1 SUPPLEMENTARY: PRELIMINARY EVALUATION ON A PEDAGOGICAL STUDENT REGISTRATION EXAMPLE

Our tool provides provisions for software professionals to extract relevant ontology as applicable to a new problem by providing two optional input holders: (a) sequence diagram holder and (b) domain requirement holder. Both these inputs help extract refined or customized ontology. As a proof of work we have evaluated it on a Sample Domain - Student course registration used in the paper as running example.

1.1 Extracts Reasonable Ontology

We compare the extracted ontology from a design diagram of a solution proposed for university student registration problem to a golden ontology. High values of matching entities between the two ontologies indicate that our approach extracts similar ontology to existing ones.

In our case we consider domain 'University' (whose subset is the course registration) and obtain the golden ontology by aggregating several individual 'University' ontologies from [22] [23] [24]. The aggregation is done and a golden ontology thus obtained is shown in Figure 12.

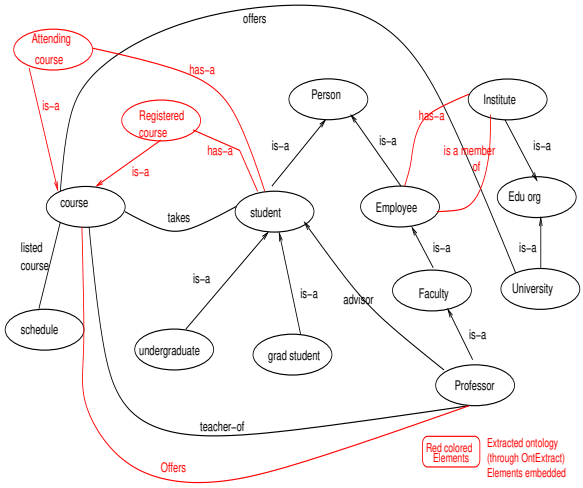

Figure 12: Golden Ontology - University

Figure 13 presents the ELRD (after removing properties using rules as presented in Table 1) of the solution used in the paper. We refer to this as BaseOnt. Base-ELRD is the term used to refer to this ELRD. We compare Base-ELRD with the golden ontology of university (Figure 12).

We compute the matching entities between the extracted ontology (ELRD) to the Golden Ontology and the results are presented in Table 6

Table 6 gives the number of matching entity and relationship elements between the golden ontology and Base-ELRD (Figure 12 and Figure 13). The class Course is related to registered course through inference property of 'HAS-A' relationship. Using Table 6 and Equations 1 and 2, we calculate the matching (similarity) entities and relationships metrics as presented in Table 7:

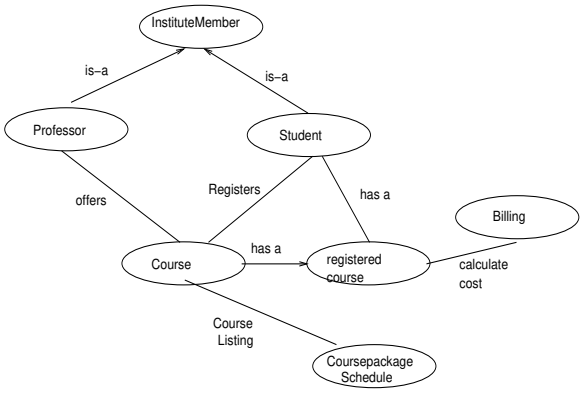

Figure 13: ELRD for student course registration

Matching entities	Matching Relationship
Student, Professor, Course (registered course, through inference property of 'HAS-A' relationship), schedule/courselisting, Institutemember. **Total: 6**	is-a (Professor, institute-member), is-a(Student, institutemember) , has-a (student,course ;sub-category registered course), Offer(professor,course) term 'offer' is matched with 'teacher-of', has-a(course, registered course), has-a (course [subcategory: registeredcourse], courselisting). **Total:6**

Table 6: Matching found for golden ontology and extracted ontology

Criteria	Number Value
Total number of entities class in ELRD	7
Total number of relationship(s) found in ELRD	8
Total number of matching entities found	6
Total number of matching relationship(s) found	6
Extra entity identified	1
Extra Relationship identified	2
Percentage E_{sim}	$\frac{6}{7} * 100 = 85.7\%$
Percentage R_{sim}	$\frac{6}{8} * 100 = 75\%$

Table 7: Comparison of the ELRD with University Ontology (gold standard)

The extracted ontology from the UML diagrams by preserving relationship and entities shows about 85.7% match with the golden standard and about 75% match with the relationships. High values of matching entities and relationships between the extracted ontology and golden ontology show that our approach extracts ontology similar to existing (golden) ontologies. In addition, our method detects other elements that does not have corresponding matches in the golden ontology thereby enriching the existing ontologies with additional elements. This directly proves that our method is correct since otherwise there would have been large mismatch between the golden ontology and ontology extracted through through our approach.

1.2 Re-usability Evaluation

Design of software solutions is greatly enhanced if the software architect has/gets 'a-priori' knowledge about concepts involved in the problem specification [3]. The aim is to show that the extracted ontology is reusable in a number of situations like: (a) Reuse in new solution development (b) Reuse in new problem scenario.

1.2.1 Reuse In New Solution Development. Let us consider a solution for a problem domain to be developed by different group of architects. A solution provided by a group will greatly help the other architects to detect artifacts and provide better solution in the problem. We illustrate this by showing that there exists large amount of common concepts among the solutions, if the solution is developed by different group of architects.

We take two different solutions SolB and SolC provided by different group of experts for the course registration problem specification. Solution 1 was taken from the tutorial provided by Rational Software Corporation and Solution 2 was taken from from a classroom lecture [58]. Comparisons are made to determine the number of common matching entities between these solutions and the BaseOnt. High value of this metric indicates that the ontology from one solution is highly re-usable for new solutions of the problem. We measure entity reuse using Equation 3. In the similar way, Equation 4 gives the relationship reuse. R_{reuse}.

SolB with Base Ontology: The ERLD of the extracted ontology for Solution 1 is provided in Figure 14.

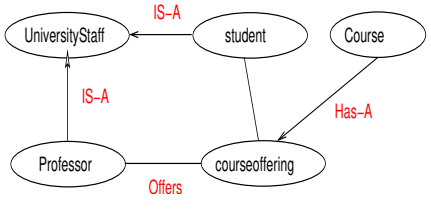

Figure 14: SolB ELRD diagram

Table 8: Matching for SolB and BaseOnt

Matching Entities	Matching Relationship
Course, Student, courseoffering, Professor, Universitystaff	is-a (professor,university staff), is-a(professor, universitystaff), has-a(course, courseoffering), Using (Courselisting, student)

Table 8 presents the matching elements between the ELRD of SolB and Base Ontology. In some cases, inference rules were extended to find similar relationships. For example has-a(course, courseoffering) in Solution 1 is similar to has-a(course, courseoffering) in the BaseOnt through 'courseSchedule'. Classes universityStaff in Solution 1 and instituteMember in BaseOnt represent the same entity and is resolved manually.

Equation 3 and 4 are used to measure re-usability as presented in Table 9. Reuse of entities and relationship(s) has been observed as high as 100% and 80% respectively.

Criteria	Number value
Number of entities present in solution 1 ELRD	5
Number of entities matching with base ontology	5
Number of relationship(s) present in the solution 1 ELRD	5
Number of relationship(s) matching with base ontology	4
Re-use of entities present E_{reuse}	$\frac{5}{5} * 100 = 100\%$
Re-use of relationship present R_{reuse}	$\frac{4}{5} * 100 = 80\%$

Table 9: Comparison of Solution 1 with BaseOnt

SolC with Base Ontology: Figure 15 gives the ELRD diagram for Solution 2.

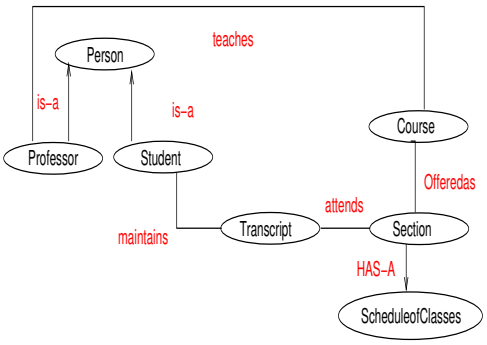

Figure 15: SolC ERLD diagram

Table 10 gives the matching elements between ELRD of Sol2 and BaseOnt. class person match is obtained using extended annotations using external source. Re-usability is calculated as shown in Table 11.

Matching Entities	Matching Relationship
student, professor, person,course, scheduleclass	is-a(student,person), is-a(professor,person), teaches(professor,course), has-a(course, schedulecourse)

Table 10: Matching for Solution 2 and BaseOnt

Criteria	Number value
Number of entities present in solution 2 ELRD	7
Number of entities matching with base ontology	5
Number of relationship(s) present in the solution 2 ELRD	7
Number of relationship(s) matching with base ontology	4
Re-use of entities present E_{reuse}	$\frac{5}{7} * 100 = 71.4\%$
Re-use of relationship present R_{reuse}	$\frac{4}{7} * 100 = 57.14\%$

Table 11: Comparison of Solution 2 with BaseOnt

Our results indicate that there is indeed great amount of reuse ranging from 70% -100% entity level and 57% to 80% relationship level reusable components. This shows OntExtract provides provisions for reusing previously available ontology to new solutions.

1.2.2 Reuse in New Problem Scenario. Our approach provides provisions for the software professionals to model the use cases of the new problem as sequence diagram and extract related ontology.

The architect provides his solution as design diagrams. We extract the domain concepts through *OntExtract* and call the ontology as ArchitectOnt. We model the use-case requirement as a sequence diagram and use this as an filter to extract ontology from the BaseOnt. We call this ontology as NewScenarioOnt. We find the number of elements of NewScenarioOnt is present in the ArchitectOnt. High values of this metric shows that large number of domain entities could have been reused.

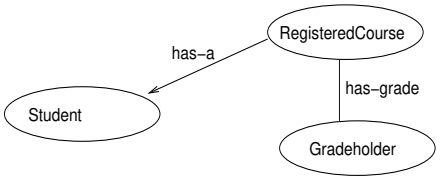

Figure 16: Class diagram by architect

We take an example where the software architects wants to design requirements to implement 'ClearDues' use-case of the administration department. The 'ClearDues' checks if the student is valid and has cleared all the courses with clear grades. The architect intends to design new classes and relationship with existing course registration system. For the sake of brevity, only the ELRD diagram (Figure 16) of the class diagram developed by the architect (ArchitectOnt) is provided.

We use the sequence diagram pertaining to 'ClearDues' (Figure 18) as a filter to extract ontology from Figure 6. This ontology extracted is henceforth referred to as the NewScenarioOnt. All the classes in the sequence diagram are considered to extract NewScenarioOnt. The NewScenarioOnt extracted in ELRD form is given in Figure 17.

Criteria	Number Value
Number of matching entities between the NewScenarioOnt and ArchitectOnt	2
Number of matching relationship(s) between the NewScenarioOnt and ArchitectOnt	1
Total number of entities present in the NewScenarioOnt	2
Total number of relationship present in the NewscenarioOnt	1
Reuse of entities E_{reuse}	$\frac{2}{2} * 100 = 100\%$
Reuse of relationship E_{reuse}	$\frac{1}{1} * 100 = 100\%$

Table 12: Reuse in similar but different domain

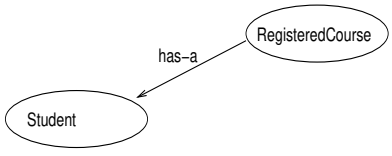

Figure 17: Relevant Ontology Extracted

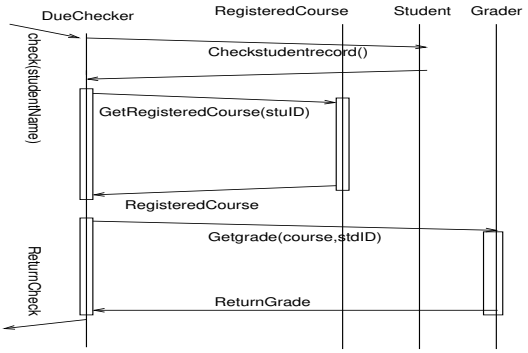

Figure 18: Sequence diagram for Usecase 'ClearDues'

The results presented in Table 12, indicates that the sequence diagram of a new functionality helps identifying reusable components from previously available solutions. Though this situation is ideal and has 100% reusable components, in practice the amount of re-usability might be less than the one presented here. However, the domain ontology is intended for reuse of domain concepts whereever required, one might expect high values of reusability.

2 GOLDEN ONTOLOGY FOR DIFFERENT DOMAINS

Figure 19 presents the Golden ontology for the road accident domain.

A elaborate Golden Ontology for the media domain is presented in Cite [55] presents the Golden Ontology for the media domain.

Golden ontology for retail was taken directly from [56]. It covers about 51 class concepts around retail industry.

Partial listing of the golden ontology for Banking domain is presented in Figure 20.

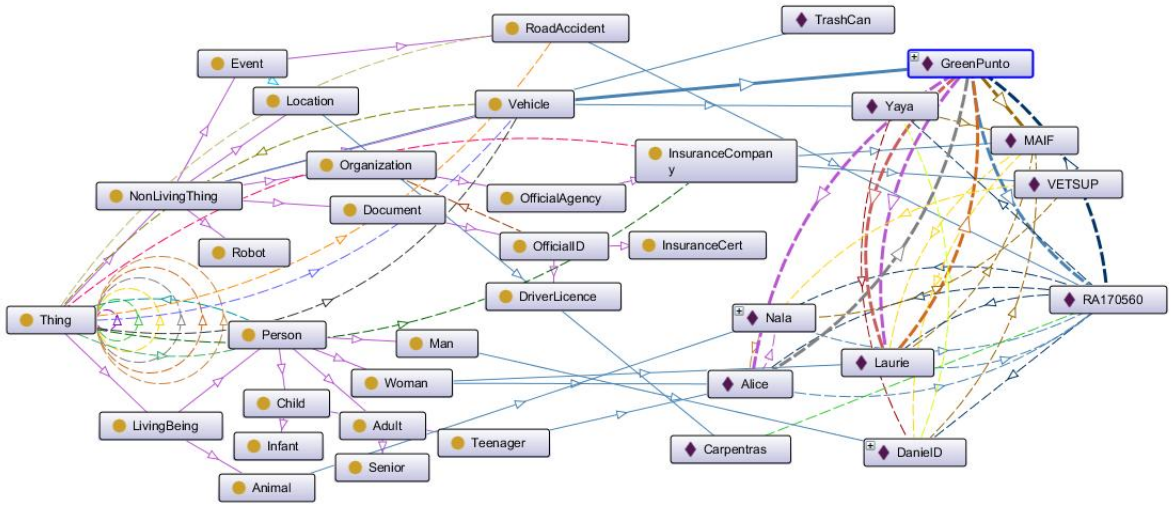

Figure 19: Road Accident Golden Ontology

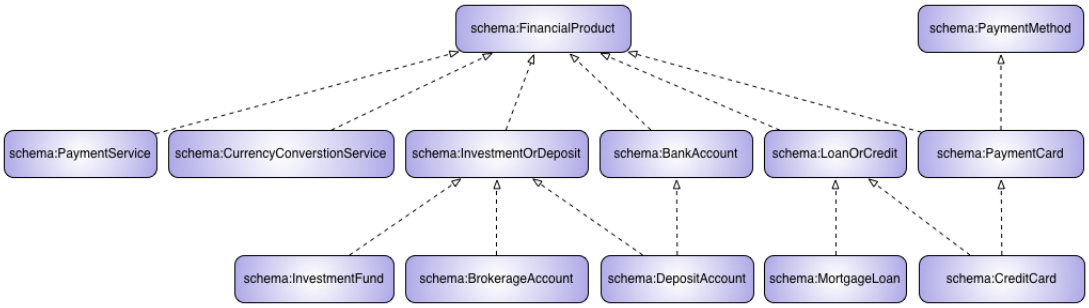

Figure 20: Partial illustration of Golden Ontology for Banking domain from [57]