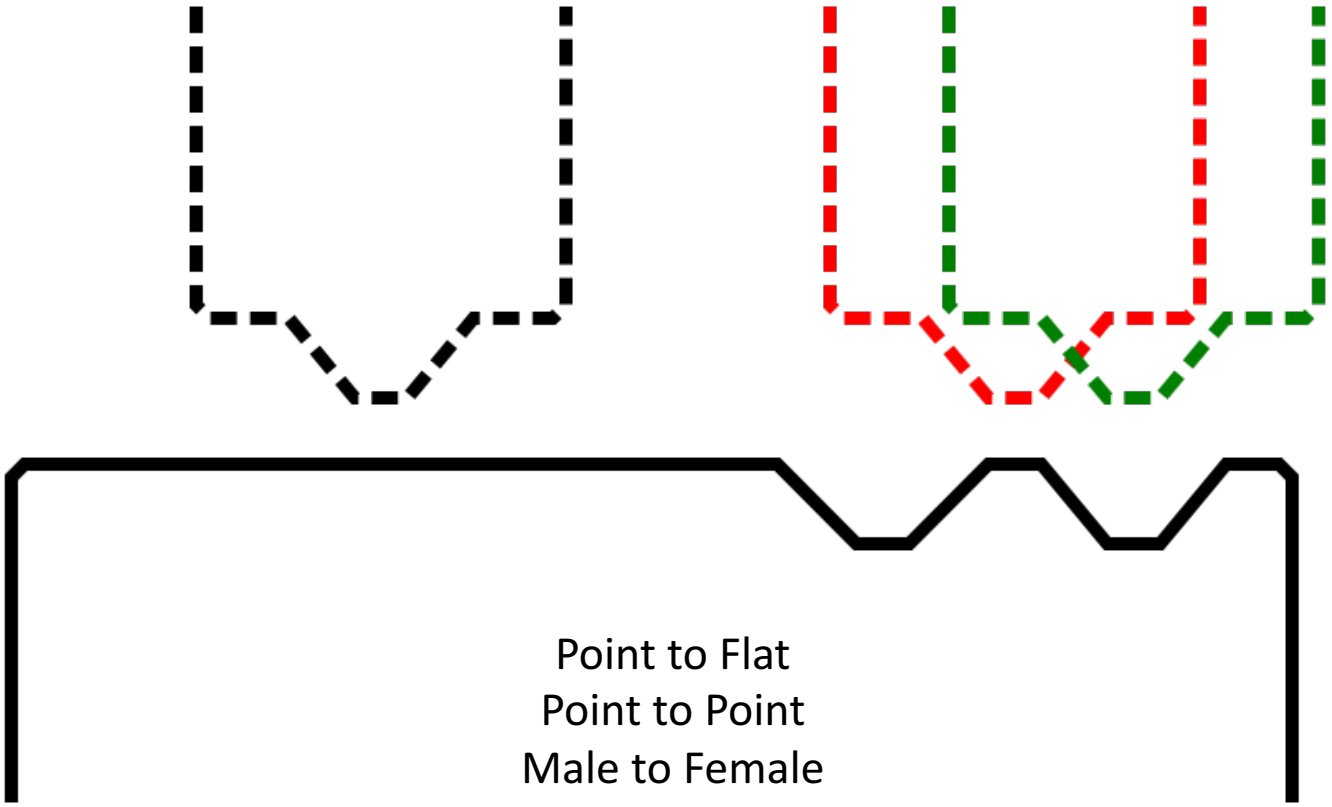
Modeling for Industrial Machinery
xtUML Days 2018

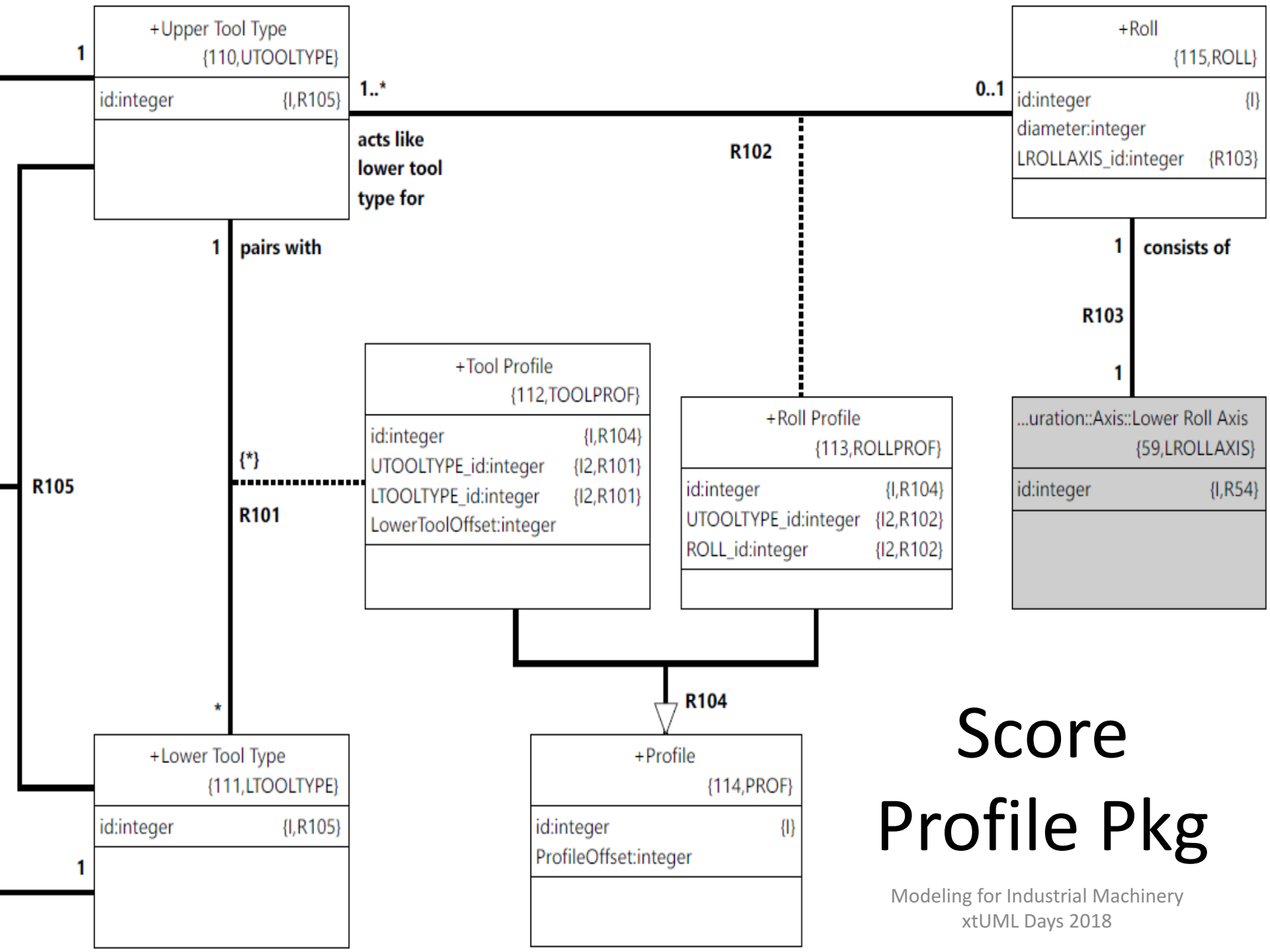
Changeable Polarity Score Profiles



NOTE: LOWER TOOLING IS IDENTICAL TO UPPER TOOLING

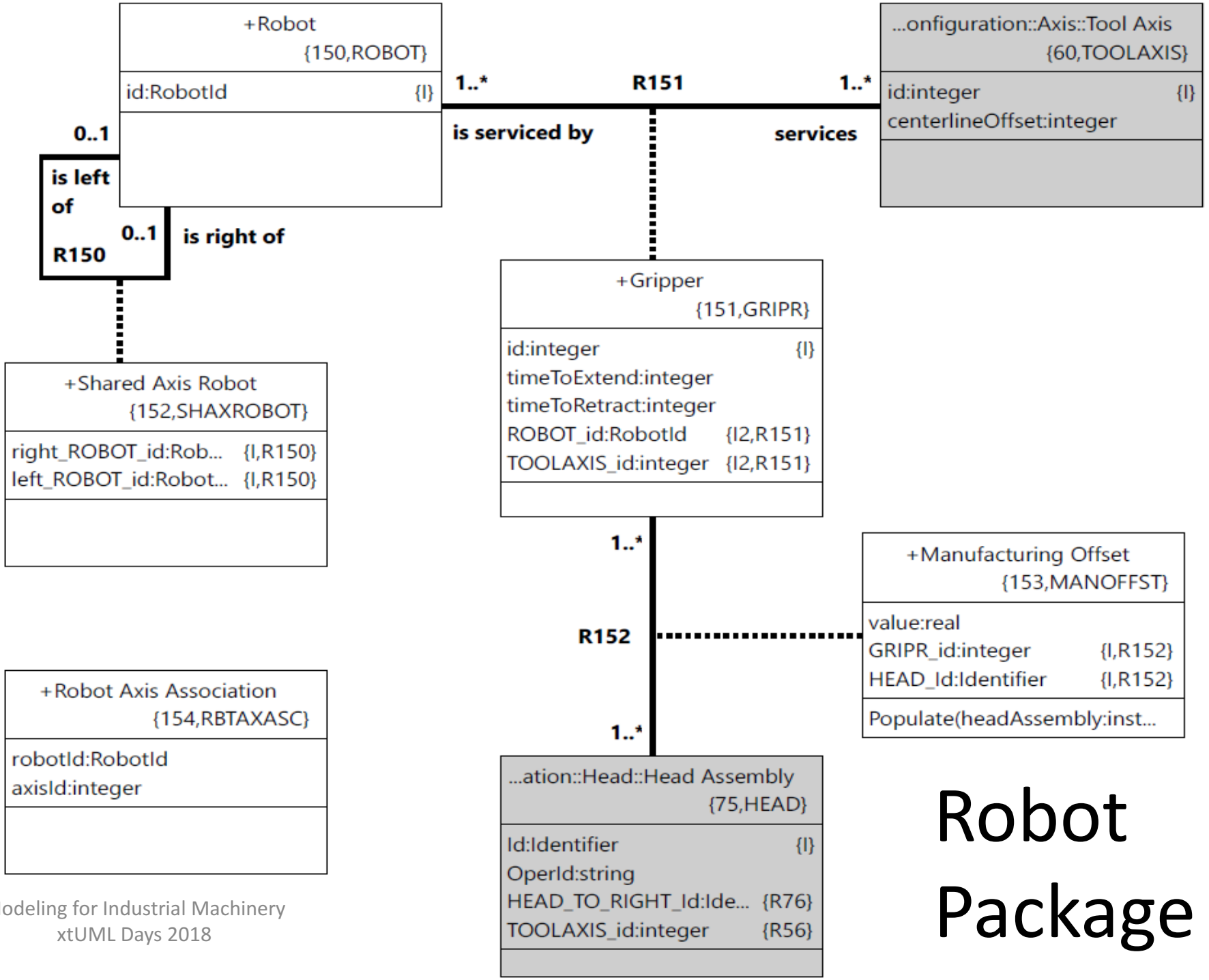
3-in-1 Score Profiles





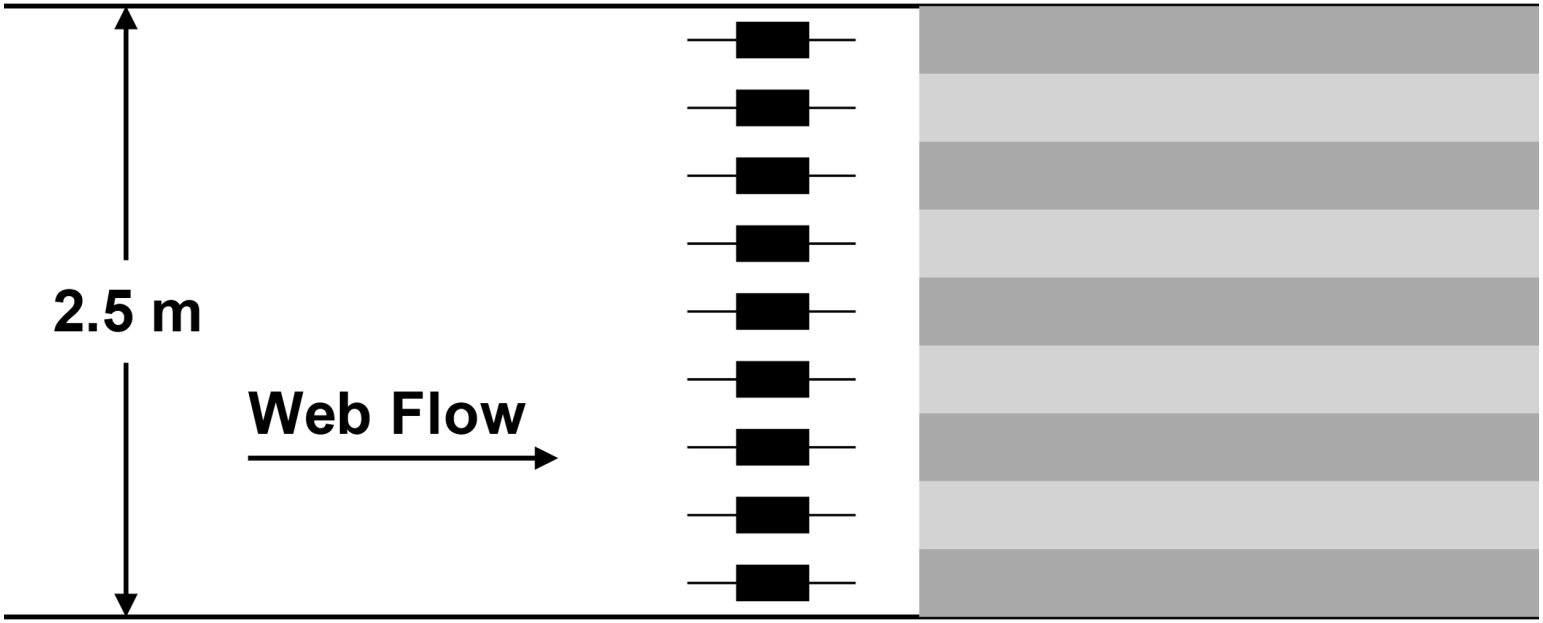
Score Profile Pkg

Modeling for Industrial Machinery
xtUML Days 2018



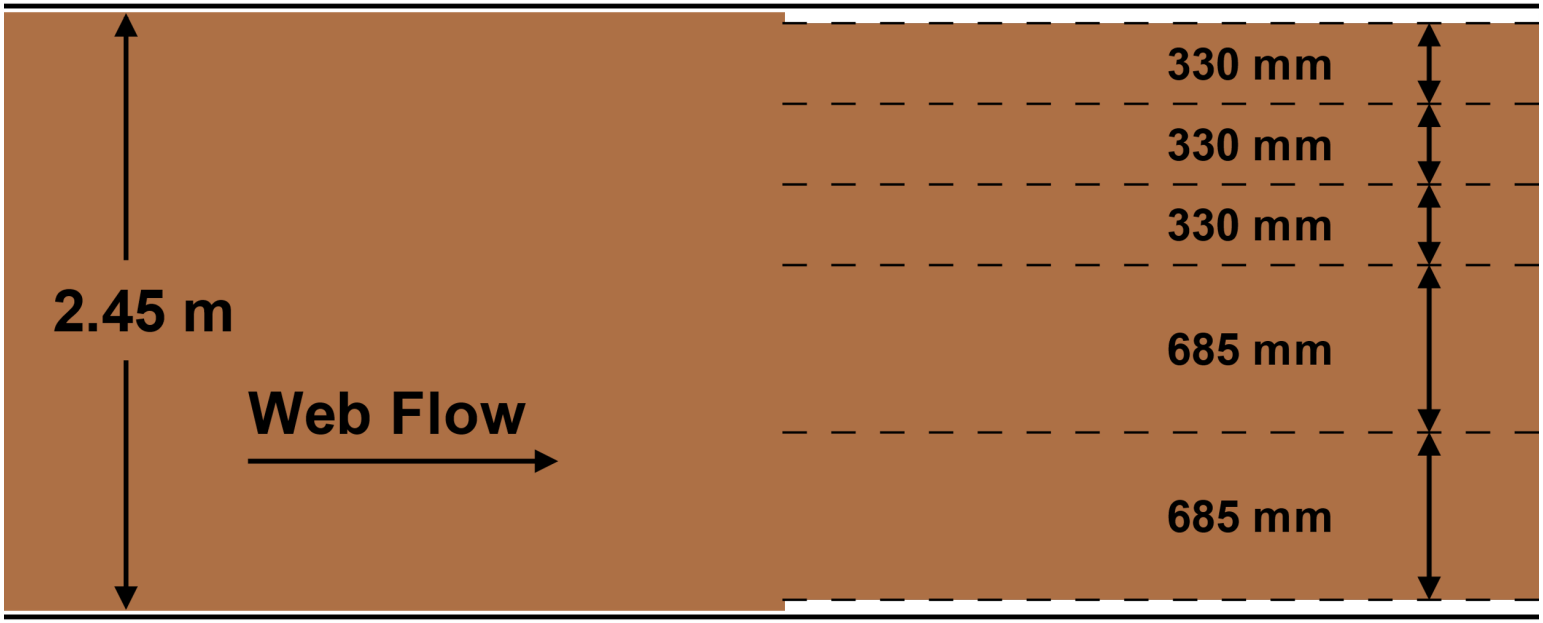
Robot Package

Slit Assignment Problem



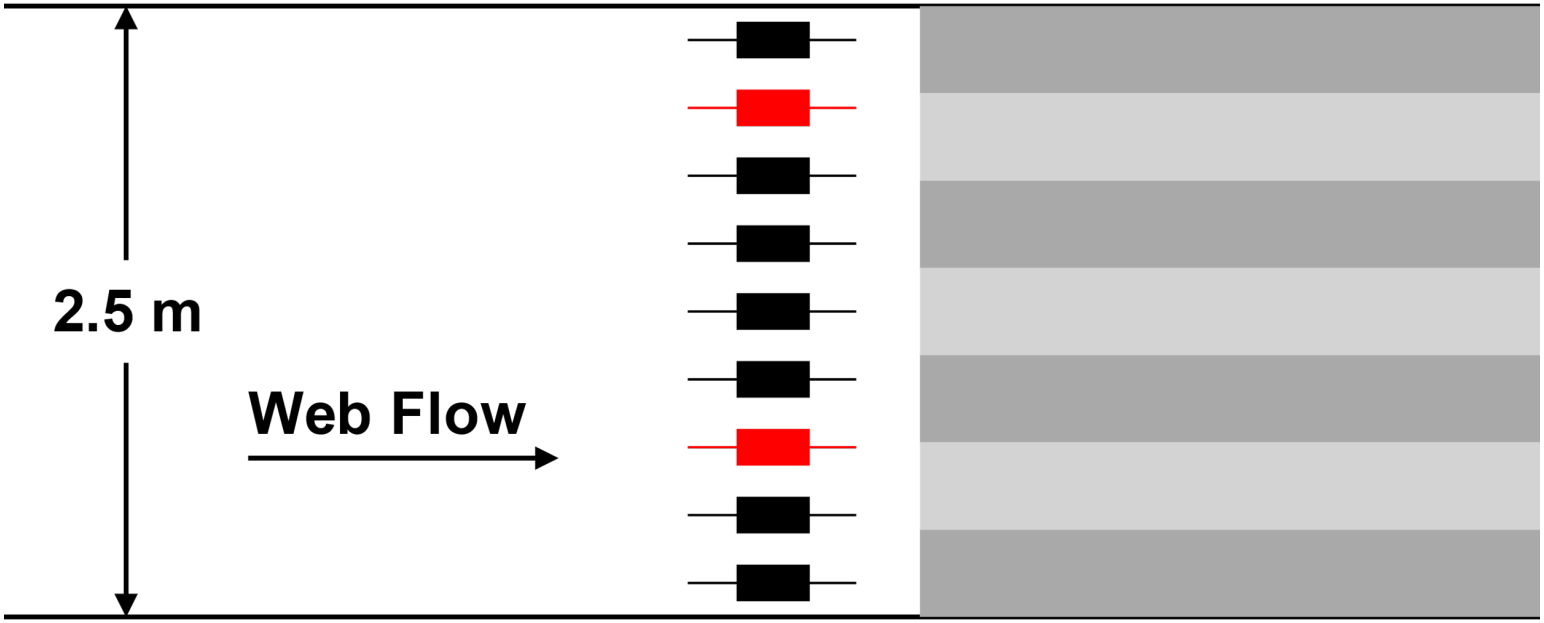
Slit Tools with Optimal Position Range

Slit Assignment Problem



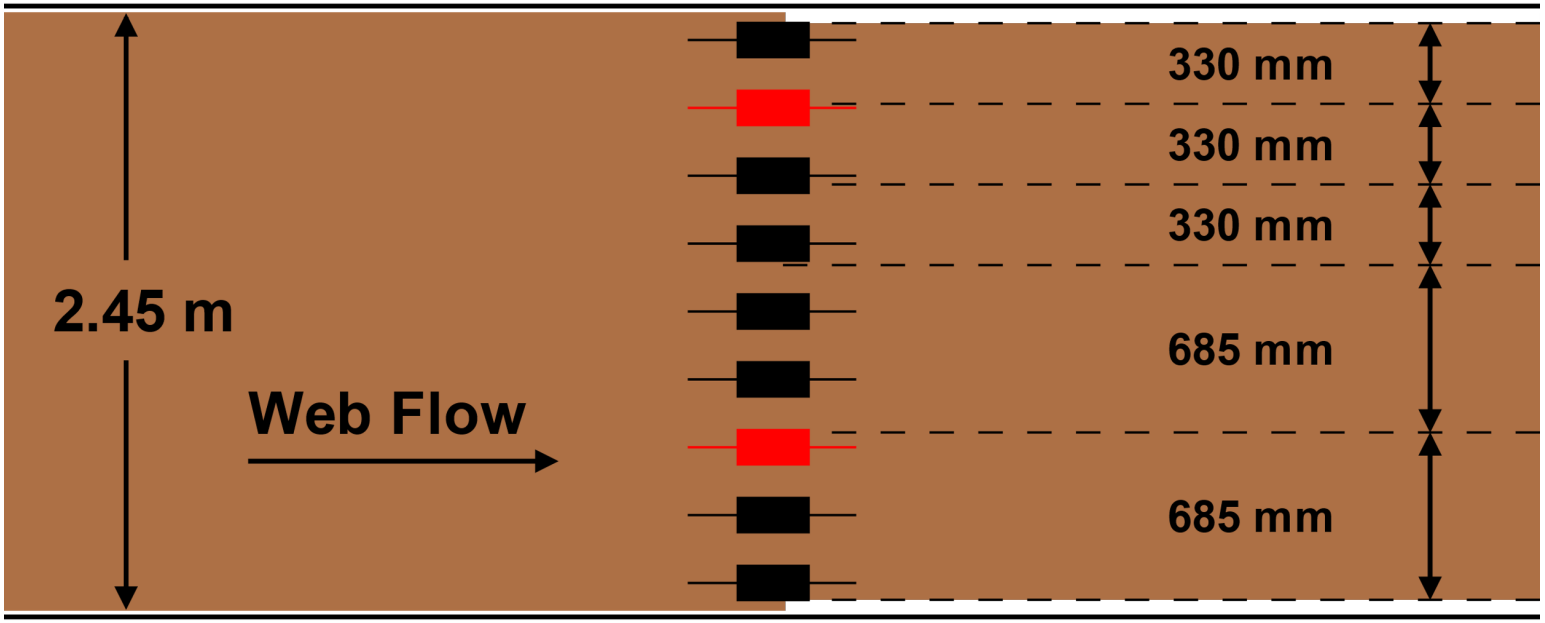
Web In **Web Strips Out**

Slit Assignment Problem



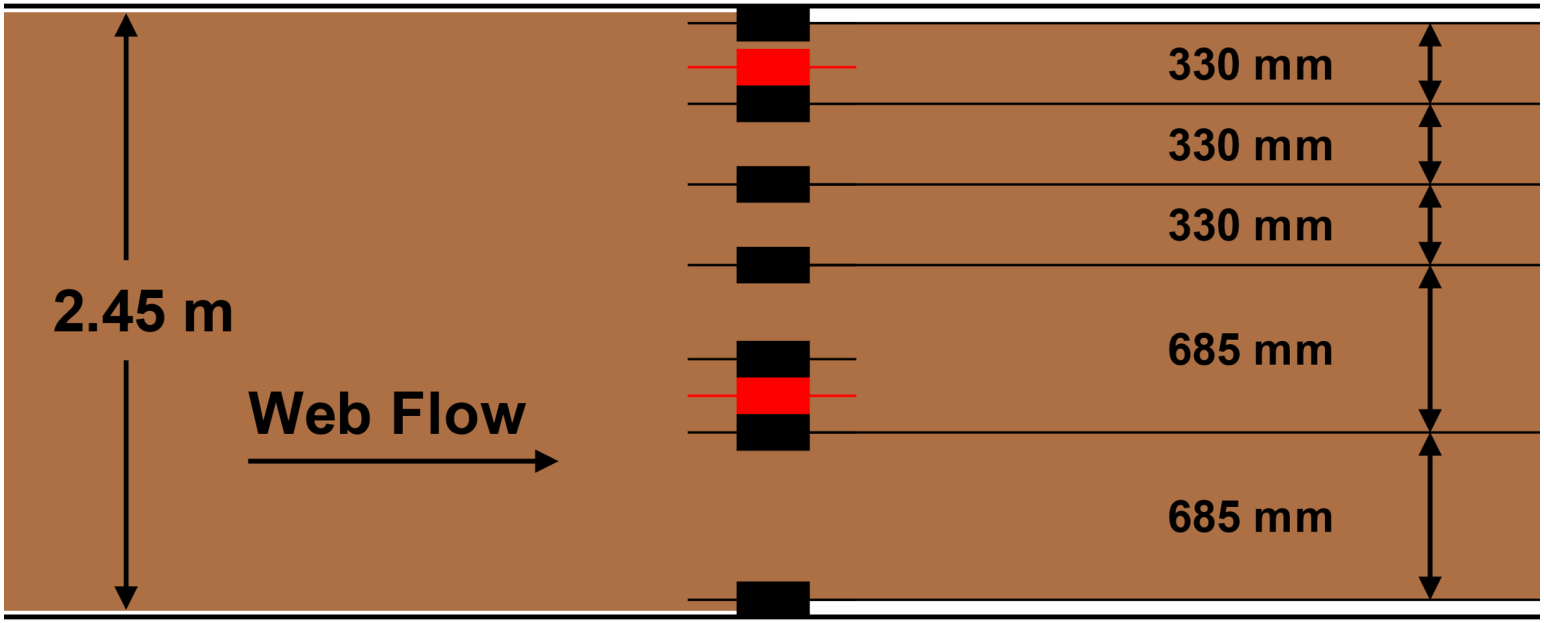
Disabled Slit Tools with New Optimal Position Range

Slit Assignment Problem



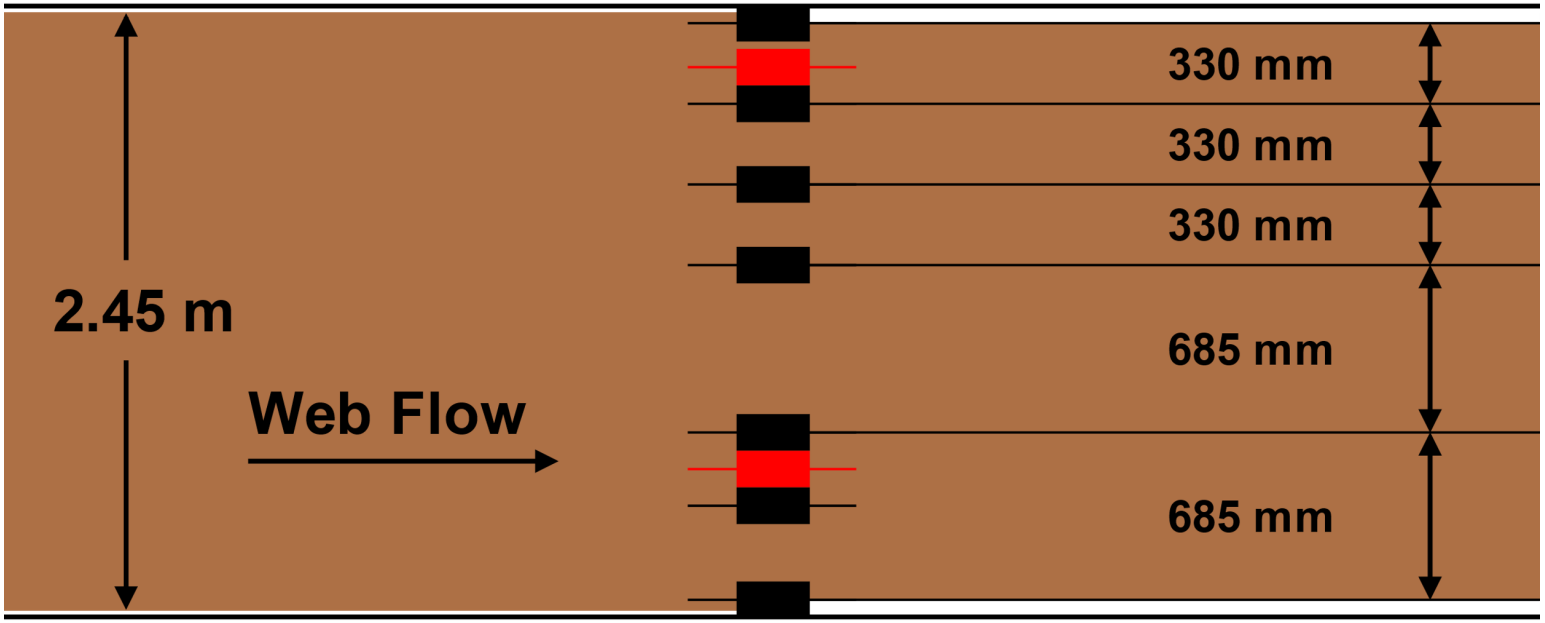
Slit Tools and Desired Machine Output

Slit Assignment Problem



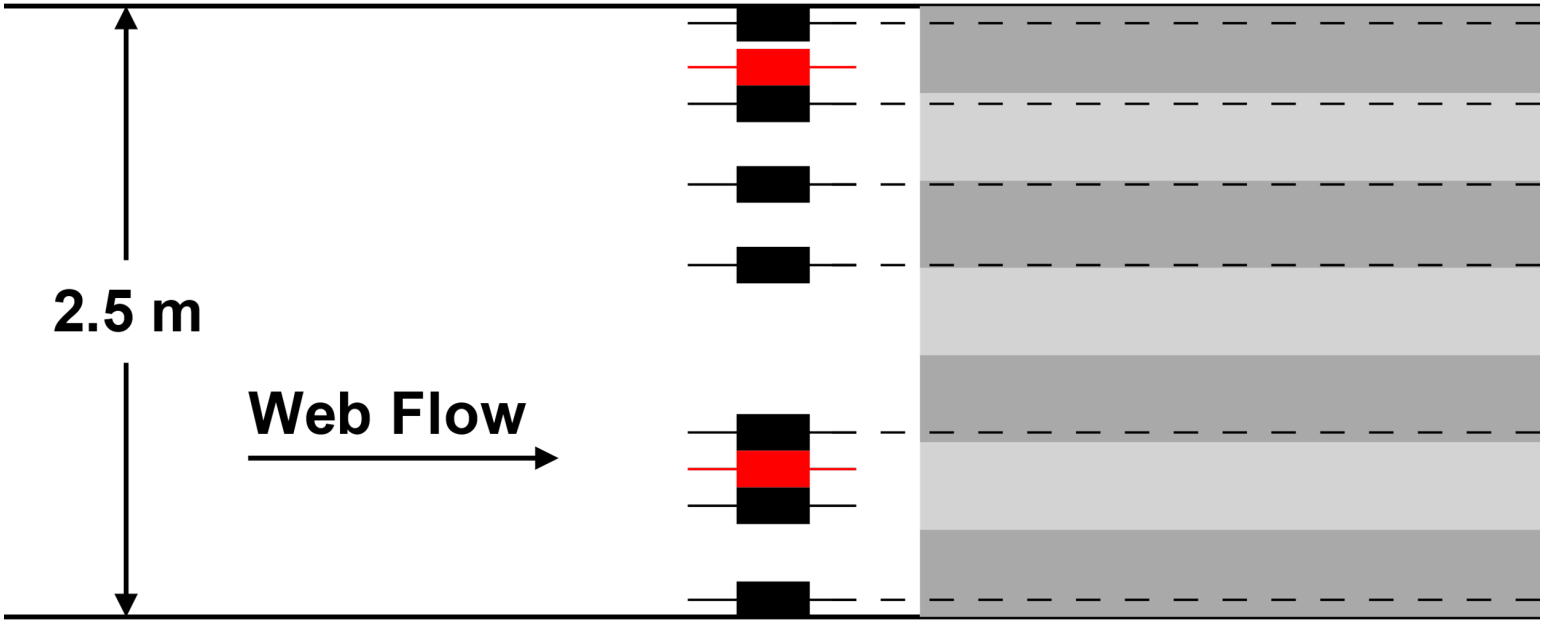
Assigned Slit Positions

Slit Assignment Problem



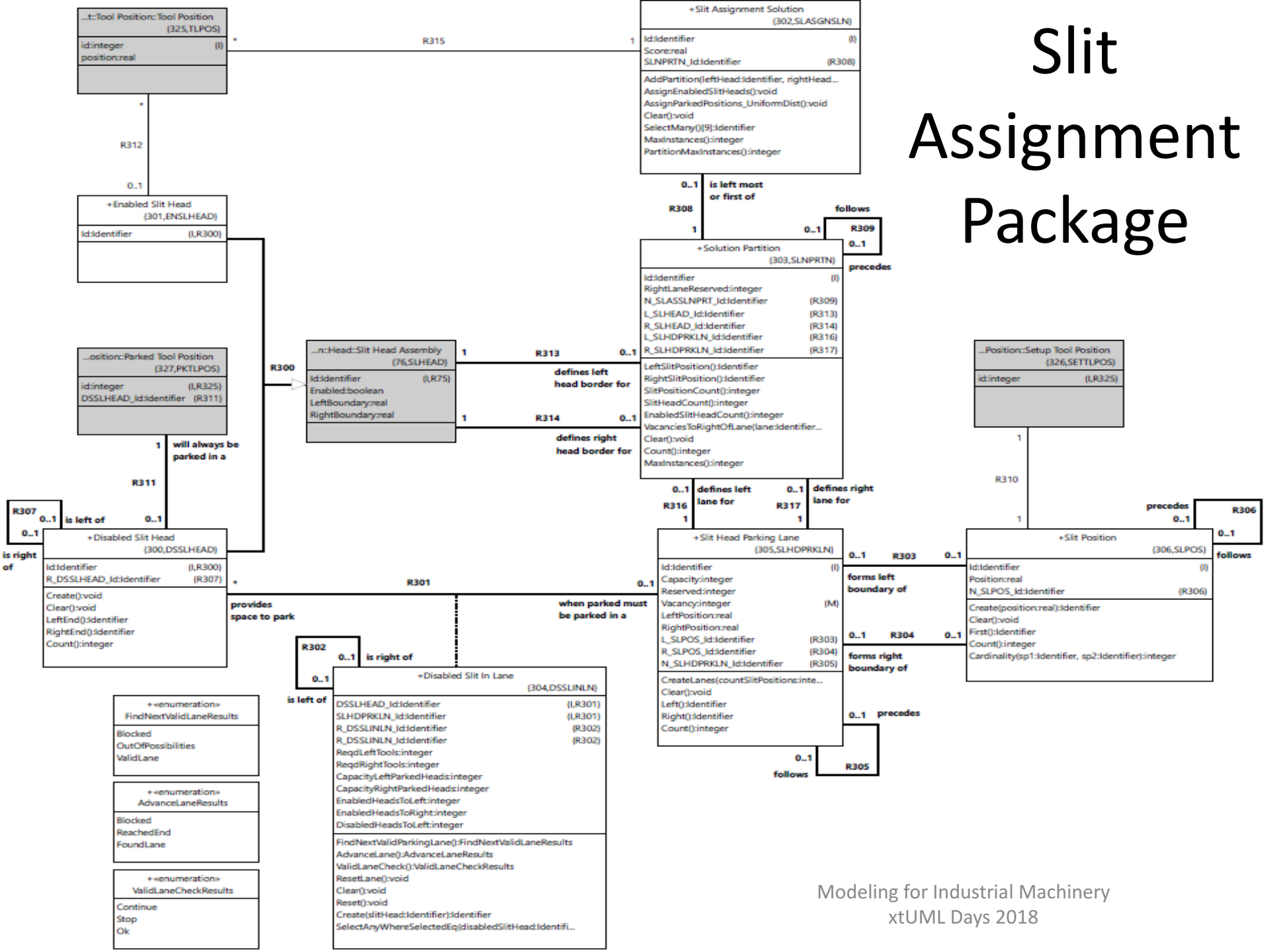
Assigned Slit Positions - 2

Slit Assignment Problem



Assigned Slit Positions - 2 with Optimal Position Range

Slit Assignment Package



Slit Assignment in Operation

- C# Application

PLC

- Programmable Logic Controller
- IEC 61131-3 Standard
  - Ladder Logic
  - Function Block
  - Sequential Function Chart
  - Instruction List
  - Structured Text (Pascal Syntax)
- All Vendors Implement – All Incompatible!!!

PLC

- XML Import / Export
  - Potential for Code Generation
- Restrictions Placed on Architecture
  - Dynamic Memory Allocation – No
  - Structures - Yes
  - Pointers / References – Maybe
  - Functions / Subroutines – Yes, but may be costly
  - Recursion – No
  - Limited length variable and type names

C# Architecture

- Disclaimer – I am intentionally not using the full capabilities of C#. Why?
- Portability of the model between C# & Structured Text.
- Aids testing by providing a visual representation of a complex algorithm.

C# Architecture

- Subject Matter

Model Construct	C# Construct
Class	Class
Class Attribute	Class Property
Class Operation	Class Method
Associative Relationships	Instance Reference
Reflexive Relationships	Dual Instance References

C# Architecture

- Instance Management
  - Static Variables, Properties, Methods in Class

Model Construct	C# Construct
Instance Storage	Fixed Length Array
Instance Creation	Dynamic using new
Instance Identifier	Debugging Tool
Instance Deletion	Abandoned for GC
Select-Any-Where	Method: Parameters define ‘Where’

C# SlitPosition Class

```
public class SlitPosition
{
    private SlitPosition(float position)
    {
        Position = position;
        NextPosition = null;
        PrevPosition = null;
    }

    public float Position { get; set; }

    // Relationship - R306
    public SlitPosition NextPosition { get; set; }
    public SlitPosition PrevPosition { get; set; }
```

C# SlitPosition Class

```
    // Instance Management
    private static SlitPosition[] instances = new SlitPosition[9];
    private static int nextInstance = 0;

    public static SlitPosition Create(float position) {...}

    public static void Clear() {...}

    public static SlitPosition First {...}

    public static SlitPosition Last {...}

    public static int Count {...}

    public static int Cardinality(SlitPosition item1, SlitPosition item2) {...}
}
```

Structured Text Architecture #1

- PLC Implements OOP Extensions
- Organization similar to BridgePoint
  - Folder for Class, Package & Component

Structured Text Architecture #1

- Subject Matter

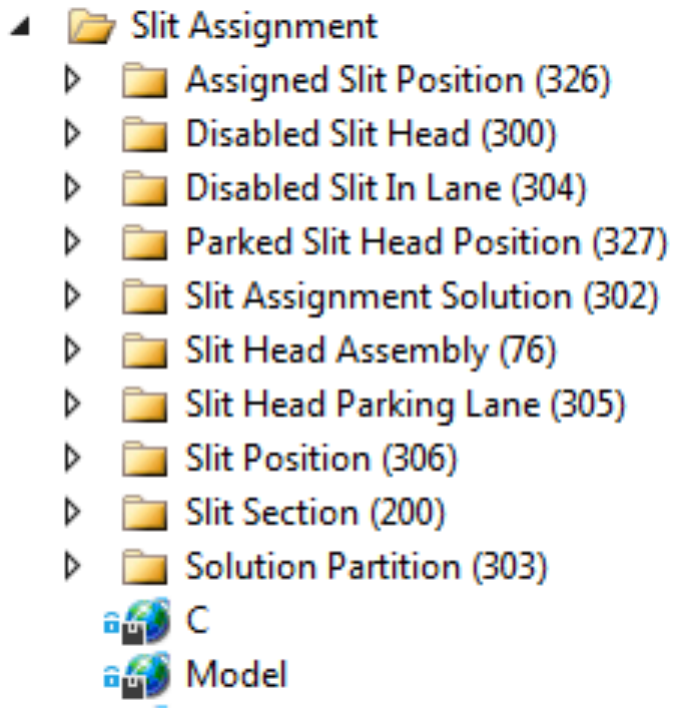
Model Construct	Structured Text Construct
Class	Overloaded Function Block (FB)
Class Attribute	FB Property
Class Operation	FB Method
Associative Relationships	Instance Reference
Reflexive Relationships	Class Mgr Method Call (param: Id)

Structured Text Architecture #1

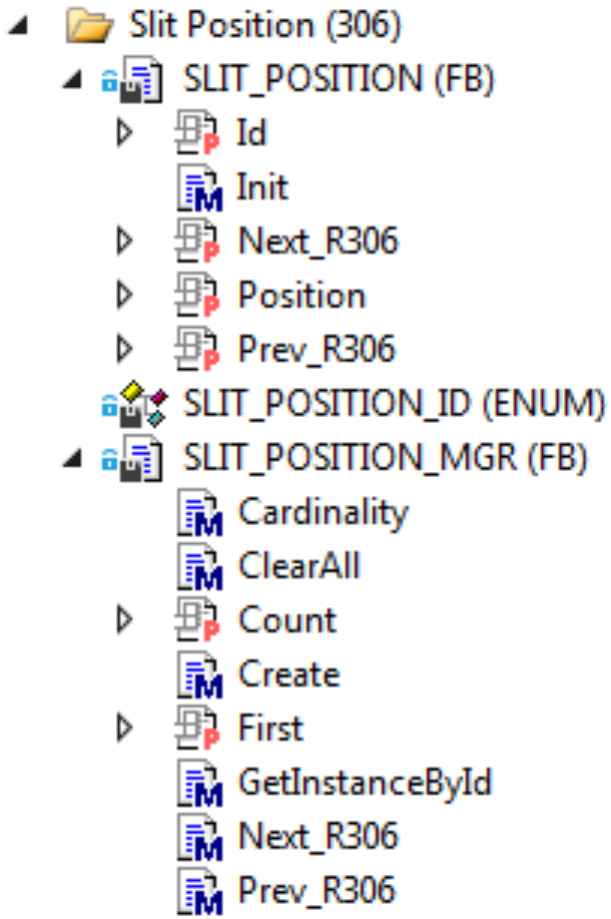
- Instance Management
  - Variables, Properties, Methods in Manager Class

Model Construct	Structured Text Construct
Instance Storage	Fixed Length Array
Instance Creation	Set Instance Id = Enum Id
Instance Identifier	Enumeration
Instance Deletion	Set Instance Id = Enum Null Id
Select-Any-Where	Method: Parameters define ‘Where’

ST-1 Slit Assignment Package



ST-1 Slit Position Class



ST-1 Slit Position Class

```
1   FUNCTION_BLOCK SLIT_POSITION
2   VAR_INPUT
3       _idxId : SLIT_POSITION_ID;   (* Architecture Identifier *)
4   END_VAR
5   VAR_OUTPUT
6   END_VAR
7   VAR
8       _position : REAL;
9       _next_r306 : SLIT_POSITION_ID;
10      _prev_r306 : SLIT_POSITION_ID;
11  END_VAR
```

ST-1 Slit Position Identifiers

```
1   {attribute 'qualified_only'}
2   TYPE SLIT_POSITION_ID :
3   (
4       Null_SlitPosition := 0,
5       SlitPosition_1,
6       SlitPosition_2,
7       SlitPosition_3,
8       SlitPosition_4,
9       SlitPosition_5,
10      SlitPosition_6,
11      SlitPosition_7,
12      SlitPosition_8,
13      SlitPosition_9
14  );
15  END_TYPE
16
```

ST-1 Slit Position Mgr Class

```
1   FUNCTION_BLOCK SLIT_POSITION_MGR
2   VAR_INPUT
3   END_VAR
4   VAR_OUTPUT
5   END_VAR
6   VAR   (* Architecture *)
7       _instances : ARRAY [1..C.maxSLIT_POSITION] OF SLIT_POSITION;
8       _nextInstance : INT;
9
10      _first : REFERENCE TO SLIT_POSITION;
11  END_VAR
12
```

```
1   ClearAll();
```

ST-1 Slit Position Method

```
METHOD GetInstanceById : REFERENCE TO SLIT_POSITION
VAR_INPUT
    id : SLIT_POSITION_ID;
END_VAR
VAR
    inst : REFERENCE TO SLIT_POSITION REF= 0;
END_VAR
```

```
IF (_nextInstance > 1) THEN
    IF (id >= 1 AND id < _nextInstance) THEN
        inst REF= _instances[id];
    END_IF
END_IF

GetInstanceById REF= inst;
```

ST-1 Model

```
1  {attribute 'qualified_only'}
2  VAR_GLOBAL
3      assigned_slit_position_mgr    : ASSIGNED_SLIT_POSITION_MGR;
4      disabled_slit_head_mgr        : DISABLED_SLIT_HEAD_MGR;
5      disabled_slit_in_lane_mgr     : DISABLED_SLIT_IN_LANE_MGR;
6      parked_slit_head_position_mgr : PARKED_SLIT_HEAD_POSITION_MGR;
7      slit_assignment_solution_mgr  : SLIT_ASSIGNMENT_SOLUTION_MGR;
8      slit_head_assembly_mgr        : SLIT_HEAD_ASSEMBLY_MGR;
9      slit_head_parking_lane_mgr    : SLIT_HEAD_PARKING_LANE_MGR;
10     slit_position_mgr             : SLIT_POSITION_MGR;
11     slit_section                  : SLIT_SECTION;
12     solution_partition_mgr        : SOLUTION_PARTITION_MGR;
13     setup_mgr                     : SETUP_MGR;
14 END_VAR
15
```

Structured Text Architecture #2

- Same PLC as Arch #1 but w/o OOP extensions
- Organization similar to BridgePoint
  - Folder for Class, Package & Component
- Some name mangling required

Structured Text Architecture #2

- Subject Matter

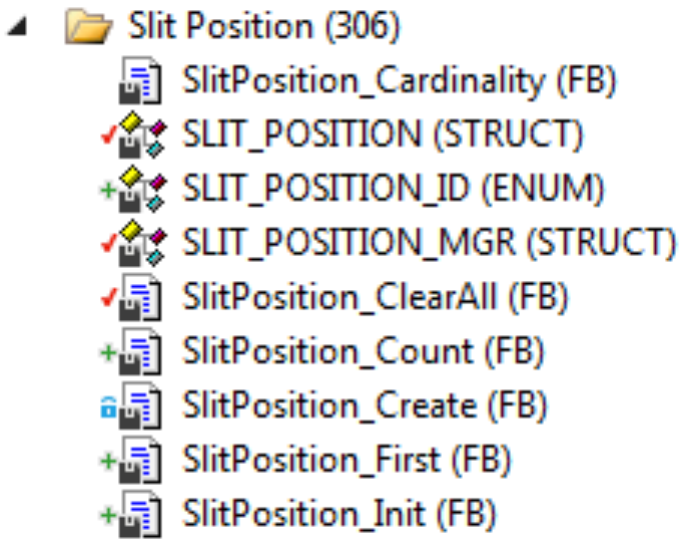
Model Construct	Structured Text Construct
Class	Structure & Functions
Class Attribute	Structure Member
Class Operation	Function
Associative Relationships	Referred To Object Id
Reflexive Relationships	Referred To Object Id

Structured Text Architecture #2

- Instance Management
  - Instance Id is set to array index

Model Construct	Structured Text Construct
Instance Storage	Fixed Length Array
Instance Creation	Set Instance Id = Enum Id
Instance Identifier	Enumeration
Instance Deletion	Set Instance Id = Enum Null Id
Select-Any-Where	Inline or Function

ST-2 Slit Position



ST-2 Slit Position Structure

```
1  TYPE SLIT_POSITION :
2  STRUCT
3      Id         : SLIT_POSITION_ID;
4      Position   : REAL;
5      Next_R306  : DINT;
6      Prev_R306  : DINT;
7  END_STRUCT
8  END_TYPE
9
```

ST-2 Slit Position Mgr Structure

```
1  TYPE SLIT_POSITION_MGR :
2  STRUCT
3      NextInstance : DINT;
4      First : DINT;
5  END_STRUCT
6  END_TYPE
7
```

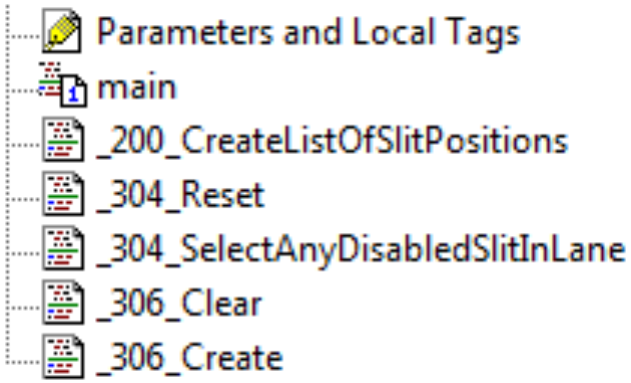
ST-2 Model

```
1   {attribute 'qualified_only'}
2   VAR_GLOBAL
3       SlitPosition : ARRAY[1..9] OF SLIT_POSITION;
4       DisabledSlitHead : ARRAY[1..7] OF DISABLED_SLIT_HEAD;
5       DisabledSlitInLane : ARRAY[1..45] OF DISABLED_SLIT_IN_LANE;
6
7       SlitPositionMgr : SLIT_POSITION_MGR;
8       DisabledSlitHeadCls : DISABLED_SLIT_HEAD_CLS;
9       DisabledSlitInLaneCls : DISABLED_SLIT_IN_LANE_CLS;
10  END_VAR
11
```

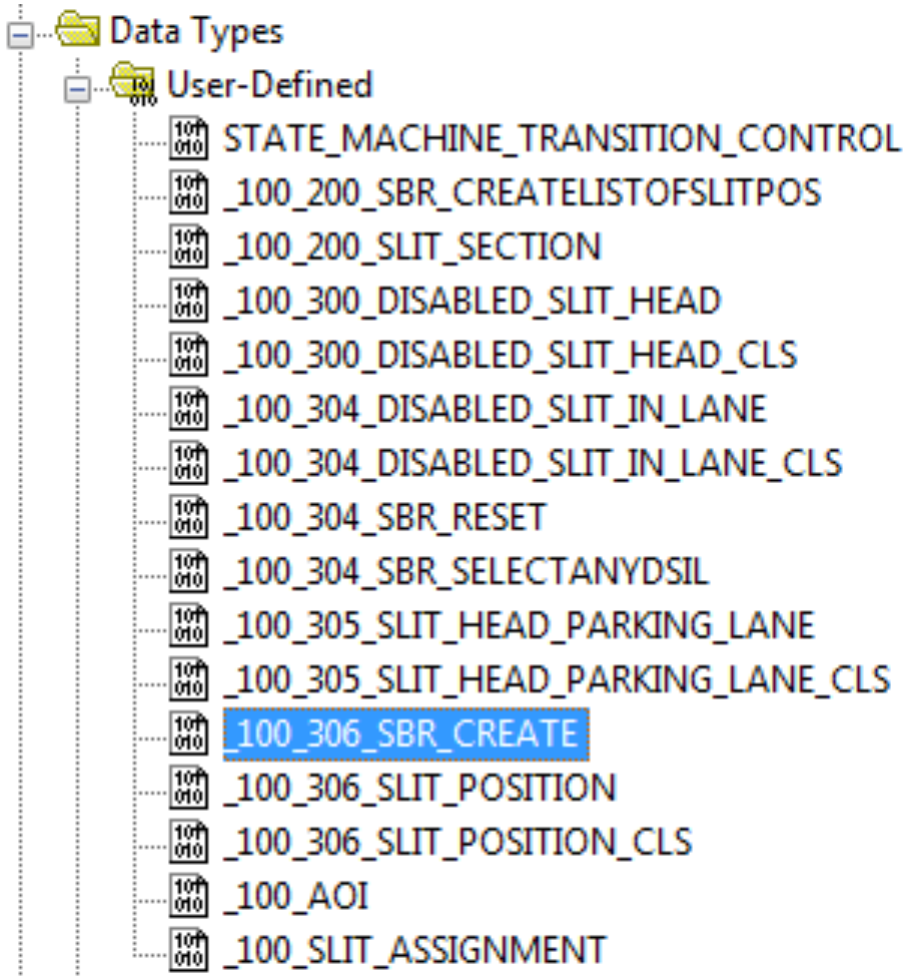
Structured Text Architecture #3

- Another vendor's PLC
- No Organization
  - Everything is Global!
    - Model
    - Subroutine Parameters
    - Local Variables
- Name mangling is required
- Organized code is difficult to read

ST-3 Subroutine List



ST-3 User Defined Data Type List



ST-3 Parameters & Local Tags

Name	Data Type
+ _200_CreateListOfSlitPos	_100_200_SBR_CREATELISTOFSLITPOS
+ _200_index	DINT
+ _200_last_slit_pos_id	DINT
+ _200_slit_pos_id	DINT
+ _304_inst	DINT
+ _304_Reset	_100_304_SBR_RESET
+ _304_SelectAnyDSIL	_100_304_SBR_SELECTANYDSIL
+ _306_Create	_100_306_SBR_CREATE
− model	_100_SLIT_ASSIGNMENT
+ model.slit_section	_100_200_SLIT_SECTION
+ model.disabled_slit_head	_100_300_DISABLED_SLIT_HEAD[7]
+ model.disabled_slit_head_cls	_100_300_DISABLED_SLIT_HEAD_CLS
+ model.disabled_slit_in_lane	_100_304_DISABLED_SLIT_IN_LANE[7]
+ model.disabled_slit_in_lane_cls	_100_304_DISABLED_SLIT_IN_LANE_CLS
+ model.slit_head_parking_lane	_100_305_SLIT_HEAD_PARKING_LANE[10]
+ model.slit_head_parking_lane_cls	_100_305_SLIT_HEAD_PARKING_LANE_CLS
+ model.slit_position	_100_306_SLIT_POSITION[9]
+ model.slit_position_cls	_100_306_SLIT_POSITION_CLS

ST-3 Example Subroutine

```
SBR(_304_SelectAnyDSIL.disabledSlitHeadId);

For _304_SelectAnyDSIL.inst := 0 to model.disabled_slit_in_lane_cls.d_next_instance by 1 do
    if (model.disabled_slit_in_lane[_304_SelectAnyDSIL.inst].d_R301_disabled_slit_head_id =
        _304_SelectAnyDSIL.disabledSlitHeadId) then

        RET (_304_SelectAnyDSIL.inst);
    end_if;
end_for;

RET (-1);
```

ST-3 Example Subroutine

```
_200_CreateListOfSlitPos.last_slit_pos_id := -1;
_200_CreateListOfSlitPos.slit_pos_id := -1;

for _200_CreateListOfSlitPos.index := 0 to 9 by 1 do
    JSR(_306_Create, 1, 10.0, _200_CreateListOfSlitPos.slit_pos_id);

    if (_200_CreateListOfSlitPos.index > 0) then
        model.slit_position[_200_CreateListOfSlitPos.last_slit_pos_id].d_R306_next := _200_CreateListOfSlitPos.slit_pos_id;
        model.slit_position[_200_CreateListOfSlitPos.index].d_R306_prev := _200_CreateListOfSlitPos.last_slit_pos_id;
    end_if;

    _200_CreateListOfSlitPos.last_slit_pos_id := _200_CreateListOfSlitPos.slit_pos_id;
end_for;
```

Event Driven State Machine

- Events are represented by bits in an integer
- No event queue
- All SM are executed once per program scan
- While loop inside SM ensures all events are processed, including self generated
- Case statement for each state
- Supports ‘entry’ & ‘exit’ action
- Code generation

ST – Event Driven SM

```
while (not done) do
    done := true;
    case (current_state) of
    1:   (* State: 'Drive Disabled' *)
         (* Entry Action *)
         if (b_txn) then
              b_txn := b_FALSE;

              (*  Action:
                   select one cross_transfer_servo related by self->CRSXFRSVO[R1];
                   generate CRSXFRSVO5:servo_disabled() to cross_transfer_servo;
              *)

              (* TODO: Implement state action code here *)
          end_if;

         (* Exit Conditions *)
         if (b_enable_drive) then
```

ST – Event Driven SM

```
        (* Exit Action *)

        (* Consume the Event *)
        b_enable_drive := b_FALSE;

        d_current_state := 2;
        b_txn := b_TRUE;
        b_done := b_FALSE;

    elsif (b_disable_drive) then
        (* Can't Happen - The model indicates this event cannot occur in this state.
             Description:
        *)
        (* TODO: Add a diagnostic here. *)
    end_if;

2:  (* State: 'Initiate Enable' *)
```

State Machine Execution

- Assigner SM are processed first
- Instance SM
  - Make a copy of global event reception mailbox for all class instances
  - Process polymorphic events
  - Clear global event reception mailboxes
  - Execute the SMs

ST – SM Processing

```
(*** Assigner (class) State Machines ***)

(* _000_003_asmCrossTransferServo
  Qualified Name: 'Diverter Xtfr Conveyor::Design::Diverter Xtfr Conveyor::Drive::Cross
                  Transfer Servo‘
*)

if ((  _events.event_cross_transfer_servo_000_003_a.evt_lst.b_jog Or
       _events.event_cross_transfer_servo_000_003_a.evt_lst.b_event1 Or
       _events.event_cross_transfer_servo_000_003_a.evt_lst.b_Event2)) then

     (* Receive the events and execute the state machine. *)
     CPS(_events.event_cross_transfer_servo_000_003_a,
           _obj.cross_transfer_servo_000_003_a.events, 1);
     CPS(empty_event_struct, _events.event_cross_transfer_servo_000_003_a.evt_lst, 1);
     JSR(_000_003_asmCrossTransferServo);
end_if;
```

ST – SM Processing

```
(*** Instance State Machines ***)

(*** Receive Events - Move event data from global mailboxes into local
                                        event structures. ***)

(* _000_003_smCrossTransferServo *)
for inst := 0 to _CONSTANTS.OBJ_ARRAY_UPPER_BOUND.d_000_003_CRSXFRSVO by 1 do
      if (_obj.cross_transfer_servo_000_003[inst].epi_id <> -1) then
                CPS(_events.event_cross_transfer_servo_000_003[inst],
                      _obj.cross_transfer_servo_000_003.events, 1);
      end_if;
end_for;

(* _000_013_smAutomaticSequence *)
if (_obj.automatic_sequence_000_013.pi_id_r8 <> -1) then
      CPS(_events.event_automatic_sequence_000_013, _obj.automatic_sequence_000_013.events, 1);
end_if;
```

ST – SM Processing

```
(*** Receive Polymorphic Events ***)

(* Conveyor_Length_Rollout_000_015 *)
if (_obj.conveyor_length_rollout_000_015.pi_id_r9 <> -1) then
    _obj.conveyor_length_rollout_000_015.events.evt_lst.b_initiate_auto_seq :=
        _events.event_automatic_sequence_000_013.evt_lst.b_initiate_auto_seq;
    _obj.conveyor_length_rollout_000_015.events.initiate_auto_seq_event_data :=
        _events.event_automatic_sequence_000_013.initiate_auto_seq_event_data;
    _obj.conveyor_length_rollout_000_015.events.evt_lst.b_jog_complete :=
        _events.event_cross_transfer_conveyor_000_007.evt_lst.b_jog_complete;
    _obj.conveyor_length_rollout_000_015.events.evt_lst.b_motion_completed :=
        _events.event_cross_transfer_conveyor_000_007.evt_lst.b_motion_completed;
    _obj.conveyor_length_rollout_000_015.events.motion_completed_event_data :=
        _events.event_cross_transfer_conveyor_000_007.motion_completed_event_data;
end_if;
```

ST – SM Processing

```
(*** Clear Event Mailboxes - Clear event data in global mailbox. ***)

(* _000_003_smCrossTransferServo *)
for inst := 0 to _CONSTANTS.OBJ_ARRAY_UPPER_BOUND.d_000_003_CRSXFRSVO by 1 do
     if (_obj.cross_transfer_servo_000_003[inst].epi_id <> -1) then
          CPS(empty_event_struct,
               _events.event_cross_transfer_servo_000_003[inst].evt_lst, 1);
     end_if;
end_for;

(* _000_013_smAutomaticSequence *)
if (_obj.automatic_sequence_000_013.pi_id_r8 <> -1) then
     CPS(empty_event_struct, _events.event_automatic_sequence_000_013.evt_lst, 1);
end_if;
```

ST – SM Processing

```
(*** Execute State Machines ***)

for inst := 0 to _CONSTANTS.OBJ_ARRAY_UPPER_BOUND.d_000_003_CRSXFRSVO by 1 do
      if (_obj.cross_transfer_servo_000_003[inst].epi_id <> -1 Or
          _obj.cross_transfer_servo_000_003[inst].events.evt_lst.b_creation_event) then

          (* Only execute instance state machine if instance exists or creation event will create it. *)

          if ((_obj.cross_transfer_servo_000_003[inst].events.evt_lst.b_creation_event Or
                _obj.cross_transfer_servo_000_003[inst].events.evt_lst.b_initiate_jog Or
                _obj.cross_transfer_servo_000_003[inst].events.evt_lst.b_servo_enabled Or
                _obj.cross_transfer_servo_000_003[inst].events.evt_lst.b_jogging Or
                _obj.cross_transfer_servo_000_003[inst].events.evt_lst.b_jog_stopped Or
                _obj.cross_transfer_servo_000_003[inst].events.evt_lst.b_servo_disabled Or
                _obj.cross_transfer_servo_000_003[inst].events.evt_lst.b_stop_jog) Or
               (_obj.cross_transfer_servo_000_003[inst].events.evt_lst.b_ev2)) then

               JSR(_000_003_smCrossTransferServo, 1, inst);
          end_if;
      end_if;
end_for;
```

Challenges of PLC Translation

- Inconsistencies among vendors
- Platform limitations – forced organization
- Language limitations – ptrs/ref
- Lack of encapsulation – all global

What is next for the project?

- Model and implement look ahead
- Add robot & machine sequencer simulators
- Create the robot instructions
- Create user interface to configure machine

Summary

- Corrugator, Slitter / Scorer
- Model of robotic slitter /scorer
- Slit Assignment problem
- Translation
  - C#
  - 3 Structured Text variants
- State machine translation

Modeling for Industrial Machinery

Questions ?

Dennis Tubbs

Intelligent Control Solutions, Inc.

dennis.tubbs@gmail.com