BridgePoint Roadmap



May 2017

BridgePoint Roadmap - Background

- Need for excellent xtUML tools…forever
- Balancing
  - Needs of existing projects
  - Community growth
  - Technology convergence
- OneFact Vision
  - Balancing demands
  - Leveraging technological insight
  - Providing a starting point
  - Seeking input
  - Driven by contracts



BridgePoint Roadmap (via Redmine)

- 5.8
  - Model Element Move
  - Imported classes via inter-project reference
  - Various defect repairs, including combined ref. issue
- 6.2
  - Raven Advanced
    - Syntax highlighting (MASL)
    - Completion assistance (MASL)
    - Automatic reference maintenance (MASL)
    - User-defined identifier validation (MASL)
    - OAL-to-MASL conversion
  - Package references, Phase 1
  - Various defect repairs



Suggested BridgePoint Roadmap Additions

- OAL completion assistance
- OAL automatic reference maintenance
- Activity signature editor
- Multi-body activity editor
- Advanced type system (Ada, Darwen and Date)
- Package references, Phase 2
- Persistence layer update
- Proxy removal
- Compare and merge
- Verifier for MASL
- Inspector integration
- Marking solution
- Pre-existing instance solution
- Migration to Papyrus platform



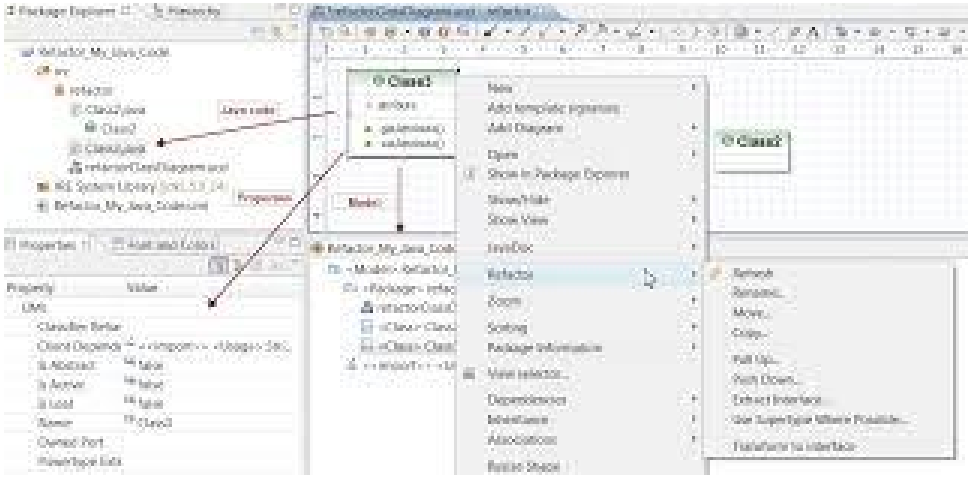
OAL Completion Assistance

- Additional intelligence in editor
- Provides completion assistance for items such as:
  - Keywords
  - Identifiers
  - Event names
  - Association phrases
- Saab #7551, position 9,
  - intersects with Raven



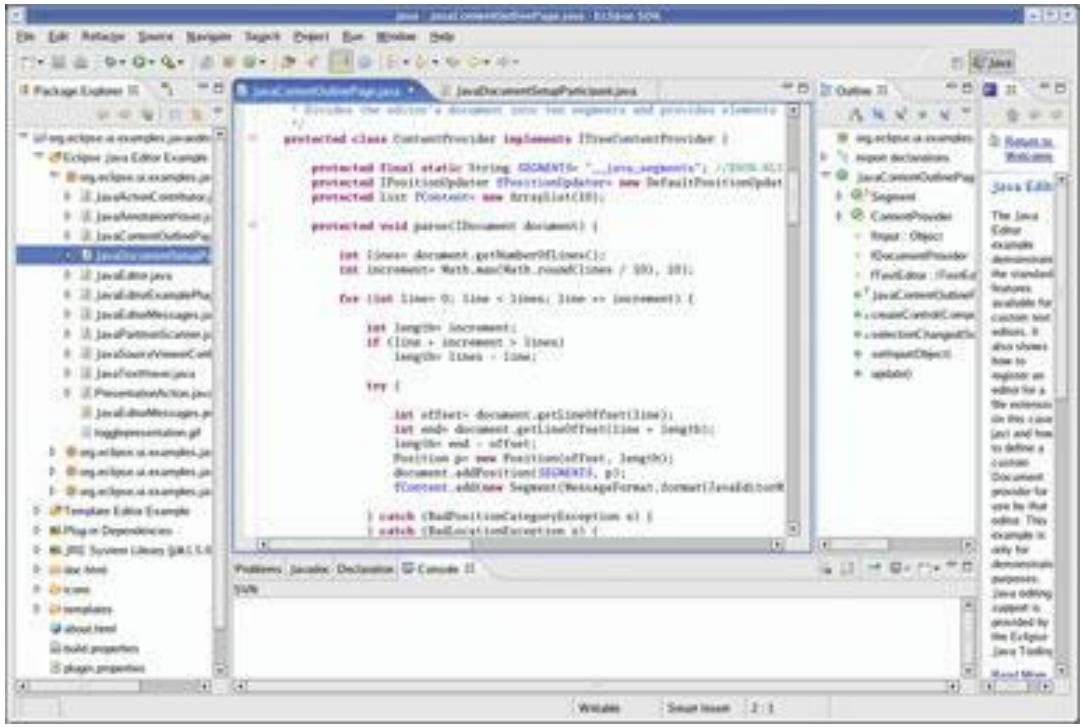
OAL Automatic Reference Maintenance

- Additional intelligence in editor
- Automatically updates OAL references to identifiers
- Normalizes model, maintaining one fact in one place
- Intersects with Raven



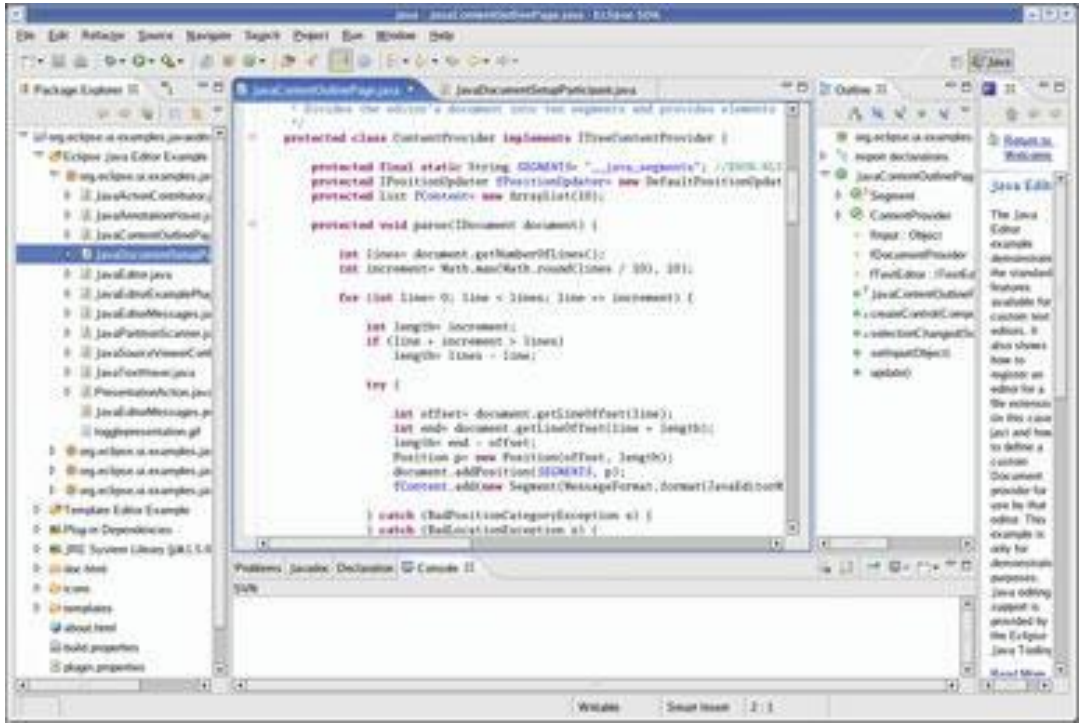
Activity Signature Editor

- Current approach requires many menu clicks
- Add ability to edit activity signatures within a file



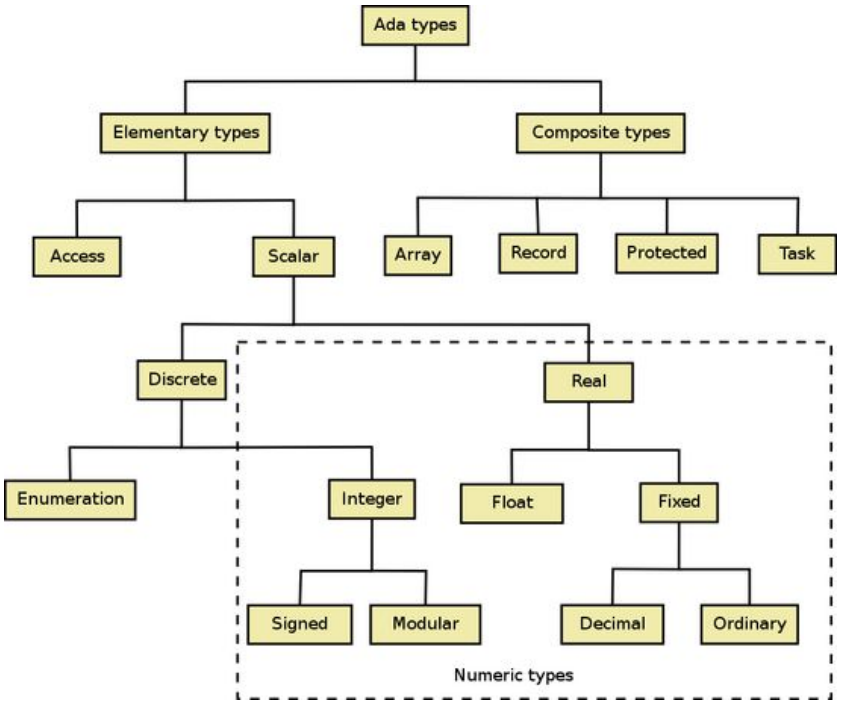
Multi-body Activity Editor

- Edit multiple activities within a single editor session
- Eases refactoring and restructuring of activities
- Increases mainstream appeal



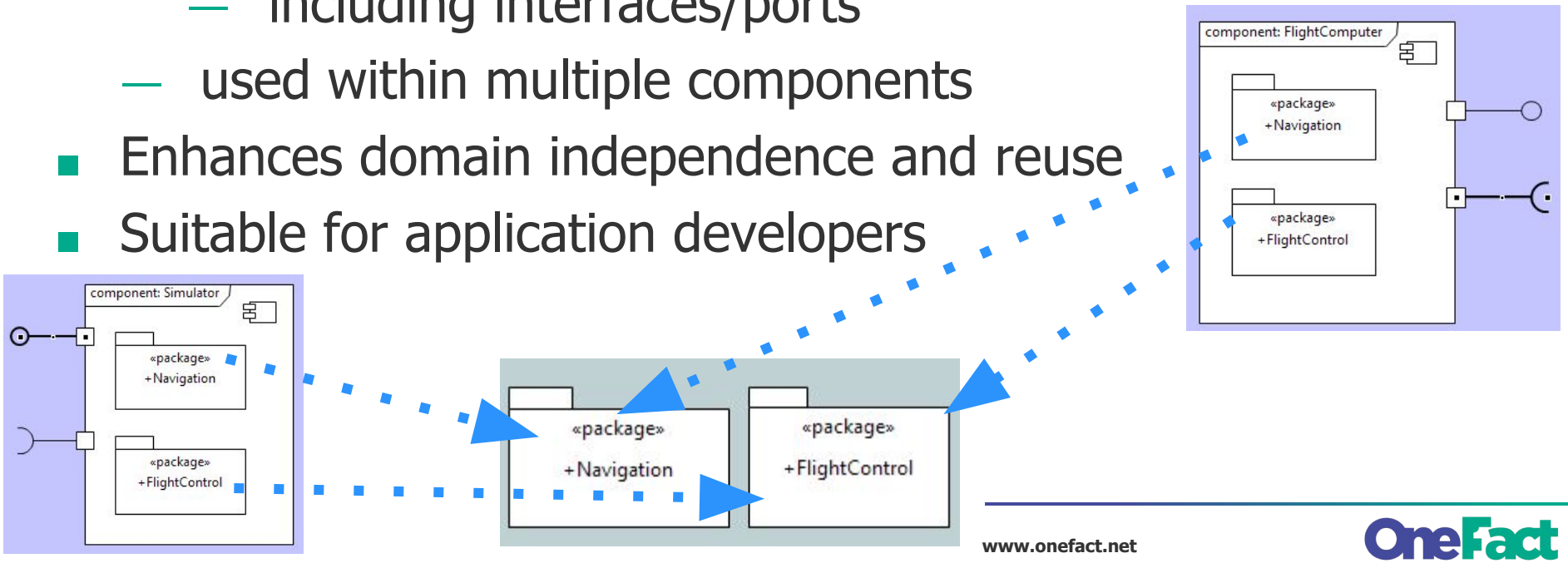
Advanced Type System

- Based on:
  - Ada
  - Darwen and Date
- Saab #7549, position 10,
  - intersects with Raven



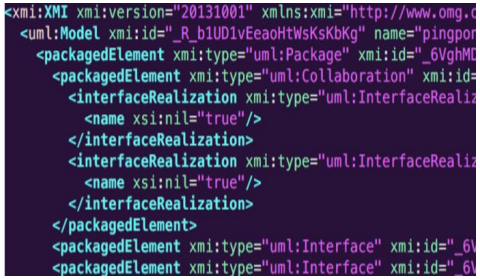
Package References, Phase 2

- BP 5.8, Package References Phase 1:
  - Reuse passive class models (e.g., meta-model)
  - Mainly for model compilers and tooling
- Phase 2 allows entire domains to be:
  - defined within packages, outside components
    - including interfaces/ports
  - used within multiple components
- Enhances domain independence and reuse
- Suitable for application developers



Persistence Layer Update

- BP currently stores models as:
  - meta-model instances, rendered in SQL
- EMF persistence would store models as:
  - meta-model instances, rendered in XML
- Model persisted as meta-model instances:
  - are large, hard to read and repair
  - significantly complicate compare/merge
- Models persisted in a modeling language
  - are more concise, easily read and repaired
  - simplify compare/merge




```
object Group is
  group_id   : preferred unique integer;
  group_name : string;

end object;
pragma id (3);
pragma key_letter ("G");

object Group_For_Session is
  group_id   : preferred referential (R2.is_member_of.
  session_id : preferred referential (R2.current_membe

end object;
```

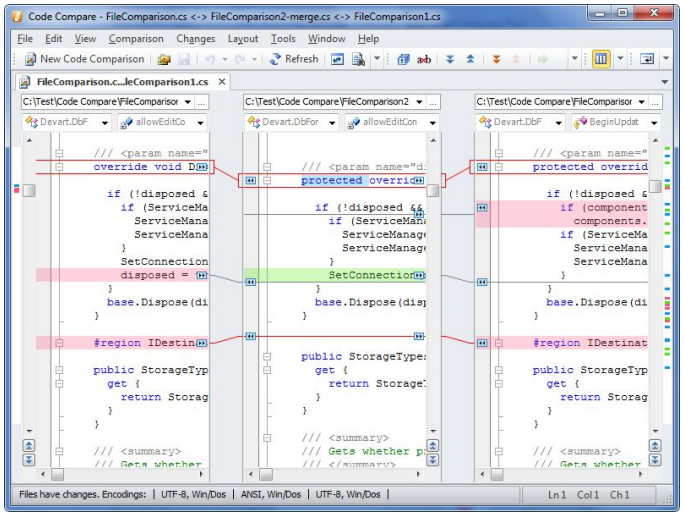
Proxy Removal

- Current BP architecture relies on proxies between:
  - referring elements and referred-to elements
  - example: classes carry proxies for data types
- Introduced years ago to reduce model load times
  - no longer necessary
  - complicates tool enhancement
  - examples: model element move, compare/merge
- Removal requires significant work
  - perhaps best done after persistence layer is updated



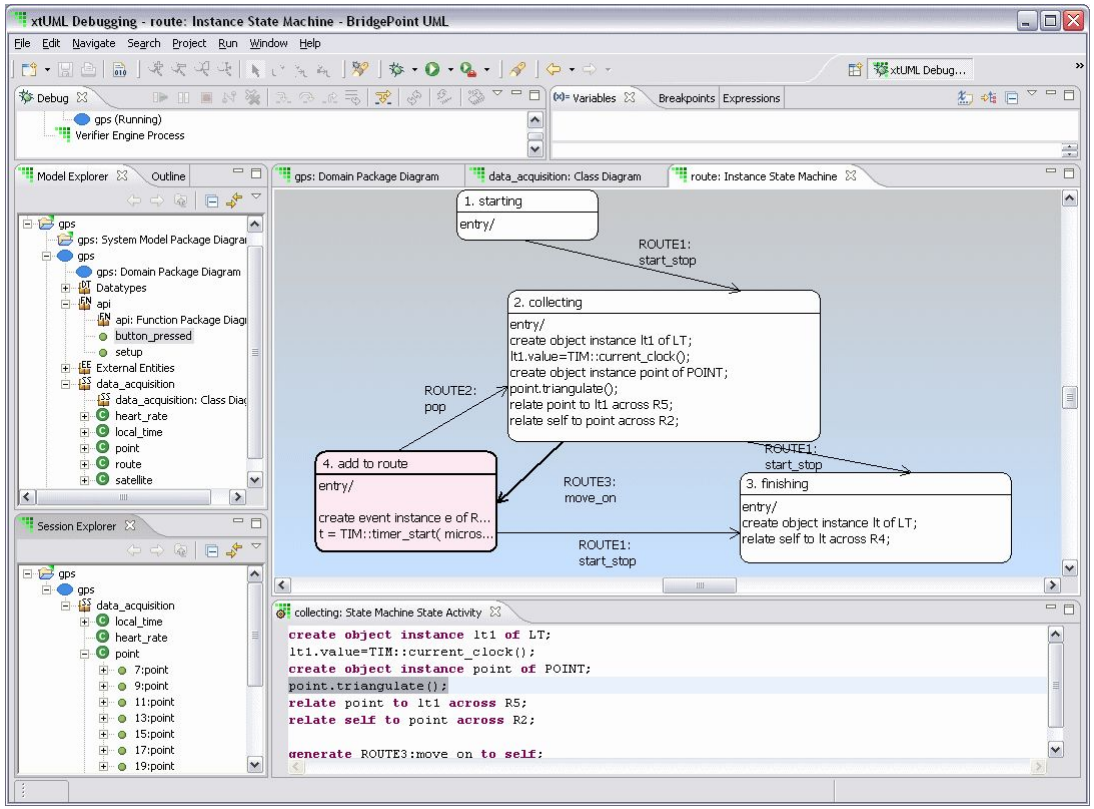
Compare and Merge

- Current BP support works in many circumstances
  - requires disciplined, process-focused use
  - failures can be difficult to untangle
- Robust support is required to
  - support existing users
  - appeal to mainstream users
- Perhaps best tackled after persistence layer is updated



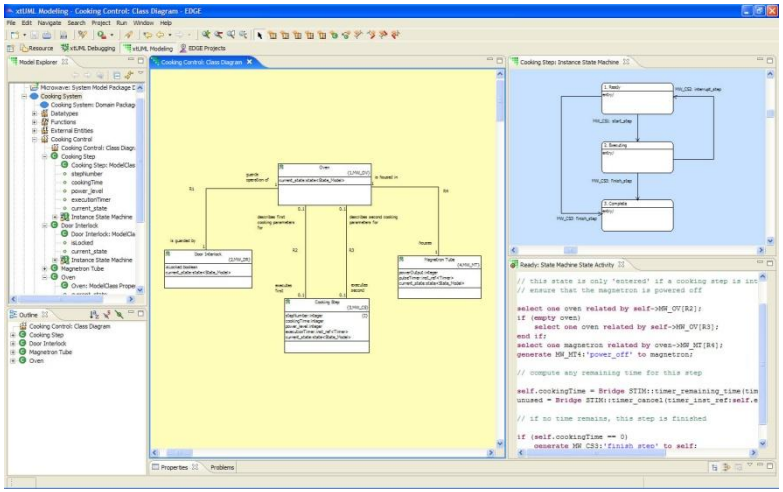
Verifier for MASL

- Verifier provides interpretive execution of models
  - presently supports only OAL-infused xtUML models



Inspector Integration

- Provides interactive debugging of generated code
  - presently supports only MASL models
  - stand-alone tool, not integrated with Eclipse or BP
- Integration could include
  - support for OAL and MASL
  - integration with Eclipse and BridgePoint
  - support for multiple model compilers
  - integration with Verifier (hybrid execution)



Marking Solution

- BP presently provides no support
- Some model compilers rely on text files
  - maintained separately from models ☺
  - difficult to correlate with models being marked ☹
- Some solutions intermix marking and model data ☹
- Requirements
  - maintain marking and model data separately
  - easily view, edit marking data within model context
  - support multiple model compilers





Pre-existing Instance Solution

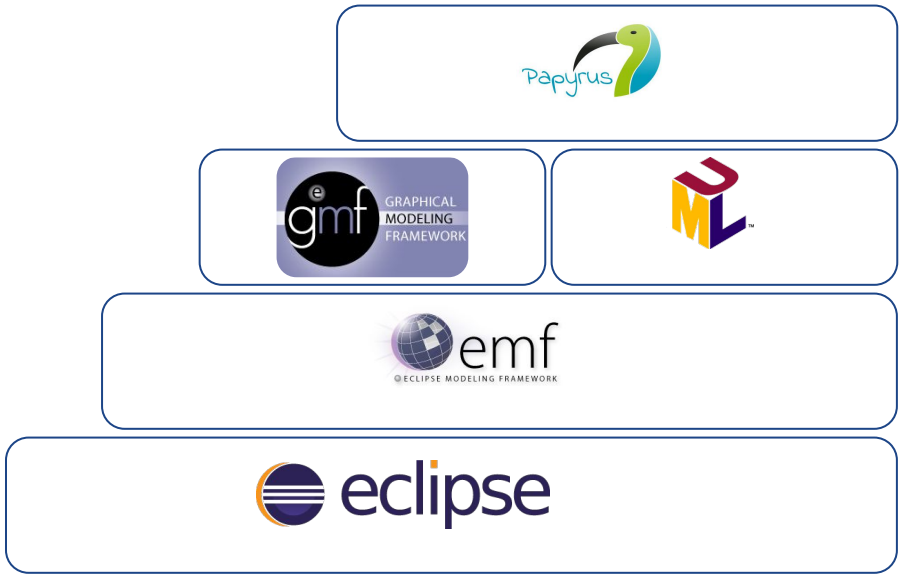
- BP presently provides no support
- Action-language based idiom
  - single definition for Verifier and generated code ☺
  - current tool support has…rough edges ☹
- Other approaches considered:
  - tabular
  - network
- Requirements?



Migration to Papyrus Platform

- Adopt mainstream OSS packages
  - focus on adding capabilities, not duplicating them
- Leverage growing community of developers and users
- Benefit from Papyrus Industry Consortium marketing







onefact.net