



Towards Conflict-Free Collaborative Modelling using VS Code Extensions



Rijul Saini



Gunter Mussbacher



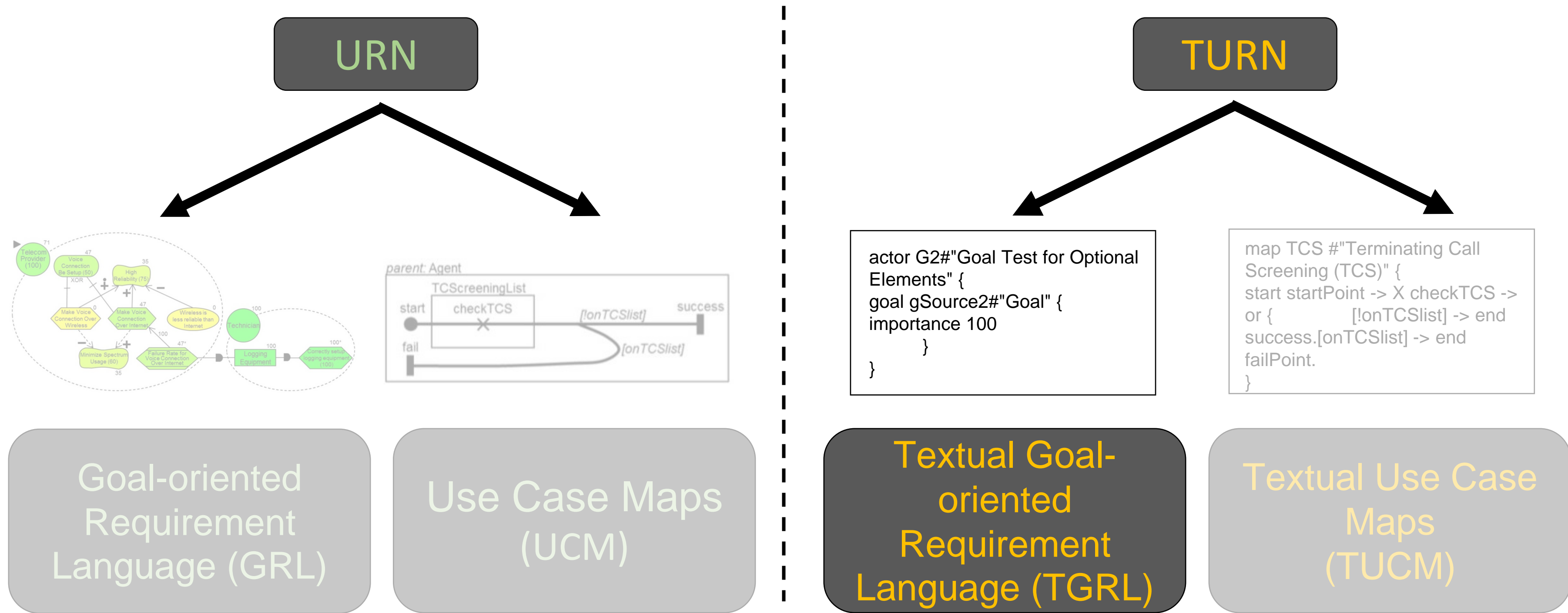
**ACM/IEEE 24th International Conference on
Model Driven Engineering Languages and Systems (MODELS'21)
Hands-on Workshop on Collaborative Modeling 2021 (HoWCoM)
October 11, 2021**

With the Need for Reduced Development Time and Distributed Teams in Software Development, Better **Collaboration** Support is Required



The Need to Collaborate More Quickly While Building and Analyzing Models Becomes More Imperative.

Our Previous Work [1] - We Introduce **tColab Framework** To Enable Collaborative Modelling of **TURN (Textual User Requirements Notation)**



[1] Saini, R., Bali, S., and Mussbacher, G. (2019) Towards Web Collaborative Modelling for the User Requirements Notation Using Eclipse Che and Theia IDE. 11th Workshop on Modelling in Software Engineering (MiSE 2019), Montreal, Canada, May 2019. IEEE CS

Some Challenges Remain Unaddressed with the Current Support (tColab) for Collaborative TGRL Modelling

Current Support - tColab

TGRL Xtext

extend

Sprotty Language

Combination of Textual Notation and its Graphical Representation

Language Server Protocol Removes Redundancy Problem

TGRL
Theia

Integration Support for Eclipse Theia IDE and Eclipse Che

Services

Challenges

Last-Write-Wins Policy in Eclipse Che

Modelling Conflicts Upon an Undo Operation

Lack of Awareness of Modellers' Changes

Conflicts in Modelling Intentions

Editor Support Only for Eclipse Theia IDE

Synchronization in a Multi-Tenancy View

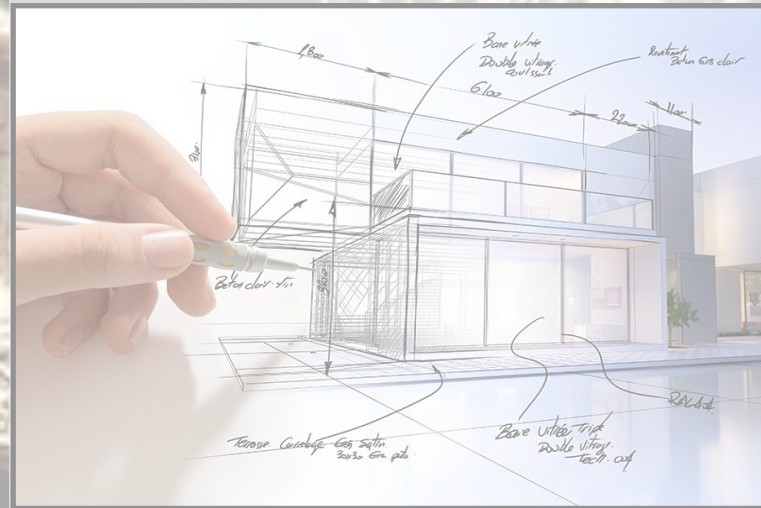
Support For Conflict-Free Collaborative Modelling is Required Without Reinventing the Wheel



Support For Conflict-Free Collaborative Modelling is Required Without Reinventing the Wheel



**Possible Solutions
For Conflict-Free
TGRL Modelling**



**Architectural
Design of
Proposed VS Code
TGRL Extension**



**Demonstration of
TGRL
Collaborative
Modelling**



**Open Research
Questions for
Participants**

Possible Solutions to Facilitate Conflict-Free TGRL Collaborative Modelling

Current Support - tColab

TGRL Xtext

extend

Sprotty Language

Combination of Textual Notation and its Graphical Representation

Language Server Protocol Removes Redundancy Problem

TGRL
Theia

Integration Support for Eclipse Theia IDE and Eclipse Che

Services

Challenges

Last-Write-Wins Policy in Eclipse Che

Modelling Conflicts Upon an Undo Operation

Lack of Awareness of Modellers' Changes

Conflicts in Modelling Intentions

Editor Support Only for Eclipse Theia IDE

Synchronization in a Multi-Tenancy View

Possible Solutions to Facilitate Conflict-Free TGRL Collaborative Modelling

Challenges

Modelling Conflicts
Upon an Undo Operation

Conflicts in Modelling
Intentions

Synchronization in a
Multi-Tenancy View

Solution 1

VS Code TGRL Extension +
VS Code Live Share

Supports Recovery of
Desired State of
TGRL Models

Supports IDE features for
Increasing Awareness and
Communication

Supports VS Code like
Editors, e.g., VS Code and
Eclipse Theia (new version)

Implementation Available

Solution 2

VS Code TGRL Extension +
VS Code Teletype Share

Supports Recovery of
Desired State of
TGRL Models

Supports IDE features for
Increasing Awareness
(Modellers' Cursors)

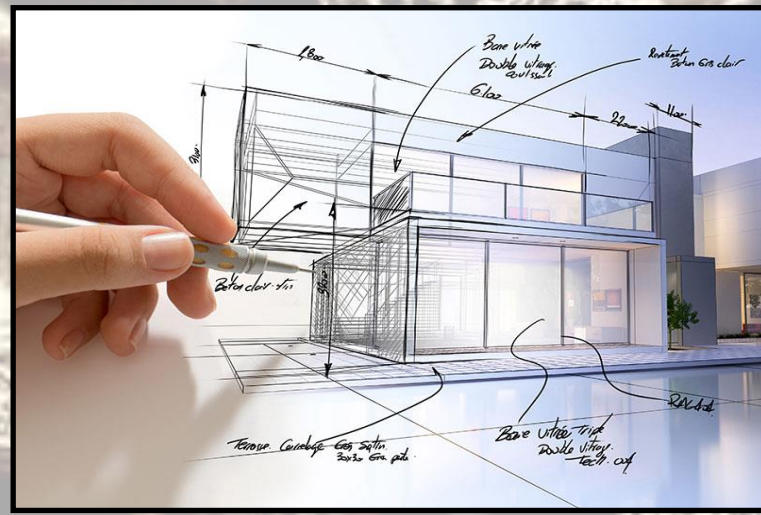
Need to Build Headless
Teletype Host for Multiple
Editors

Work in Progress

Support For Conflict-Free Collaborative Modelling is Required Without Reinventing the Wheel



Possible Solutions
For Conflict-Free
TGRL Modelling



**Architectural
Design of
Proposed VS Code
TGRL Extension**

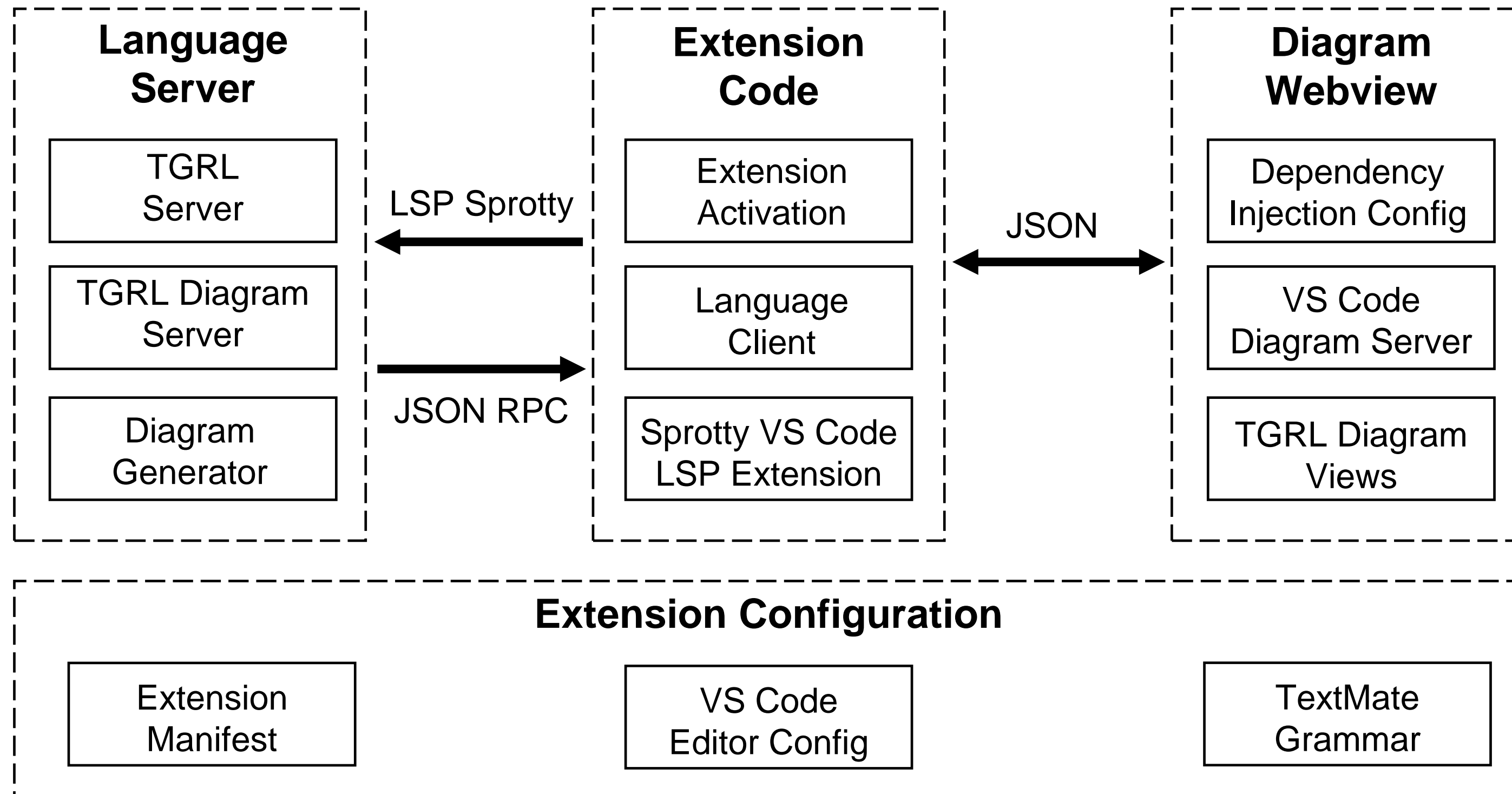


Demonstration of
TGRL
Collaborative
Modelling



Open Research
Questions for
Participants

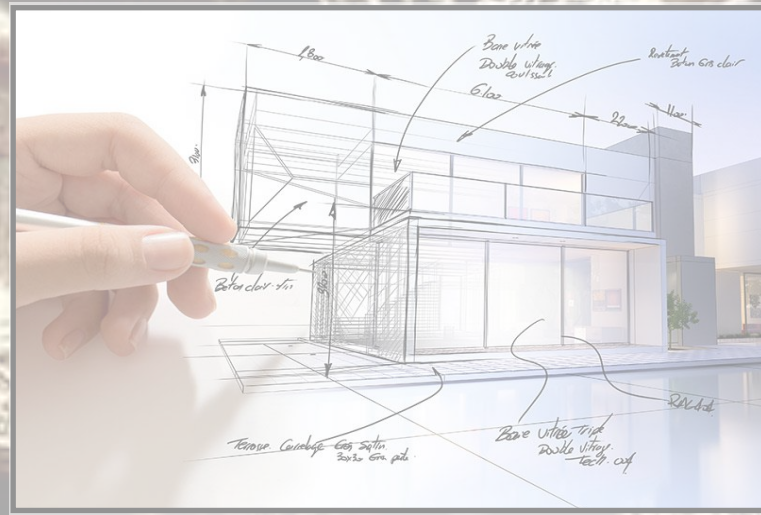
Architecture of Proposed TGRL VS Code Extension



Support For Conflict-Free Collaborative Modelling is Required Without Reinventing the Wheel



Possible Solutions
For Conflict-Free
TGRL Modelling



Architectural
Design of
Proposed VS Code
TGRL Extension



Demonstration of
TGRL
Collaborative
Modelling

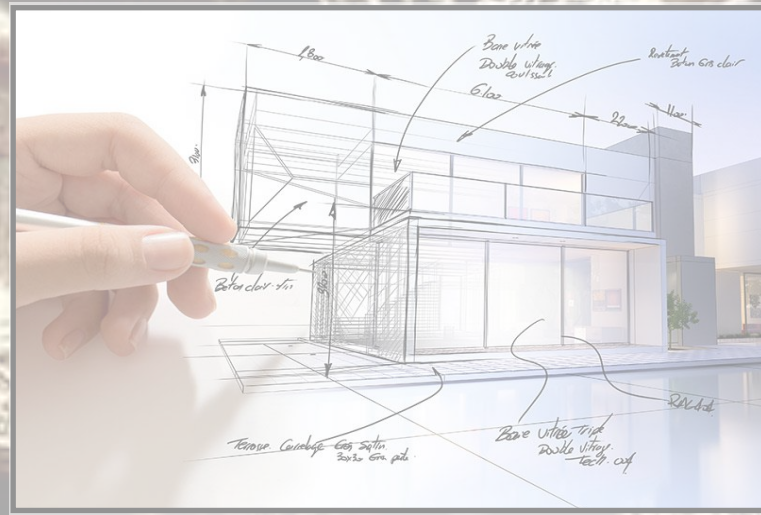


Open Research
Questions for
Participants

Support For Conflict-Free Collaborative Modelling is Required Without Reinventing the Wheel



Possible Solutions
For Conflict-Free
TGRL Modelling



Architectural
Design of
Proposed VS Code
TGRL Extension

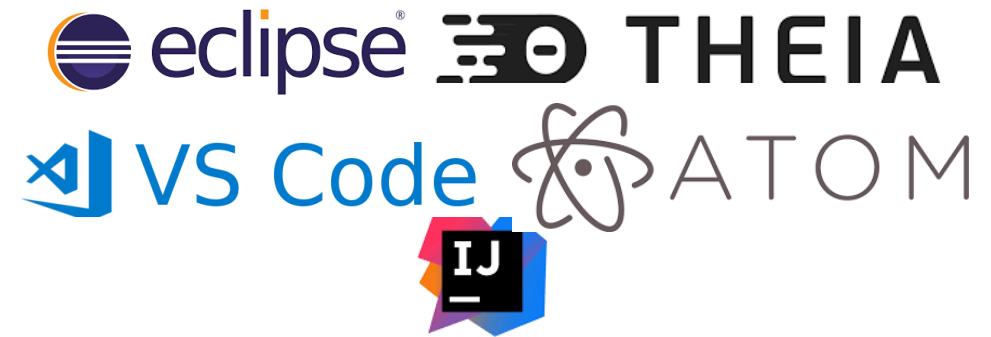


Demonstration of
TGRL
Collaborative
Modelling



Open Research
Questions for
Participants

Open Research Questions for Participants



RQ1: Which editors or IDEs are preferred by participants for performing modelling and development together?



RQ2: What is the satisfaction level of participants for handling conflicts with our proposed tool?



RQ3: What is the performance of the collaborative environment facilitated by our solution in terms of scalability and time?



RQ4: What other potential features and use case scenarios are possible with our proposed tool?

Some Challenges Remain Unaddressed with the Current Support (tColab) for Collaborative TGRL Modelling

Current Support - tColab

TGRL Xtext

extend

Sprotty Language

Combination of Textual Notation and its Graphical Representation

Language Server Protocol Removes Redundancy Problem

Integration Support for Eclipse Theia IDE and Eclipse Che

Services

TGRL
Theia

Challenges

Last-Write-Wins Policy in Eclipse Che

Lack of Awareness of Modellers' Changes

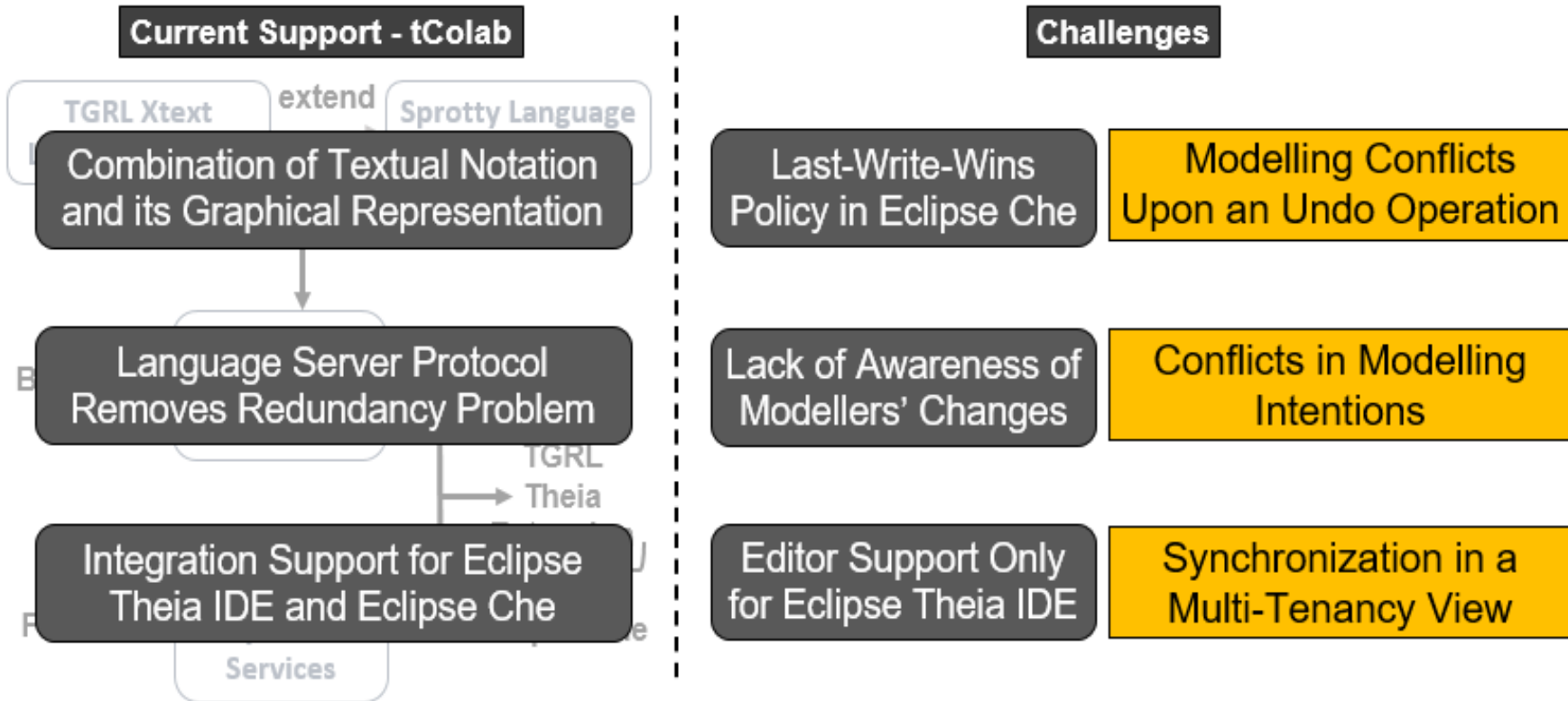
Editor Support Only for Eclipse Theia IDE

Modelling Conflicts Upon an Undo Operation

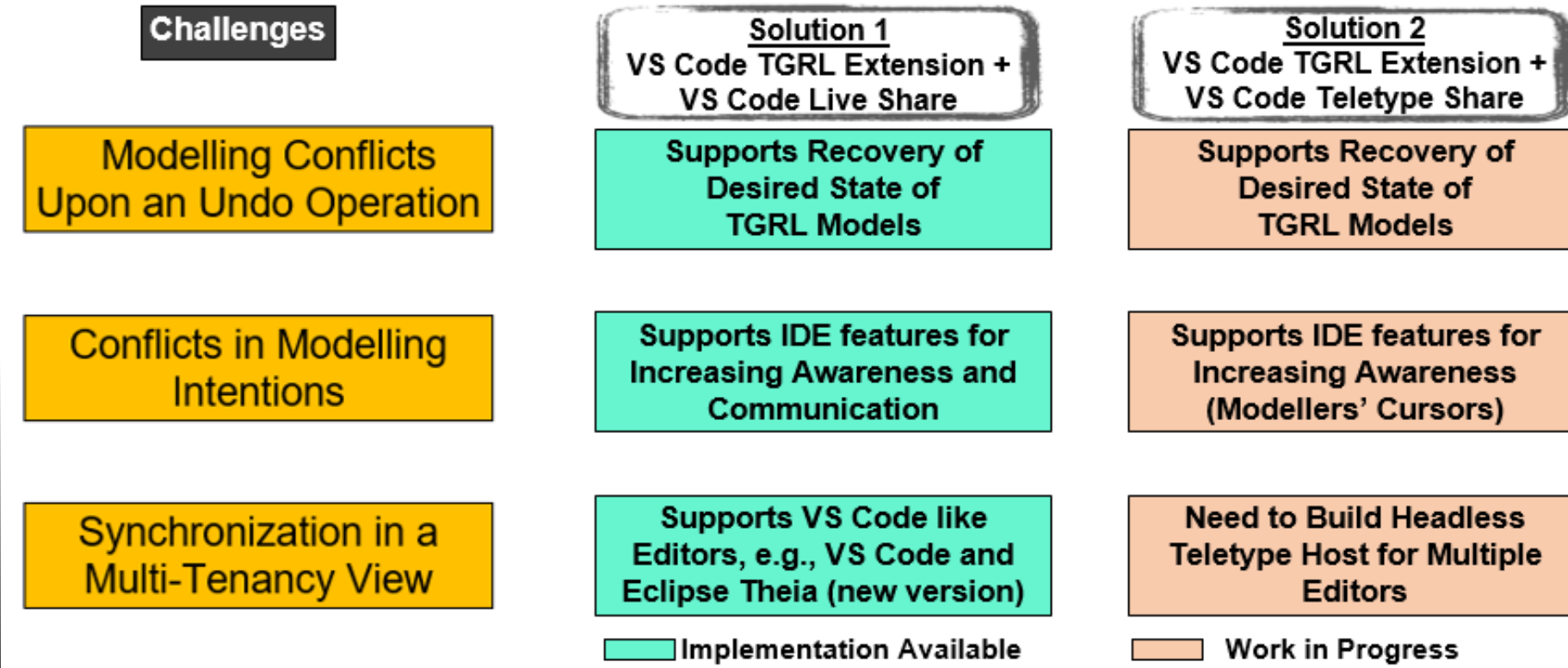
Conflicts in Modelling Intentions

Synchronization in a Multi-Tenancy View

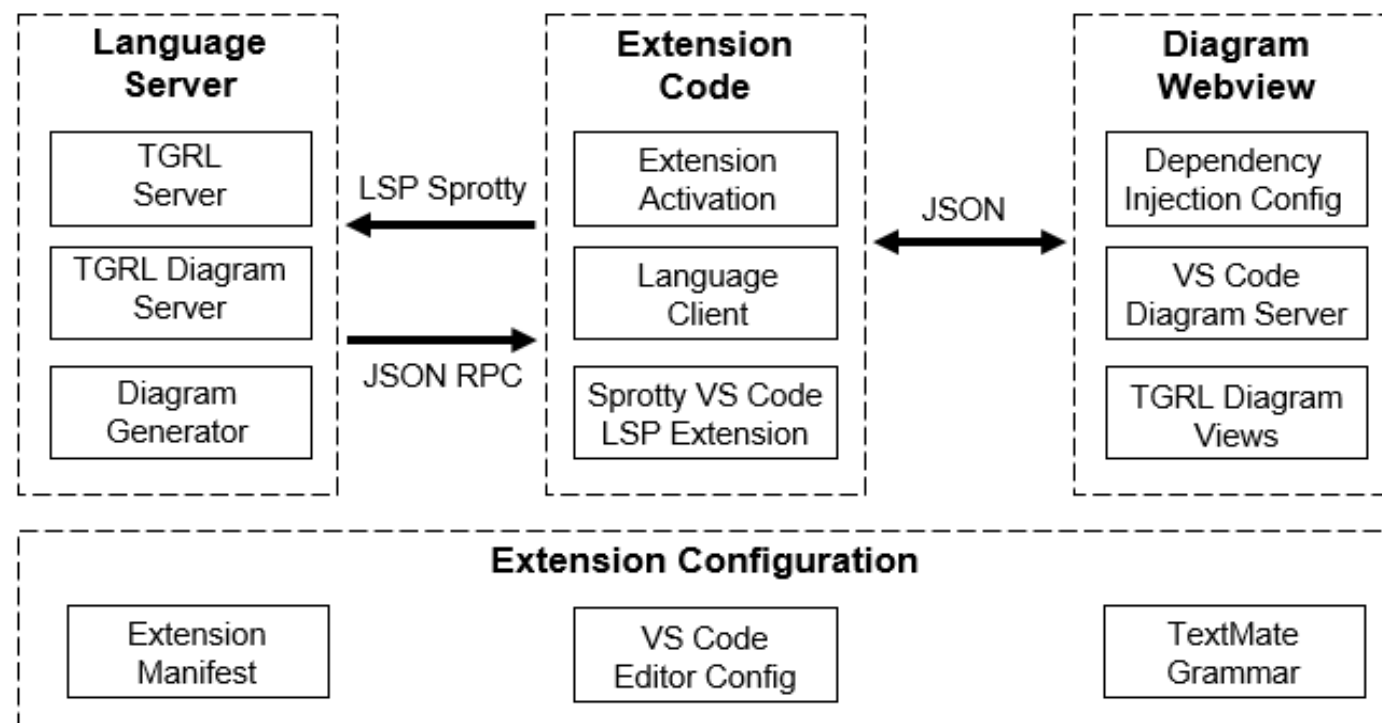
Some Challenges Remain Unaddressed with the Current Support (tColab) for Collaborative TGRL Modelling



Possible Solutions to Facilitate Conflict-Free TGRL Collaborative Modelling



Architecture of Proposed TGRL VS Code Extension



Open Research Questions for Participants

