

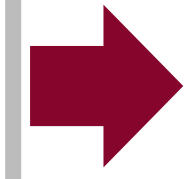
A Build Server for Model-Driven Engineering

Henrik Steudel, Regina Hebig, and Holger Giese

Hasso Plattner Institute at the University of Potsdam, Germany
{forename.surname}@hpi.uni-potsdam.de

MDE in Practice

- Multiple models complement each other, are merged, transformed, and manipulated in diverse orders
- Manual and automated activities are *intermixed*
- Manual activities are *expensive* and require *much time*

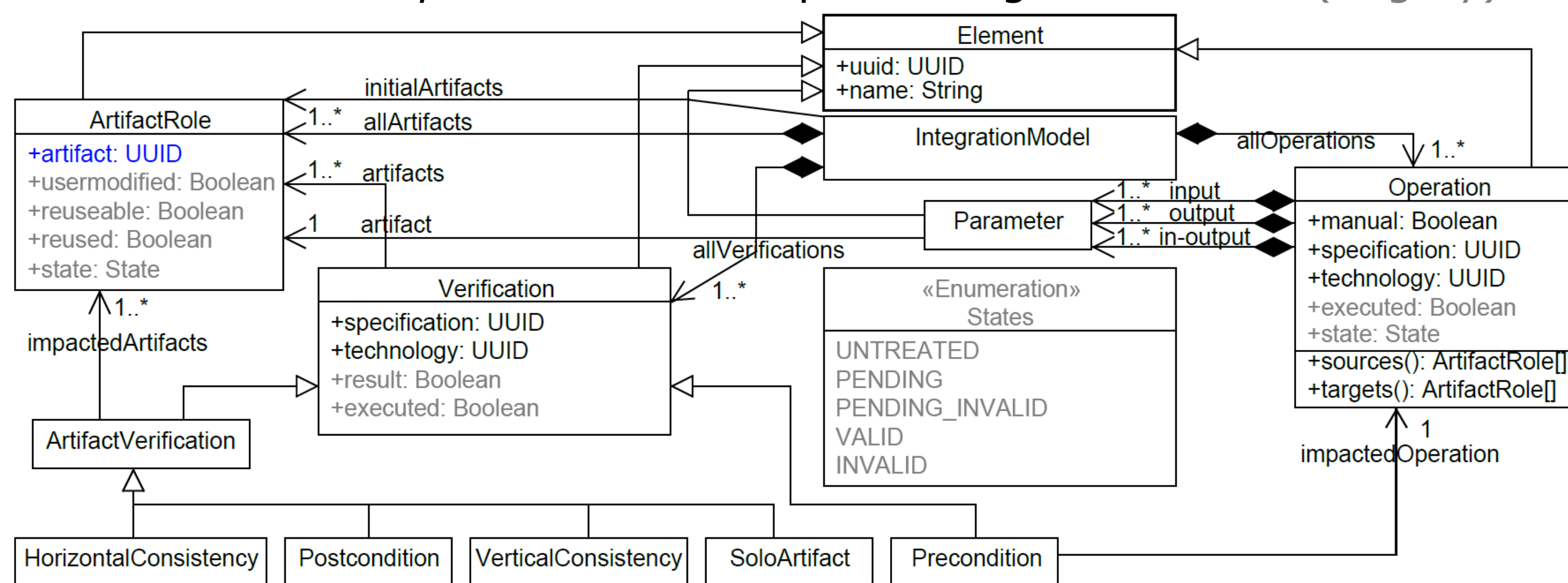


Challenges for an MDE Build Server

- Handle Manual Operations
 - Deal with missing artifacts
 - Provide all built artifacts to developers (as input for manual operations)
 - Notify developer when manual activities have to be performed (only if necessary)
- Integrated Validation and Verification (V&V) Operations:
 - Prevent usage of already erroneous artifacts in expensive manual activities
- Lightweight Build Script
 - Allow maximum freedom (spontaneous iterations and execution sequence of manual activities)
 - With minimum specification effort
- Partial build as far as possible with the given and successfully verified artifacts
 - Goal: provide all valid intermediate products for further manual treatment

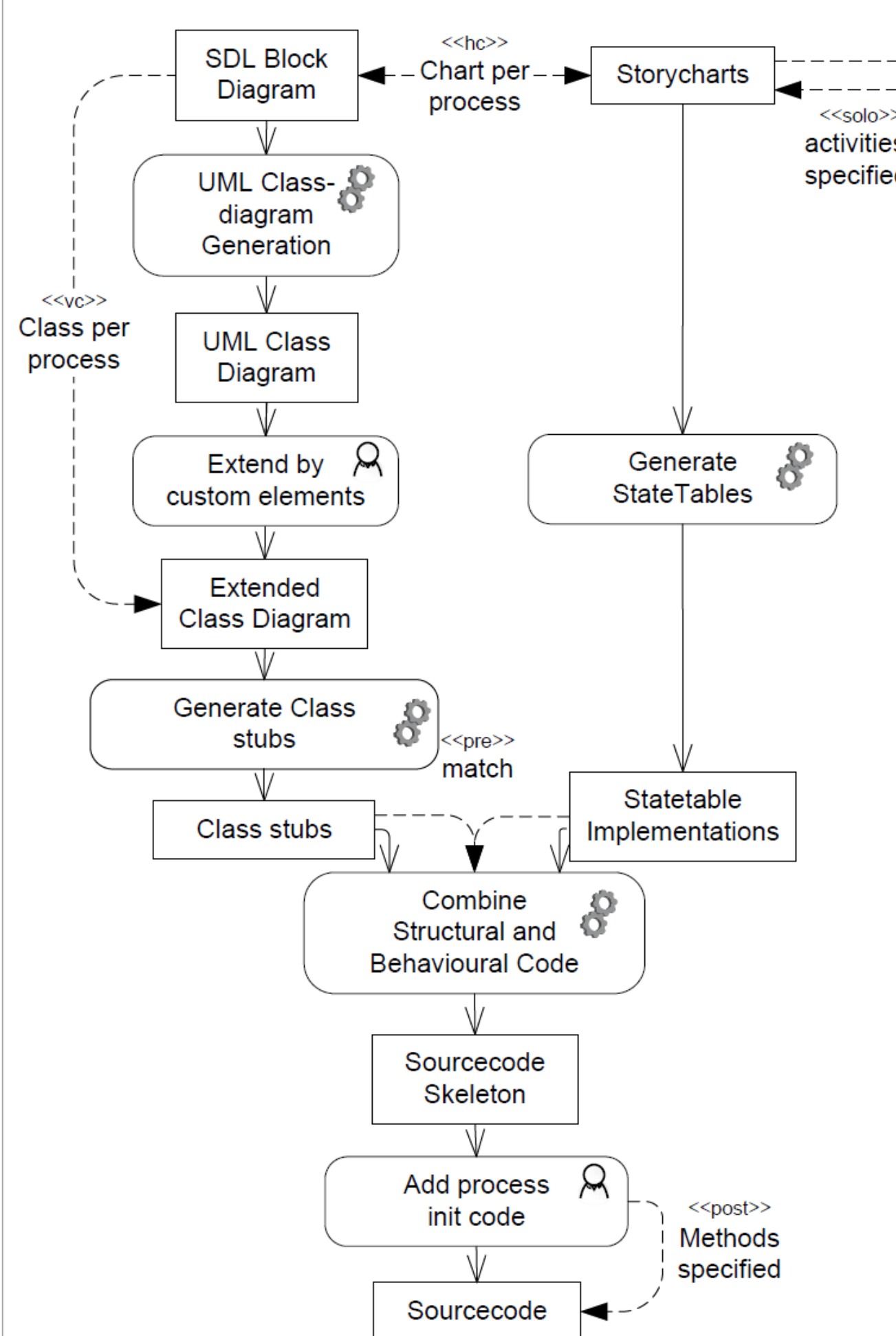
Integration Model

- Integration model (IM) serves as
 - Reusable* build script (in black)
 - Project-global* role mapping to artifacts within the VCS (Version Control System) (in blue)
 - Model of *specific build* with processing annotations (in gray)



- IM specifies object flow, operation implementations and verifications
- Verification types differentiated by:
 - Focus
 - Appropriate artifact quality
 - Correct operation execution
 - Target of consequence
 - Artifacts vs. operations
- Execution Semantic:
 - Partial builds:
 - Executes build as far as possible respecting
 - Missing artifacts and
 - Failing verifications
 - Created artifacts are stored in VCS, where they can be accessed by developers
 - Incremental builds:
 - Reuse artifacts and verification results from previous build, if possible

- Build script for factory example from [Köhler2000]:



Execution Example

- Project for enriching publication list with personal information from a family register
- Combined two ATL examples:
 - 'Families to Persons' [ATL1]
 - 'Book to Publication' [ATL2]
- Added two example verifications

Start of scenario:

- Initially no artifacts are available in VCS

Partial build:

- All operations and verifications are invalid for execution

Change 1:

- XMLSource committed to VCS

Partial build:

- Notification to execute ManualXMLToBook is sent
 - Including VCS references to available input artifact and
 - Role of required output artifact

Change 2:

- FamilySource committed to VCS

Partial build:

- Successful execution of solo verification on FamilySource
- Execution of FamilyToPersonTransformation
- PersonTarget generated

Change 3:

- BookManualResult committed to VCS
- Execution of ManualXMLToBook confirmed

Partial build:

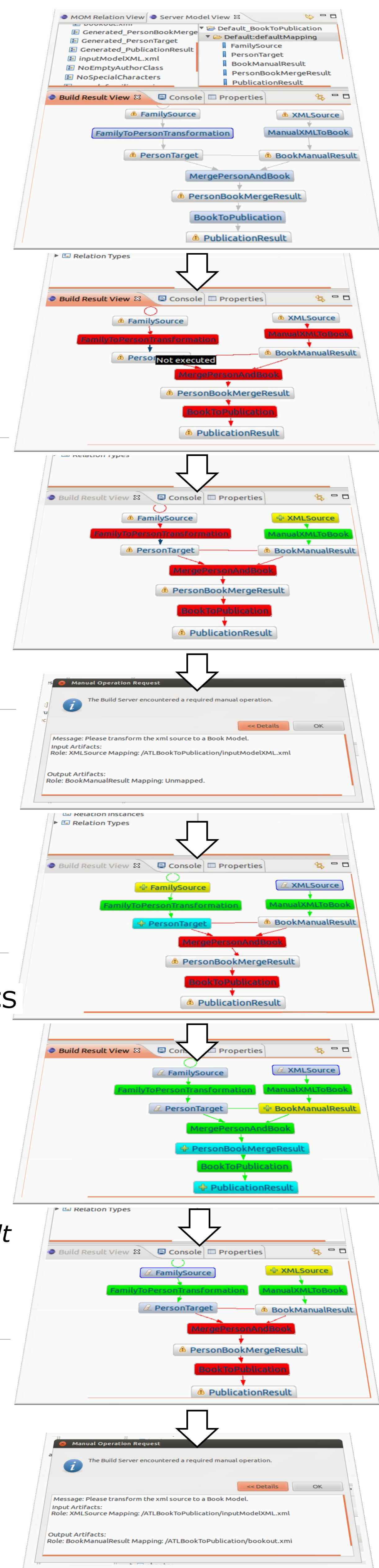
- Result of solo verification on FamilySource and PersonTarget reused from previous build
- Horizontal verification between PersonTarget and BookManualResult successfully executed
- PersonBookMergeResult generated
- PublicationResult generated

Handling of spontaneous iterations:

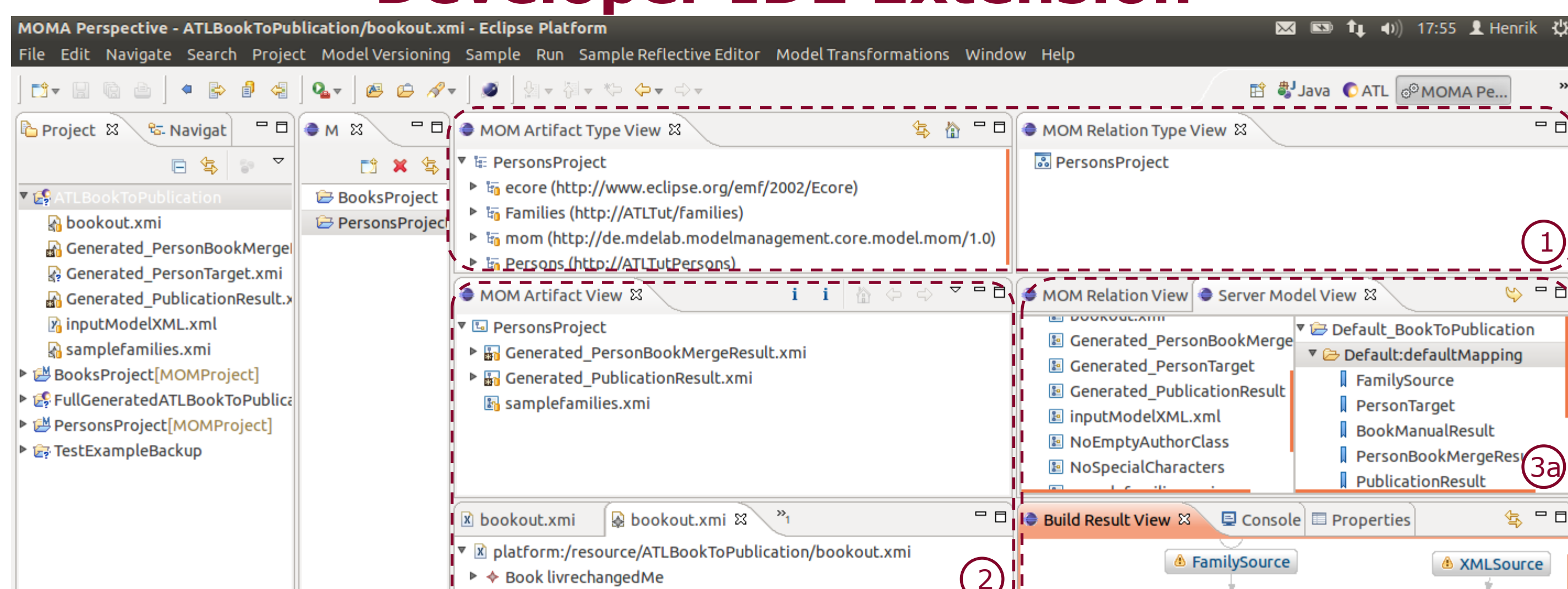
- XMLSource updated to VCS

Partial build:

- Result of solo verification on FamilySource and PersonTarget reused from previous build
- Notification to execute ManualXMLToBook is sent



Developer IDE Extension



- Type management in underlying model management VCS
- Artifacts registered to model management VCS
- Feedback view on Build Server
 - Mapping of artifacts to IM roles
 - Build result view