Appendix A. N-way Design Rational

After the integration process, as shown in Figure A.11, POLYMER provides users with merging documentation, referred to as NDR (N-way Design Rationale). This documentation allows users to verify the equivalence of elements during the merge. For example, in the tenth row of Figure A.11, the user confirms that the class Staff from version V1 is equivalent to Personnel from another version.

It's important to emphasize that the algorithms integrated into POLYMER are deterministic. This means that if the merging process is executed with the same inputs, it will consistently yield the same results. Similarly, Reuling's N-way merging approach [16] also exhibits this deterministic behavior, ensuring reliable and reproducible outcomes in both merging processes.

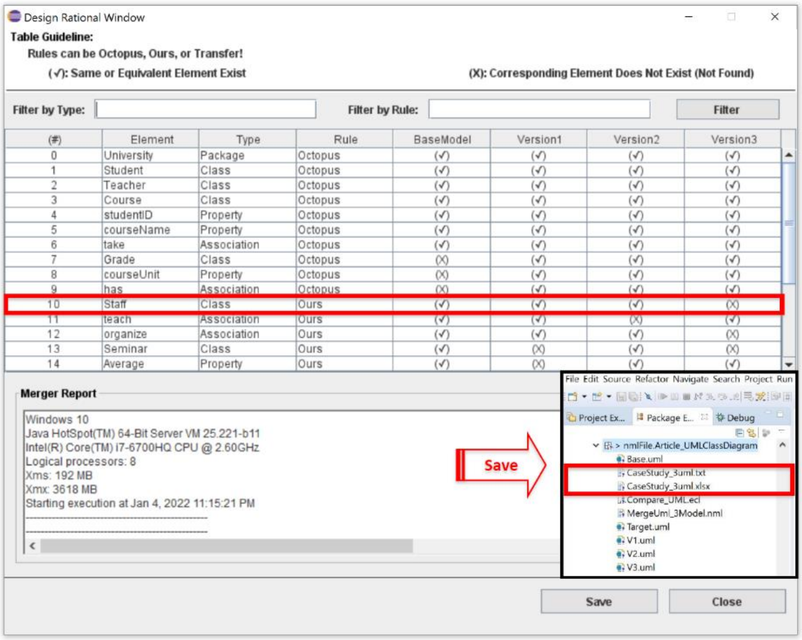

Figure A.11: Design rationale window to report merger decisions for the running example.

Appendix B. NML Editor

Figure B.12 illustrates a segment of the integration rules specified for a UML class diagram within the implemented 1010 NML editor. As shown in the Figure, the NML program is structured around two key components: a set of imported models and a collection of merging rules. These rules include mergeOctopus, mergeOurs, and transfer, each playing a specific role in the merging process. This organization of rules allows NML to efficiently manage and merge different model versions.

NML Editor

File Edit Run

```
import Version1 : "V1.uml";
import Version2 : "V2.uml";
import Version3 : "V3.uml";
import TargetModel : "Target.uml";
import BaseModel : "Base.uml";
rule CL_OctopusA
     mergeOctopus v1 : Version1!Class
     with v2 : Version2!Class , v3 : Version3!Class
     withBase v0 : BaseModel!Class
     into vt : TargetModel!Class {
          vt.name = v0.name ;
          vt.isAbstract = v1.isAbstract;
          vt.package = v2.package.equivalent() ;        }
rule CL_OursB
     mergeOurs v1 : Version1!Class
     with v2 : Version2!Class , v3 : Version3!Class
     withBase v0 : BaseModel!Class
     exists_in 2
     priority P1 : [v1, v2, v3, v0] , P2 : [v3, v2, v1, v0]
     into vt : TargetModel!Class {
          vt.name = P1.name ;
          vt.isAbstract = P2.isAbstract;
          vt.package = P2.package.equivalent() ;        }
rule CL_Transfer
     transfer vs : Source!Class
     from Source : (Version1, Version2, Version3)
     to vt : TargetModel!Class {
          vt.name = vs.name ;
          vt.isAbstract = vs.isAbstract ;
          vs.eContainer.equivalent().packagedElement.add(vt);        }
```

Figure B.12: Excerpt of the specification of integration rules for UML class in the NML editor

Appendix C. The usability evaluation model

Table C.7 presents the usability evaluation model for the NML language, structured using the Goal Question Metric (GQM) approach. In this model, the usability of NML is evaluated based on the ISO 9241- 11 framework, which defines effectiveness, efficiency, and satisfaction as the primary quality characteristics for usability assessment.

Table C.7: Evaluation model for NML language

Quality characteristics	Goals	Question	Metrics
Effectiveness	Simplicity	Is the proposed language easy to learn?	Satisfaction with help provided
			Time taken to learn
			Number of mistakes while learning
		How do you evaluate the readability and comprehensibility of NML language?	Satisfaction with writing merging rules
			Satisfaction with finding appropriate rules
		To what extent are the NML keywords semantic transparent?	Satisfaction while learning
			Satisfaction while working with rules
			Satisfaction with understandable
Efficiency	Features	To what extent can the integration description language facilitate the integration process?	Number of models provided for merging
			Time taken to write rules
		How useful is the integration description language?	Time taken to merging models
			Satisfaction with reusability of written rules
			Satisfaction with merged model
		How much NML can facilitate the integration process of large models?	Size of models provided for merging
			Satisfaction with covering all scenarios
Satisfaction	Attractiveness	Is the created editor appropriate for writing NML integration rules?	Satisfaction with help provided
			Satisfaction while writing rules
		To what extent is the editor user-friendly?	Satisfaction with interface graphics
			Satisfaction with interface arrangement

Appendix D. The performance results

The experimental subjects were executed ten times on the proposed approach to ensure consistent and reliable results. Table D.8 provides detailed information about the runtimes of the proposed approach when applied to three different systems: Hospital, Warehouse, and PPU (Pick and Place Unit).

Table D.8: Runtimes of proposed approach on Hospital, Warehouse, and PPU systems.

		Run (Proposed Approach)									
System	Operation	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Hospital	Compare	1.93s	0.68s	1.61s	0.62s	1.66s	0.49s	0.53s	**0.42s**	0.50s	0.56s
	Merge	4.96s	4.20s	4.90s	6.43s	5.16s	4.86s	3.50s	7.38s	5.67s	**3.03s**
	Total Process	6.89s	4.88s	6.51s	7.05s	6.82s	5.35s	4.03s	7.80s	6.17s	**3.59s**
Warehouse	Compare	4.09s	1.87s	1.33s	7.30s	1.34s	1.63s	3.75s	1.73s	1.48s	**1.20s**
	Merge	11.02s	12.85s	15.36s	**7.83s**	9.79s	8.06s	9.68s	11.21s	11.15s	14.42s
	Total Process	15.11s	14.72s	16.69s	15.13s	11.13s	**9.69s**	13.43s	12.94s	12.63s	15.62s
PPU	Compare	1.13s	0.38s	0.40s	**0.22s**	0.25s	0.26s	0.23s	0.25s	0.25s	0.25s
	Merge	1.14s	0.87s	1.73s	0.78s	0.79s	2.88s	0.90s	0.80s	0.80s	**0.71s**
	Total Process	2.27s	1.25s	2.13s	1.00s	1.04s	3.14s	1.13s	1.05s	1.05s	**0.96s**

Appendix E. The scalability results

Table E.9 displays the runtimes for the merging process across ten subsets of the Warehouse dataset, each with a different number of elements. The subsets were generated by randomly modifying the original dataset to include varying proportions of model elements, with sizes ranging from 366 to 3646 elements per subset.

Table E.9: Runtimes of merging process across Warehouse subsets

		Run (Proposed Approach)									
Subset	Elements	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
1	366	1.68s	1.17s	1.05s	1.01s	**0.80s**	1.53s	0.98s	0.87s	0.94s	0.83s
2	730	4.12s	3.27s	4.35s	3.15	3.74s	2.89s	3.53s	2.62s	3.78s	**2.56s**
3	1094	5.74s	6.06s	5.61s	4.21s	7.11s	**4.10s**	7.23s	4.13s	11.58s	4.82s
4	1458	8.94s	9.19s	9.30s	5.56s	10.08s	9.59s	**5.44s**	8.37s	9.06s	9.81s
5	1826	15.11s	14.72s	16.69s	15.13s	11.13s	**9.69s**	13.43s	12.94s	12.63s	15.62s
6	2190	14.93s	17.51s	15.11s	13.22s	15.41s	16.98s	16.87s	17.70s	**12.88s**	16.64s
7	2554	16.37s	19.78s	18.96s	**11.88s**	15.68s	20.67s	18.31s	14.09s	16.07s	17.41s
8	2918	**15.28s**	17.17s	19.73s	16.95s	18.62s	20.42s	18.61s	22.79s	18.71s	15.44s
9	3274	17.58s	**17.22s**	22.26s	19.58s	21.87s	17.96s	18.70s	20.09s	19.92s	21.04s
10	3646	18.09s	26.99s	19.26s	19.26s	**17.33s**	18.19s	18.85s	18.79s	20.77s	20.03s