

# K-gileRE - Domain Knowledge Assisted Agile Requirements Evolution

Nirav Ajmeri, Manish Kumar, Preethu Rose, Smita Ghaisas

## Redefining the Way we do requirements today

Domain knowledge edge is crucially important while defining requirements

- Requirement analysts are not necessarily domain experts
- Domain knowledge is not easily available and accessible

Requirements Engineering (RE) methods presume a 'clean slate' approach

- Start with 'nothing' in place and outline a series of steps to define, analyze, specify and validate requirements collaboratively with relevant stakeholders
- *BUT* do not provide for a way to incorporate domain knowledge as an integral part of requirements definition exercise

## K-gile RE Approach

- ✓ Treats Requirements engineering as a **special case of knowledge engineering**.
- ✓ The framework presents a '**Domain Knowledge Seed**' that can be evolved into a complete specification (document + implementable models).
- ✓ The Domain Knowledge Seed consists –
  - ✓ **User Stories**
  - ✓ **Mapped Business Features**
- ✓ Features form a **Product Backlog** which is divided into multiple **Sprints**.
- ✓ Generate **Prototype** with completion of each Sprint.
- ✓ Customer **Review and Feedback**

This approach imparts agility to the requirements definition exercise.

## The K-gileRE Model

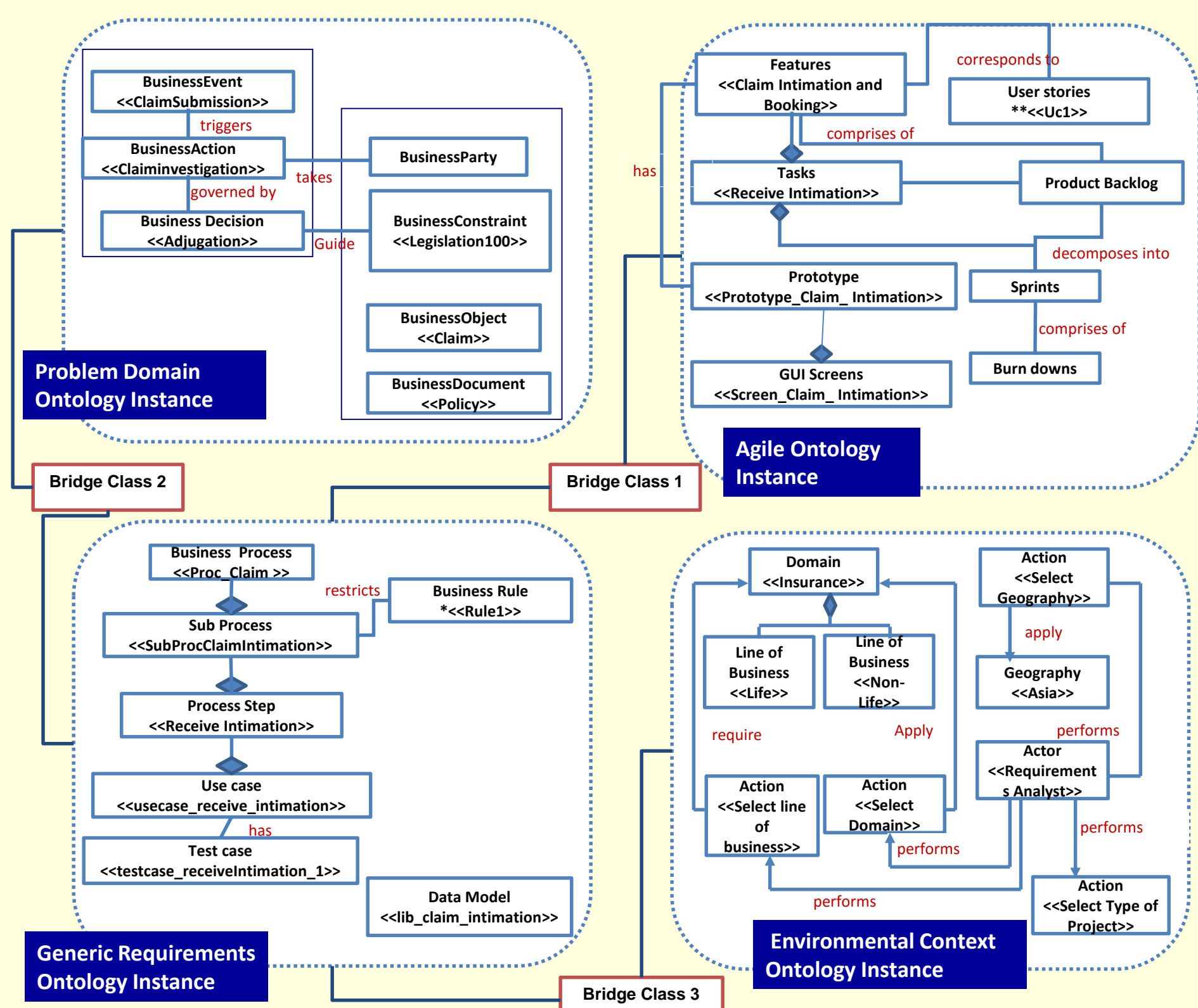
The K-gile framework integrates four different knowledge context in the form of four ontologies. We employ 'Bridge classes' and inference rules to specify semantic mappings of conclusions drawn from instance of one ontology to elements in other ontologies and provide recommendations based on the integrated inference.

The four ontologies in K-gile RE –

- **Environmental Context Ontology**
- **Agile Requirements Ontology**
- **Generic Requirements Ontology**
- **Problem Domain Ontology**

are constructed using the grounded theory and implemented using RDF-OWL schema.

*Figure shows partial example instances of the ontologies.*



## Requirements Definition and Domain Specific Assistance

Requirements Definition Activities	Domain-Specific Assistance	Example(s)
<b>Select Environmental Parameter</b>	A 'domain knowledge seed' relevant to the selected parameters is presented	<b>Parameters:</b> Domain (e.g. Insurance), line of business (e.g. life), geography (e.g. Asia) and customer (e.g. ABC), <b>Domain knowledge seed:</b> presents Modules such as Claims, Riders, Maturity
<b>Add User Story</b>	Recommendation to select from existing User Stories from Knowledge Seed.	<b>New User Story:</b> "As a <i>Claim Personnel</i> , I want <i>Claim Intimation Functionality ...</i> " <b>Recommended User Story from Domain Knowledge Seed:</b> "As a <i>Claim Personnel</i> , I want <i>Claim Intimation and Booking Feature</i> so that <i>Data captured for the claims and related details can be saved in the system for claims processing and Claim can be Booked.</i> "
	Recommendation to map to Business Feature in Knowledge Seed.	<b>New User Story:</b> "As a <i>Claim Scrutinizer</i> , I want <i>Scrutiny functionality...</i> " <b>Recommended Business Feature in Domain Knowledge Seed:</b> Initial Scrutiny
<b>Select features (from the domain knowledge seed) relevant to project</b>	Recommendations to include business rules/policies relevant to features,, Business Glossary, Business Process, , Include Closely Related Terms,	<b>Selected Feature:</b> Claim intimation and booking <b>Business Terms:</b> Assignee, <b>Rules:</b> Laws of the land with respect to claims, in Asia, Policies of the selected company (ABC), conflicting Features
<b>Form product backlog and sprints thereafter</b>	Recommendations to include inter- dependant features in the same sprints, Splitting of a Feature	'Claim intimation' and 'Claim review and inspection' may be included in the same Sprint.
<b>Editing elements such as Feature, User Story, use cases from the seed</b>	Recommendations to adhere to terminology, detection of new terms and recommendations include them in glossary, data models, recommendations to specify associations between terms.	<b>User story text:</b> "As an Insurer, I want to have Claim Intimation & Booking feature with automated agreement verification in my Insurance Application so that the verification process gets completed within 2 days." <b>Recommendation:</b> 'Policy' is the most accepted term. Would you like to replace 'Agreement' with 'Policy'?
<b>Generate Prototype</b>	Typical screens, partial data models , use cases	<b>Feature selected for prototyping:</b> 'Claim intimation' <b>Recommendations:</b> Sample screens depicting the 'Claim intimation' activities, data models ( e.g. consisting of Claim, Policy, Agent, corresponding use cases).