Final Thesis

A user interface for the ontology merging tool SAMBO

by

Bassam Abdulahad and Georgios Lounis

LITH-IDA-EX-ING--04/020--SE

2004-12-17

Linköpings universitet
Department of Computer and Information Science

Final Thesis

A user interface for the ontology merging tool SAMBO

by

Bassam Abdulahad and Georgios Lounis

LITH-IDA-EX-ING--04/020--SE

2004-12-17

Supervisor: Patrick Lambrix
Examiner: Patrick Lambrix

Abstract

Ontologies have become an important tool for representing data in a structured manner. Merging ontologies allows for the creation of ontologies that later can be composed into larger ontologies as well as for recognizing patterns and similarities between ontologies.
Ontologies are being used nowadays in many areas, including bioinformatics.
In this thesis, we present a desktop version of SAMBO, a system for merging ontologies that are represented in the languages OWL and DAML+OIL. The system has been developed in the programming language JAVA with JDK (Java Development Kit) 1.4.2. The user can open a file locally or from the network and can merge ontologies using suggestions generated by the SAMBO algorithm. SAMBO provides a user-friendly graphical interface, which guides the user through the merging process.

Acknowledgements

This final thesis has been written at the Department of Computer and Information Science, Linköpings universitet as part of the Bachelor of Science in Computer Engineering program.

We would like to thank Associate Professor Patrick Lambrix for his support, advices and comments during this final thesis. It was our pleasure to work with such an excellent supervisor.

Also thanks to He Tan for her help with the algorithms and guidelines during the programming part of this final thesis.

Bassam thanks his parents for their help through the course of life and during the whole education period at the Linköpings universitet.

Georgios thanks his devoted and wonderful fiancée Maria, for all the love and support during all the years of study at Linköpings universitet.

Contents

1. Introduction

This final thesis is a continuous work [2, 4, 5, 6, 7, 8, 9, 16] on ontologies within the University of Linköping. The idea is to create a desktop version similar to the already existing web-based system: "System for Aligning and Merging Bio-Ontologies (SAMBO)". The underlying system is the same but the desktop version in this final thesis implements some of the functionality of the web-based system SAMBO. For instance, in addition to that the suggestions are created by the system, the user can also choose his/her own suggestions to merge manually in the web-based system but not in the desktop system.

1.1 Purpose and method

The purpose of this final thesis is to create a system with a user-friendly interface that merges two ontologies in the languages DAML+OIL and OWL based on the existing system SAMBO [8]. To do this the following had to be considered:

- Get knowledge about ontologies
- Examine the user interface of some existing systems that manage ontologies
- Examine how the merging process is done in SAMBO
- Design and implement a desktop version of SAMBO

2. Background

In the last decades, ontologies have become an important research area. The reason that the interest about ontologies has increased depends mainly on their usefulness within various areas. Ontologies have an important role within bioinformatics, where the ontologies offer a specific vocabulary that can be employed both by humans and applications.

The need for ontologies and systems for creating and merging ontologies is expected to increase. Due to the increased interest in ontologies several systems are introduced and developed. Some existing systems are covered in section 2.3 [4].

2.1 Ontologies

2.1.1 Introduction

The word 'ontology' means 'the study of being' in other words the study of existence and originally came from philosophy. Within computer science, there are many definitions but one definition of ontologies is: "*Ontologies define the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary."* [11]

Within the field of artificial intelligence Gruber defined ontology in the following way: "*An ontology is an explicit specification of a conceptualization* " [3], which can be translated to a formal and legible definition of concepts and relations within a certain domain. Other definitions are presented in [5].

An ontology contains the following components [2]:

- **Concepts.** A concept is defined as a class of entities in a domain, e.g. motor is a concept in a car function-domain. There are two kinds of concepts: **primitive** and **defined**. A primitive concept defines necessary characteristics for an entity so that an entity can belong to a class. A **defined** concept defines both sufficient and necessary characteristics for an entity so that the entity can belong to a class.

- **Relations.** A relation describes the relation between a concept and its characteristics. A relation can be divided into two types: **taxonomic** and **associative** relation. Taxonomic relations organize concepts in sub- and super-concepts, e.g. "is a kind of" relation and "is part of" relation. In a car function-domain, including the concepts motor, wheels, colour, etc. diesel motor "is kind of" motor which in its turn "is part of" car. Associative relations can also be defined in a taxonomy e.g. "*hasColour is a super-relation of hasRedColour, hasBlueColor and hasBlackColor.*"

- **Instances.** An instance is an object, which belongs to a concept, e.g. Mercedes is an instance of the concept car. Most ontologies do not contain instances since ontologies are supposed to be conceptual representations of a domain. An ontology that contains instances is called a *knowledge base (KB).*

- **Functions** are a special form of relation where the n^{th} element in the relation is unique for the *n-1* preceding elements. Functions are usually definied in the following way:

  $F\text{: } C1 \text{ x } C2 \text{ x } \dots \text{ x } C_{n-1} \Rightarrow C_n$

- **Axioms** are facts, which always are true and they are used to restrict concepts and instance values. One type of axiom is relation characteristics.

2.1.2 Bio-Ontologies

Within the bioinformatics area many ontologies have been developed. In this thesis we have used two ontologies for testing: Gene Ontology (GO) and Signal-Ontology.

Database builders joined to create the Gene Ontology Consortium in 1998 [5]. These database builders were developing systems for different organisms and since then the use of ontologies in bioinformatics has grown explosively. The GO describes the rules of the genes and proteins in organisms in a structured, common and dynamic controlled vocabulary [5]. GO ontologies include three independent ontologies: biological process, molecular function and cellular component. Databases containing information about genes and proteins use the GO ontologies as a de facto standard for annotation. Consider the following example that is a part of a GO ontology in figure 1.

```
defense response ; GO:0006952
   hypersensitive response ; GO:0009626
   immune response ; GO:0006955
      acute-phase response ; GO:0006953
      antigen presentation ; GO:0019882
         antigen presentation, endogenous antigen ; GO:0019883
         antigen presentation, exogenous antigen ; GO:0019884
```

Figure 1. A part of GO ontology

In this example **defense response** is the top level concept. An is-a relationship is represented by indentation. Thus, **acute-phase response** is an **immune response,** which in its turn, is a **defense response.**

Signal-Ontology [15] is an ontology for cell signaling systems. It contains knowledge about pathways, networks and causal relationships between modules.

2.1.3 Use of ontologies

Ontologies are used in various ways depending on the information, which is represented in the ontologies. Bio-ontologies are a mixture of domain-oriented ontologies, task-oriented ontologies and generic ontologies. Ontologies are classified into types according to their uses. A description of these types and some examples for bio-ontologies are: [5]

- *Neutral authoring.* In this type application-neutral ontologies are developed in a single language. To use it in multiple target applications, the knowledge must be converted into a different form, e.g. a knowledge base. Reuse and portability of knowledge across platforms and improved maintainability are the benefits of this type. E.g.

different databanks use the GO ontologies and changes in the knowledge need only to be made in one place.

- *Specification.* In this type, software development uses ontology as a basis, e.g. as the definition of the database schema. Search engines can take advantage of this type as it gives extra information. Documentation, maintenance, reliability, sharing and knowledge reuse are the benefits of this type.

- *Common information access.* This type deals with common access to information. Ontologies are used to make information that needs to be shared understandable to multiple applications or humans. The ontology can be used as an interchange format between different applications that can improve interoperability.

- *Search.* This type is used for querying information to improve the quality of the answers and to reduce the search time. A user can use the terms in the ontology as query terms.

Ontologies are used in the area of data mining and knowledge discovery, for separating domain knowledge from application-based knowledge and for validation. Using ontologies forces system developers to make domain assumptions explicit.

2.2 Ontology merging

Ontology merging is the process that will create a single coherent ontology by unifying two or more existing ontologies.

One reason to merge ontologies is to create several and smaller ontologies, which can be merged again with other ontologies or with each other to become a larger ontology. Another reason is the use of multiple ontologies (e.g. public + company) where ontologies contain overlapping information [10].

2.3 Ontology tools

Existing systems can be divided in five types of ontology handling systems depending on their functionalities. These types are: [14]

- *Ontology development tools.* These tools are used for building ontologies, e.g. Protégé-2000, Chimaera, OilEd, Ontolingua, WebOnto, Ontosaurus and KADS.
- *Ontology merge and integration tools.* These tools support users in merging or intergrating ontologies in the same domain, e.g. Protégé-2000 with PROMPT, Chimaera and SAMBO.
- *Ontology evaluation tools.* These tools support ensuring a certain level of quality for the ontologies, eg. OntoAnalyzer, OntoClean and ONE-T.
- *Ontology-based annotation tools.* These tools allow users to insert ontology-based mark-up in web pages.
- *Ontology learning tools.* The tools allow for learning ontologies.
- *Ontology storage and quering tools.* These tools allow for storing and querying ontologies.

In the next section, the tools in table 1 will be shortly described. The tools in table 1 are focused on *Ontology development* and *Ontology merge and integration.*

Ontology development	Ontology merge and integration
Protége-2000	Protége-2000 with PROMPT
OilEd 3.5.5	Chimaera
DAG-Edit 1.414	COBrA 1.0
Chimaera	SAMBO (web-based)
COBrA 1.0	

Table 1. Classification of existing systems

2.3.1 Tools

Protégé-2000 is a system developed by Stanford Medical Informatics for creating, editing and browsing ontologies. The two goals that have driven the design and development of Protégé-2000, are the following:

- Compatibility with other systems for ontology representation
- User-friendly configurable system for ontology extraction.

Protégé is available as free software implemented in Java and should be installed locally. It also has a number of plug-ins, among others PROMPT, which is an algorithm for merging and aligning ontologies [13].
An ontology in Protégé consists of classes, slots, facets and axioms. Classes are concepts in the domain. Slots describe properties or attributes of classes. Facets describe properties of slots and axioms define additional constraints. Both individuals and classes can be instances of classes.

OilEd is a system that was developed at the University of Manchester for creating and editing OIL ontologies and is installed locally. One of the goals of OilEd is to show the use of the DAML+OIL language. OilEd uses the FaCT system [17] which is a description logic system for checking the consequences of the statements in the ontology.

The system is not completely developed and is not intended to create large and complex ontologies and neither created to merge ontologies. However there exist basic tools to create OIL ontologies.

DAG-Edit is a system that was developed by The Gene Ontology Consortium to edit and browse GO ontologies. It is open source software implemented in Java and is installed locally. The relationships that are supported are is-a and part-of.
Dag-Edit allows multiple inheritance i.e. one concept can have more than just one parent. The ontologies are shown in a tree structure.

Chimaera [1] is a web-based ontology system, developed by the Knowledge Systems Laboratory at Stanford University. Chimaera is used to create, browse, edit, merge and diagnose ontologies. This application builds on a system called Ontolingua, written in Common Lisp and has a set of tools to analyze and translate ontologies.

COBrA is an ontology system that allows the user to browse and edit two ontologies. The user can also manually create links between terms in two ontologies. For example, links between tissues in an anatomy and the cell types of the tissues can be recorded and stored: these links are referred to as *mappings*. Mappings can make reference to a third resource - the *reference ontology*. COBrA also allows ontologies to be translated, read, and written into Semantic Web languages including OWL.

COBrA supports reading and writing files in the following formats:

- Gene Ontology (GO) flat file format
- DAG-Edit flat file format
- GO XML/RDF
- GO RDFS
- OWL

COBrA also supports the manipulation and analysis of ontologies, for example:

- Merge ontologies
- Inference
- Validation

2.3.2 SAMBO

SAMBO (web-based) is a web-based ontology system, developed at Linköpings universitet in Sweden. SAMBO supports ontologies, which are represented in DAML+OIL and OWL and it is designed to allow two merging types:

- Suggestion Merge - Suggestions for possible merges are created by SAMBO by comparing the names and synonyms of slots and classes in the two ontologies but the user chooses which suggestions to merge or not.

- Manual Merge - The user can choose which slots and classes to merge without any suggestions from SAMBO and each merged item is added individually to the new ontology. The slot and class definitions that are not merged are copied to the new ontology.

SAMBO provides a number of reasoning services such as consistency, satisfiability and equivalenve checking by using the FaCT reasoner.

2.3.3 Evaluations of ontology tools

There have been two evaluations of ontology tools using biological ontologies. In the first [7], Protégé, Chimaera, OilEd and DAG-Edit were evaluated with respect to criteria such as functionality, data model learning and user interface.

The second evaluation [6, 8] compared Protégé with PROMPT, Chimaera and SAMBO as ontology merge tools.

3. Requirements

The main requirement is to merge two ontologies into one and to save the merged ontology. The system should give suggestions on which slots, classes and relations can be merged between the two ontologies so that the user can choose which suggestions to merge. The user should have the possibility to change a name from the two ontologies of a slot, class or relation suggestion, if the names are equal, before the user merges the current suggestion. The user must change a name from the two ontologies of class suggestion, if the names are equal, before the user rejects the current suggestion.

Other requirements are that the user interface would be user-friendly, each ontology should be shown in a different colour and in a tree structure, the merged ontology would be shown in the colours of the two ontologies so that the user can easily see from which ontology the slot, class or relation came from and in a tree structure.

The system should at least support ontologies in the OWL and DAML+OIL languages and must work both in UNIX and in Microsoft Windows operating systems.

The following features are required:

- Open an ontology
- Save merged ontology
- Close an ontology
- Undo merged ontology
- Redo merged ontology
- Abort the merging process
- View left ontology
- View right ontology
- View bottom ontology
- View all ontologies
- Help
- About SAMBO
- Exit

Quick commands for the different features are also required for the users' convenience.
Before exiting, the system should ask the user to save the merged ontology if the merged ontology has not been saved before.

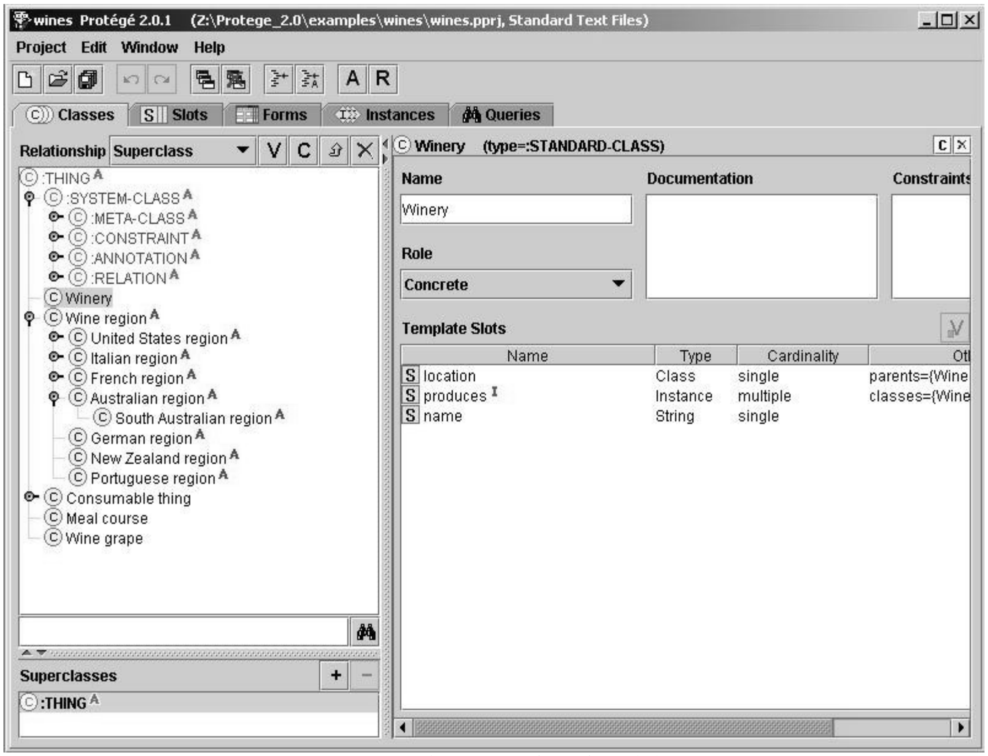
4. Analysis of user interfaces

The analysis of the existing ontology systems, some of them are web-based systems, affected the system to be a hybrid of the existing systems: Protegé-2000 with PROMPT, DAG-Edit 1.414, OilEd 3.5.5, COBrA 1.0 and SAMBO.
These systems were chosen because they have some good features that are important and because they are all non web-based systems. One exception is SAMBO, which is a web-based system that uses the same underlying algorithms as SAMBO the desktop version. Thus the user interface of SAMBO (web-based) was also of interest when developing SAMBO the desktop version.

4.1 Protege-2000 with PROMPT

The Protégé system has a good structure in the user interface (see figure 2). The plugin PROMPT must be included to have the possibility to merge ontologies. The user interface makes Protégé easy to learn and to manage ontologies.
There is a help menu where a user can access help of various types, such as how an ontology can be created and saved.
The user interface consists of different tabs, where each tab shows information about the current ontology component. The lack of quick commands that allows to use Protégé in a smoother way is noticeable. On the other hand there is a toolbar with the most common tools, i.e. open, save etc. One drawback is that only one file can be opened in a project and a user can only work with one project at a time [4].

There are few buttons that accelerate the execution of certain actions, e.g. the ☒ button in the class tab to delete the marked class and its sub components.



Relationship: Superclass

Name: Winery

Role: Concrete

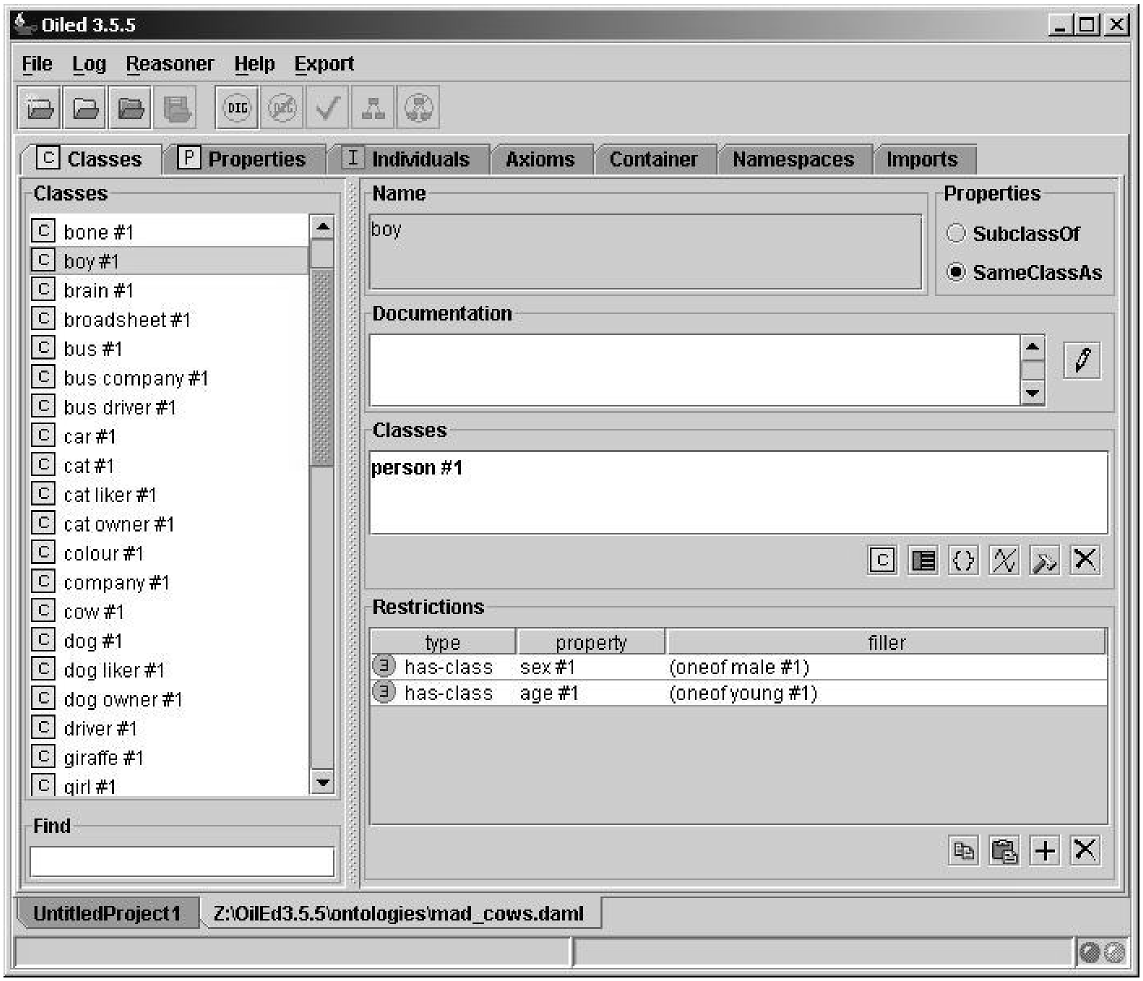
Name	Type	Cardinality	Ot
location	Class	single	parents={Wine
produces I	Instance	multiple	classes={Wine
name	String	single	

Figure 2. The user interface of Protégé

4.2 OilEd 3.5.5

The user interface in this system (see figure 3) is more difficult to use if it is compared to Protégé. The lack of quick commands that allows to use OilEd in a smoother way is noticeable. On the other hand there is a toolbar with the most common tools, i.e. open, save etc. One drawback is that there are no tools to create a new class, the user must right-click with the mouse on the tab class and choose to create a new class instead.
Some advantages that OilEd has are that a user can change the name of a class and a user can work with several projects at the same time.

OilEd has various types of tabs where each tab shows information on the current ontology component [4].
The help function is not built-in OilEd but it is available on the Internet for download at OilEd's homepage.



Classes:
- bone #1
- boy #1
- brain #1
- broadsheet #1
- bus #1
- bus company #1
- bus driver #1
- car #1
- cat #1
- cat liker #1
- cat owner #1
- colour #1
- company #1
- cow #1
- dog #1
- dog liker #1
- dog owner #1
- driver #1
- giraffe #1
- girl #1

Name: boy

Properties: ○ SubclassOf ● SameClassAs

Classes: person #1

type	property	filler
∃ has-class	sex #1	(oneof male #1)
∃ has-class	age #1	(oneof young #1)

Z:\OilEd3.5.5\ontologies\mad_cows.daml

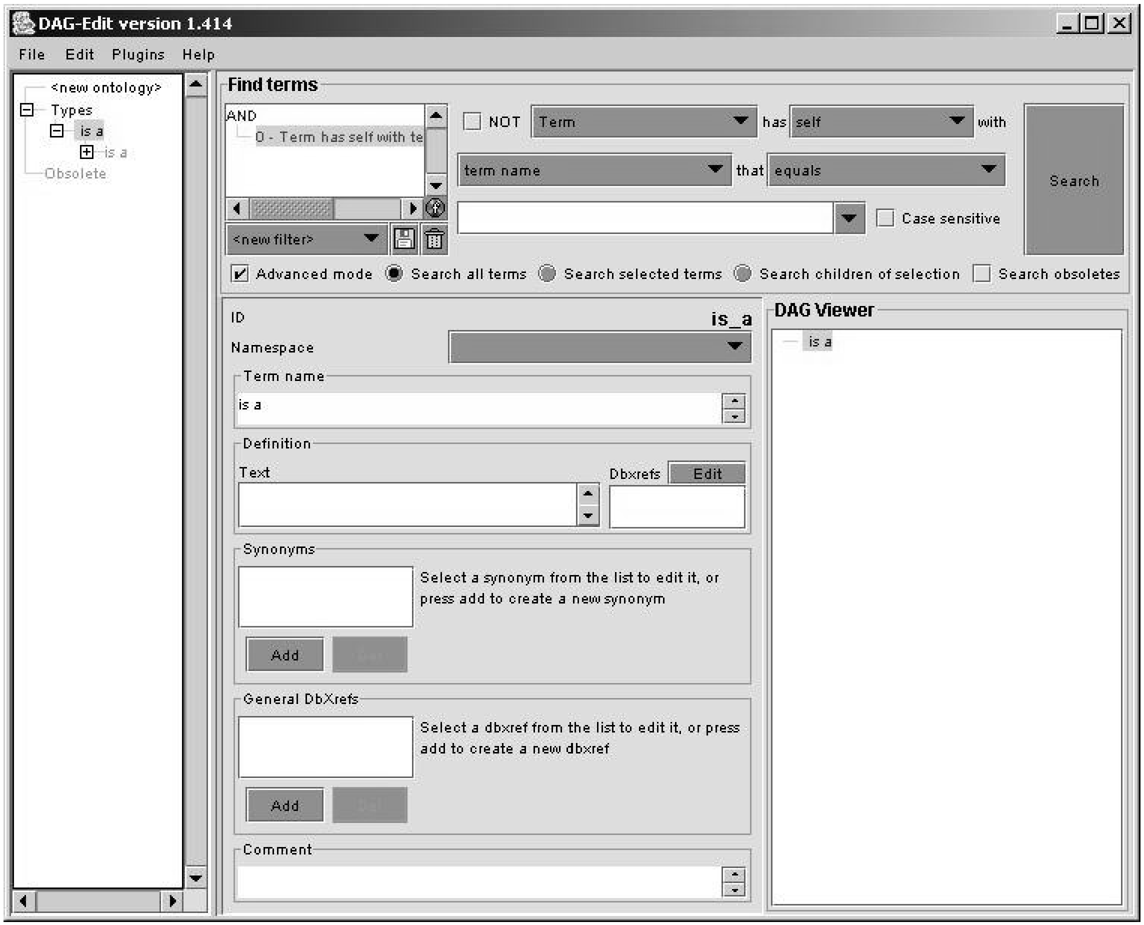
Figure 3. The user interface of OilEd

4.3 DAG-Edit 1.414

The user interface in DAG-Edit (see figure 4) is rather confusing and it is difficult to know how to do things if the user is not familiar with DAG-Edit. One annoying thing is that a user cannot distinguish between ordinary fields and input fields.
The menus are complicated and it takes a while to learn them. Something that really is missing in DAG-Edit is an implementation of a help function. There is only a kind of user guide but it does not help much.

Most of the functions, which are needed when a user works with ontologies, are implemented but other important functions such as that the system asks if the user wants to save before closing the system are not implemented.

DAG-Edit seems to be a little bit unstable, it crashes and a user must restart DAG-Edit when a crash occurs.



Term | has self | with
term name | that equals

☐ NOT
☐ Case sensitive
☑ Advanced mode ● Search all terms ○ Search selected terms ○ Search children of selection ☐ Search obsoletes

ID is_a

Term name: is a

Select a synonym from the list to edit it, or press add to create a new synonym

Select a dbxref from the list to edit it, or press add to create a new dbxref

Figure 4. The user interface of Dag-Edit

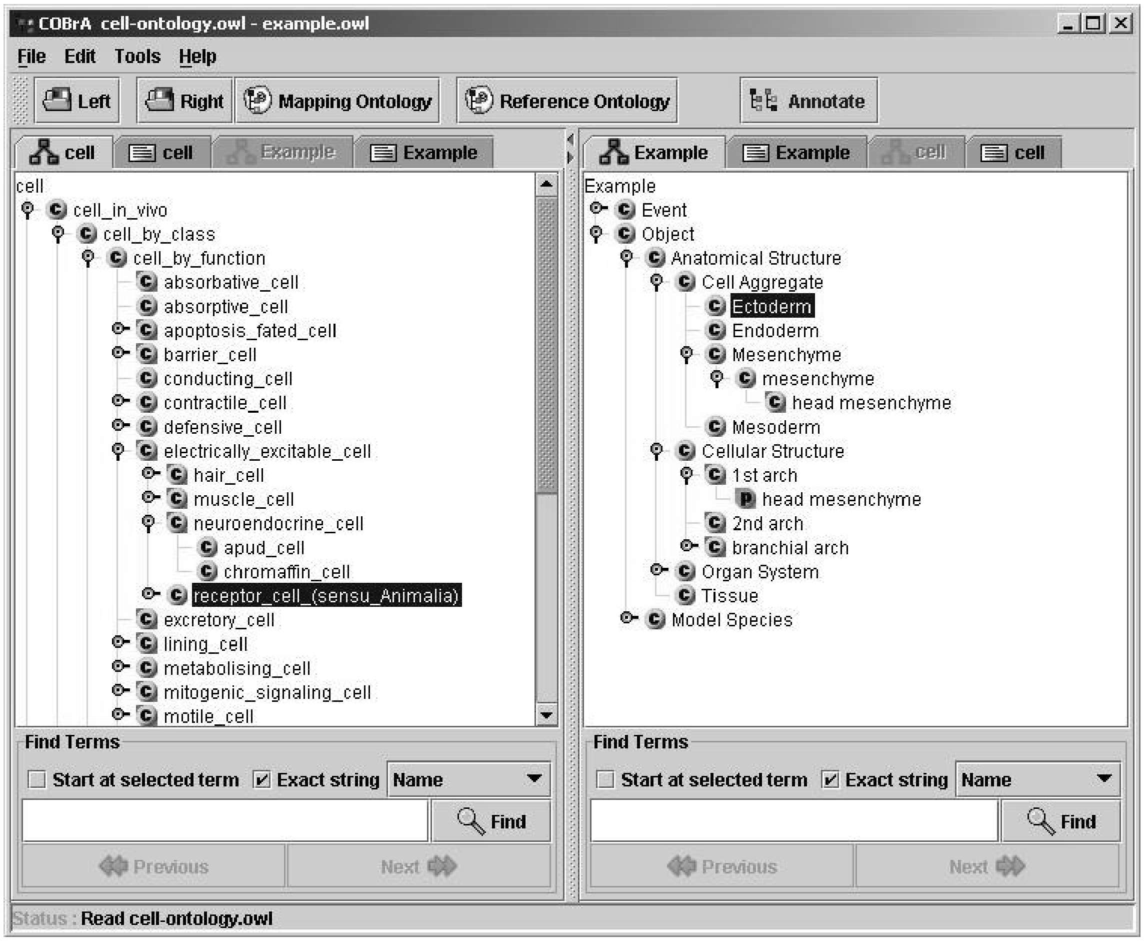
4.4 COBrA 1.0

COBrA has the most user-friendly interface of the evaluated systems (see figure 5). COBrA has an easy and simple layout where two ontologies are shown and can be worked on simultaneously.

However not many functions are implemented, for instance,

- There are no quick commands
- Merge can only be done from the right ontology to the left ontology and the user cannot choose which parts to merge.
- A help function exists but it only consists of comprehensive information.
- The system does not ask if the user wants to save before closing.

In other words COBrA is very easy to use but it is a limited ontology system.



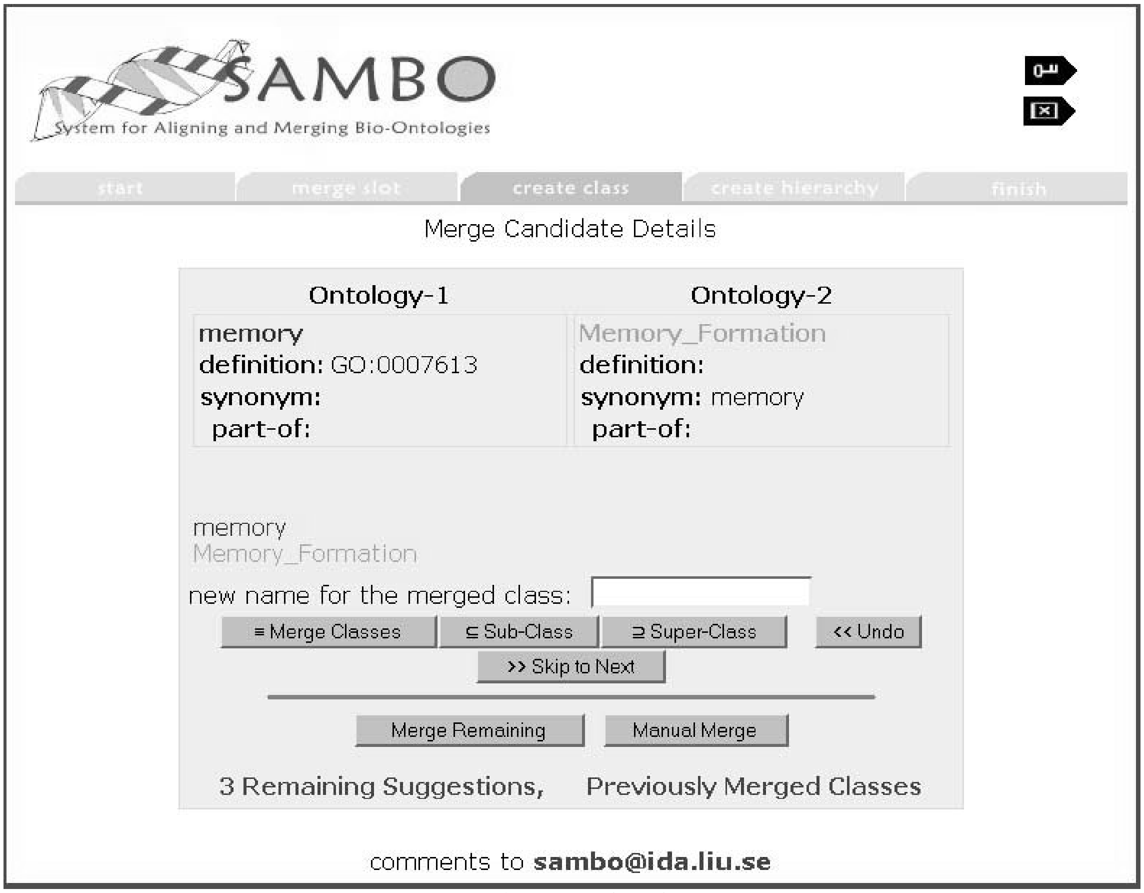
Status : Read cell-ontology.owl

Figure 5. The user interface of COBrA

4.5 SAMBO

The user interface of SAMBO (web-based) is easy to understand and to use. At start the user has to choose which two ontologies to merge and where they are located. Their location could be a file in a web address, from disk (locally) or on the server [17].

The merging process is done in a series of three steps: merging relations, merging concepts and introducing is-a relationships. Each step must be finished before continuing to the next step. At the top of each merge page a tab displays which step SAMBO is currently processing as shown in figure 6 where a class suggestion is suggested.



Merge Candidate Details

Ontology-1	Ontology-2
memory	Memory_Formation
definition: GO:0007613	definition:
synonym:	synonym: memory
part-of:	part-of:

memory
Memory_Formation

new name for the merged class:

3 Remaining Suggestions, Previously Merged Classes

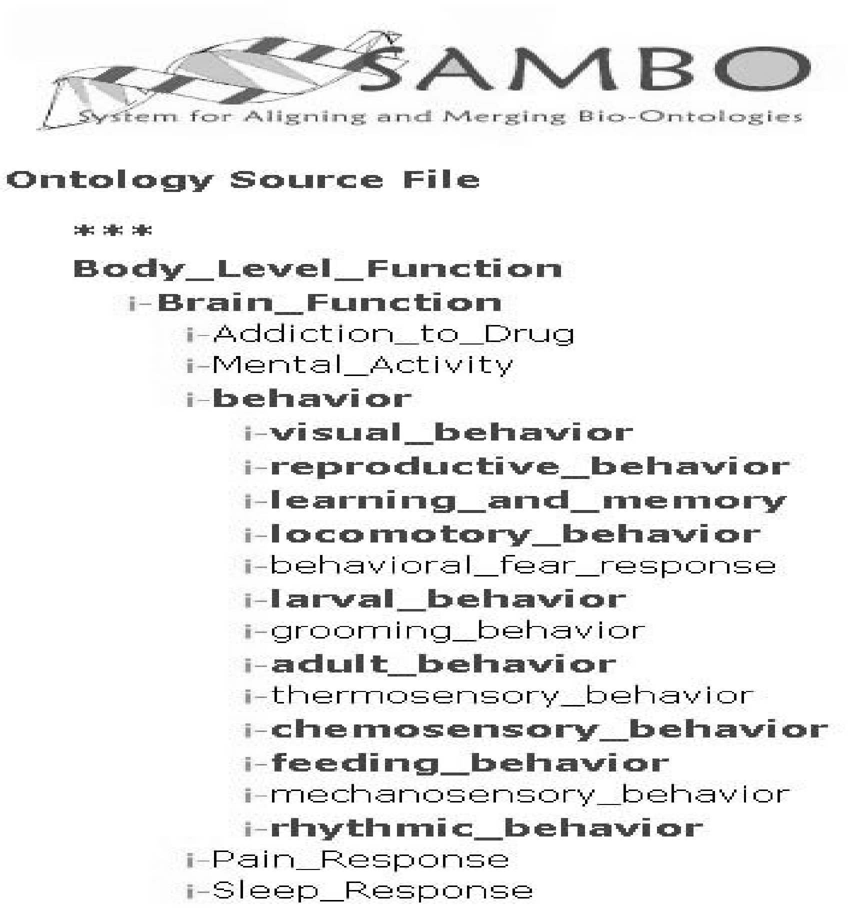
comments to **sambo@ida.liu.se**

Figure 6. The user interface of SAMBO

The user can change the name for a suggestion if the names of the two ontologies match so that a suggestion can have a unique name in the new ontology. There is also a small information bar at the bottom that says how many suggestions remain and the users can always see which suggestions have been merged so far.

The merged ontology is shown in a tree structure, based on “is-a” and “part-of” relations between the classes as shown in figure 7. One drawback could be that the user cannot see the loaded ontologies in a tree structure during the merging process in “Suggestion Merge” mode but the user can see the tree structure in “Manual Merge” mode.

The merged ontology can only be saved by clicking on “File —> Save as” on the web browser during viewing the source file of the merged ontology. The user is not informed of this by the user manual.



SAMBO
System for Aligning and Merging Bio-Ontologies

Ontology Source File

```
***
Body_Level_Function
   i-Brain_Function
      i-Addiction_to_Drug
      i-Mental_Activity
      i-behavior
         i-visual_behavior
         i-reproductive_behavior
         i-learning_and_memory
         i-locomotory_behavior
         i-behavioral_fear_response
         i-larval_behavior
         i-grooming_behavior
         i-adult_behavior
         i-thermosensory_behavior
         i-chemosensory_behavior
         i-feeding_behavior
         i-mechanosensory_behavior
         i-rhythmic_behavior
      i-Pain_Response
      i-Sleep_Response
```

Figure 7. The tree structure of the merged ontology

5. Design

The design of the evaluated existing ontology systems (Protegé-2000 with PROMPT, DAG-Edit 1.414 and OilEd 3.5.5) is rather complicated. The only existing system with a good and not so complicated design is COBrA 1.0.

The design of SAMBO is similar to COBrA's design in that way the ontologies are being shown in a tree structure, in a separate window and using icons in the tree structure, which indicate the type of the node (see figure 8).

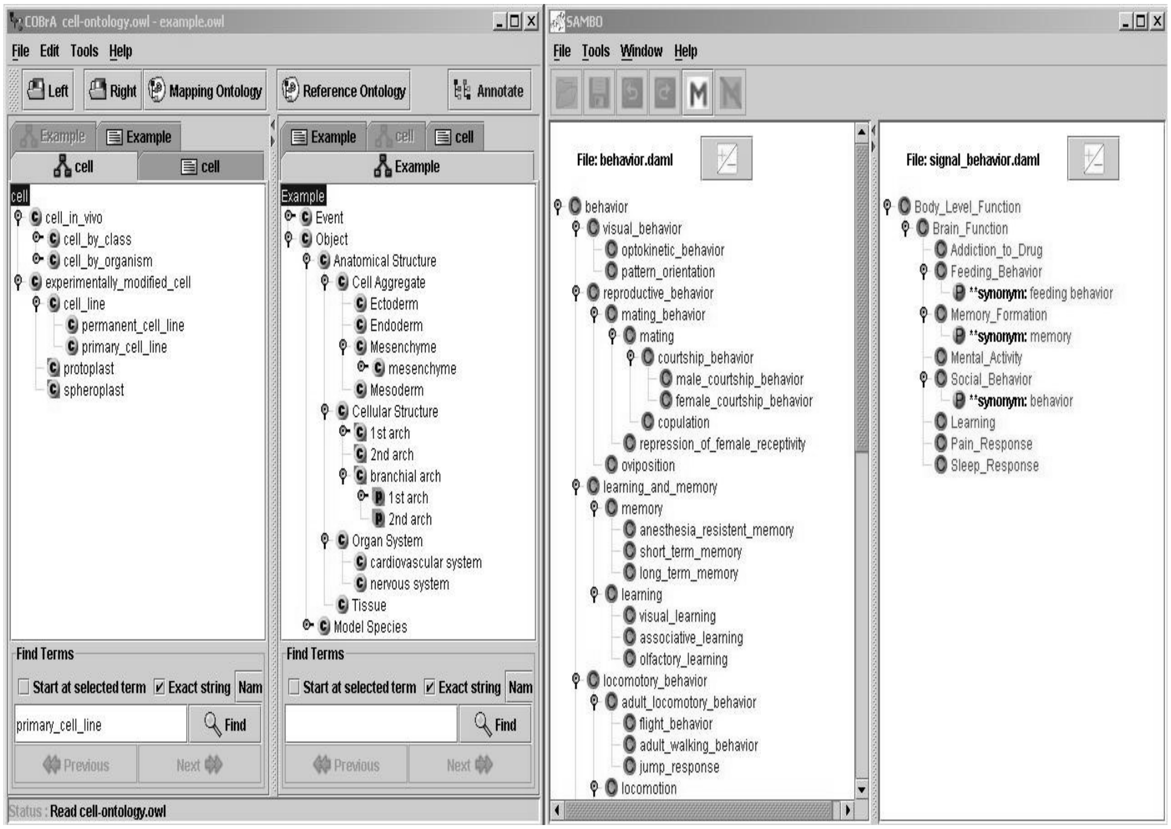


Figure 8. COBrA vs. SAMBO

The idea is that the user can see two ontologies at the same time. This makes the merging process easier to understand and use.

One difference between COBrA and SAMBO in the design is how the merging process and the merged ontology is being shown. In COBrA the user cannot see the merging process nor can he/she choose what to merge while in SAMBO a new window appears at the bottom of the program that shows the merging process and the user can choose what to merge.
Another difference is that in COBrA the merged ontology is shown where the left ontology is shown while in SAMBO the merged ontology is shown where the merging process is shown, that is to say at the bottom window of the program (see figure 14).

6. Implementation

SAMBO was implemented in an environment called IntellJ IDEA 4.0 and in the programming language JAVA with JDK (Java Development Kit) 1.4.2.
The user interface is graphical and the main function is to merge two ontologies.

The ontologies that are supported by SAMBO must be represented in OWL and DAML+OIL.

6.1 Data structures

The algorithms and the test ontologies that were used in SAMBO for handling the ontologies are the same algorithms that were created for the web-based SAMBO that was also developed by Linköpings universitet.

The most important data structures used in SAMBO are DefaultMutableTreeNode and DefaultTreeCellRenderer.

The DefaultMutableTreeNode determines how creating one node at a time and adding it to the tree will create the tree correctly according to the ontology file. When it is done the tree will be shown as a JTree component.

The DefaultTreeCellRenderer determines which node of the tree is a class or a property and sets the icon for the node to C respectively P. It also determines the color for the specific node by checking from which ontology the node comes from. Then it renders the tree accordingly. The class diagram for SAMBO is shown in Appendix A.

6.2 Testing

When the program was finished it was time to test it. A few minor bugs were discovered and were fixed immediately. The main part of the testing was to see if the trees were created and rendered correctly. By using the test ontologies (see section 6.1) and testing SAMBO while opening and merging different ontologies several times, the results of test ontologies and SAMBO could be compared. The test ontologies that were tested are shown in table 2:

	Left ontology	**Right ontology**
Test pair 1	behaviour.daml	signal_behaviour.daml
Test pair 2	defense.owl	signal_defense.owl
Test pair 3	behaviour.daml	signal_behaviour.owl

Table 2: Test ontologies represented in different languages

The conclusion was that SAMBO created and rendered the trees of the ontologies correctly, when an ontology was opened and when two ontologies were merged. The test pair 3 in table 2, where two ontologies were represented in different languages, SAMBO created and rendered the merged tree of the ontology correctly.

7. Conclusion and Future work

7.1 Conclusion

Ontologies have become an important and interesting area within bioinformatics and the need of ontology tools will increase. Thus, the need of applications that help with maintance of and the interaction with these ontologies will increase. SAMBO is an example of a simple desktop application, which enables users to upload, merge and save merged ontologies represented in DAML+OIL and OWL.

The user interface of SAMBO is easy to understand and use because the window is divided into three parts and with three different colours for the ontologies. The first part shows the first ontology in blue, the second part shows the second ontology in red and the third part shows the merging process and after that it shows the merged ontology in green. In the merged ontology, which has "is-a" relations or rejected relations, the colour of these parts is shown in blue or red depending on from which ontology they came from.

All the requirements in section 3 have been implemented in SAMBO but there are many possibilities for additional functionalies. Section 7.2 discusses a number of features, which could improve SAMBO.

7.2 Future work

More advanced functionalities that can be implemented in SAMBO in later versions are listed below:

- Manual merge.
- Update original ontologies on screen with 'Merge' symbol; show already merged items.
- Possibility to show information about concepts in a different window.
- Show information about remaining suggestions.
- A status bar that shows the current status of the system, e.g. opening, saving and merging.

To make SAMBO a complete system for handling ontologies, functionalities for creating and editing ontologies must be implemented.

References

1. Chimaera Software Description URL: www.ksl.stanford.edu/software/chimaera (Acc. 25-07-2004).

2. Edberg, A. (2002) *Merging of ontologies for bioinformatics*, LiTH-IDA-Ex-02/62, Department of computer and information science, Linköpings universitet. In Swedish.

3. Gruber, T. (1993) A translation approach to portable ontology specification, *Knowledge Acquisition,* 5(2):199-220.

4. Habbouche, M. and Pérez, M. (2002) *Evaluation of ontology tools for bioinformatics*, LiTH-IDA-Ing-Ex-02/10, Department of computer and information science, Linköpings universitet. In Swedish.

5. Lambrix, P., *Ontologies in Bioinformatics and Systems Biology*, chapter in Dubitzky, W. and Azuaje, F. (eds.) Artificial Intelligence Methods and Tools for Systems Biology, pp 129-146, Kluwer, 2004. ISBN: 1-4020-2859-8.

6. Lambrix, P., Edberg, A., *Evaluation of ontology merging tools in bioinformatics*, Pacific Symposium on Biocomputing - PSB03, 8:589-600, Kauai, Hawaii, USA, 2003

7. Lambrix, P., Habbouche, M., Pérez, M., *Evaluation of ontology development tools for bioinformatics*, Bioinformatics, 19(12):1564-1571, 2003.

8. Lambrix, P., Tan, H., *Merging DAML+OIL Ontologies*, Proceedings of the Sixth International Baltic Conference on Databases and Information Systems, pp 425-435, Riga, Latvia, 2004.

9. Manis, C. (2002) *Ontology Merge, a web application for merging ontlologies*, LiTH-IDA-Ing-Ex-02/32, Department of computer and information science, Linköpings universitet.

10. Merging Bio-Ontologies URL: www.rewerse.net/A2/ppt/LinkoepingPPT.ppt. (Acc. 20-07-2004)

11. Neches, R., Fikes, R., Finin, T., Gruber, T., Senator, T. and Swartout, W. (1991) Enabling technology for knowledge engineering, *AI Magazine,* 12(3):36-56.

12. Noy, N. F. and Musen, M. (2000) PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment, *Proceedings of Seventeenth National Conference on Artificial Intelligence - AAAI00*. pp. 450–455, Austin, Texas, USA.

13. Noy, N. F., Fergerson, R. W. and Musen, M. (2000) The knowledge model of Protégé-2000: combining interoperability and flexibility. In: *Proceedings of the 2th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2000)*, Juan-les-pins, France, 2000.

14. OntoWeb Consortium, Deliverable 1.3, *A survey on ontology tools*, 2002

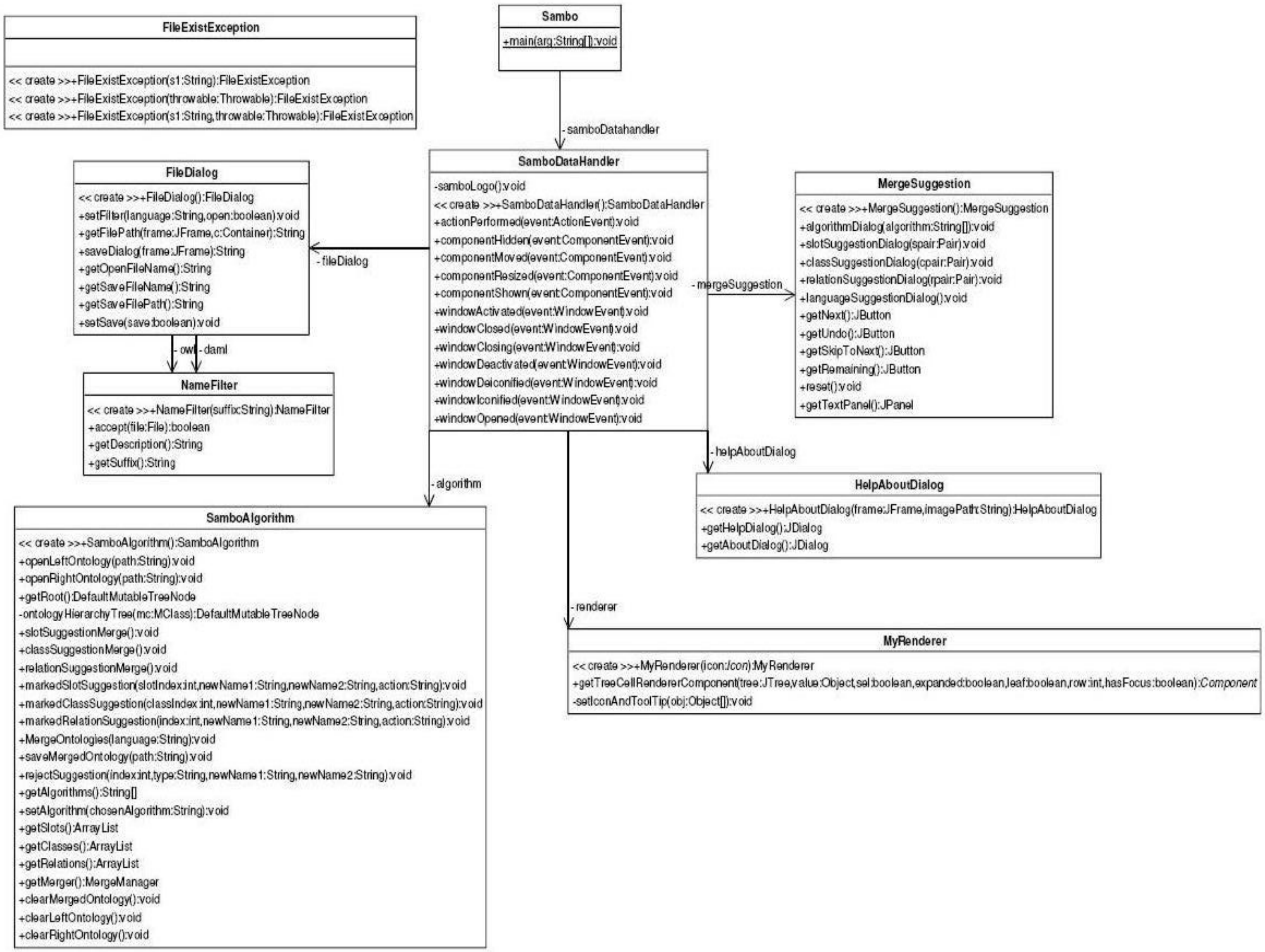
15. Signal-Ontology URL: http://ontology.ims.u-tokyo.ac.jp/signalontology. (Acc. 25-11-2004).

16. Tan, H. (2003) *Merging DAML+OIL Bio-Ontologies*, LiTH-IDA-Ex-03/51--SE, Department of computer and information science, Linköpings universitet.

17. The FaCT System URL: www.cs.man.ac.uk/~horrocks/FaCT. (Acc. 21-07-2004).

Appendices

Appendix A: Class Diagram for SAMBO


FileExistException
<< create >>+FileExistException(s1:String):FileExistException
<< create >>+FileExistException(throwable:Throwable):FileExistException
<< create >>+FileExistException(s1:String,throwable:Throwable):FileExistException
Sambo
+main(arg:String[]):void
- samboDatahandler
SamboDataHandler
-samboLogo():void
<< create >>+SamboDataHandler():SamboDataHandler
+actionPerformed(event:ActionEvent):void
+componentHidden(event:ComponentEvent):void
+componentMoved(event:ComponentEvent):void
+componentResized(event:ComponentEvent):void
+componentShown(event:ComponentEvent):void
+windowActivated(event:WindowEvent):void
+windowClosed(event:WindowEvent):void
+windowClosing(event:WindowEvent):void
+windowDeactivated(event:WindowEvent):void
+windowDeiconified(event:WindowEvent):void
+windowIconified(event:WindowEvent):void
+windowOpened(event:WindowEvent):void
FileDialog
<< create >>+FileDialog():FileDialog
+setFilter(language:String,open:boolean):void
+getFilePath(frame:JFrame,c:Container):String
+saveDialog(frame:JFrame):String
+getOpenFileName():String
+getSaveFileName():String
+getSaveFilePath():String
+setSave(save:boolean):void
- fileDialog
- owl
- daml
NameFilter
<< create >>+NameFilter(suffix:String):NameFilter
+accept(file:File):boolean
+getDescription():String
+getSuffix():String
- mergeSuggestion
MergeSuggestion
<< create >>+MergeSuggestion():MergeSuggestion
+algorithmDialog(algorithm:String[]):void
+slotSuggestionDialog(spair:Pair):void
+classSuggestionDialog(cpair:Pair):void
+relationSuggestionDialog(rpair:Pair):void
+languageSuggestionDialog():void
+getNext():JButton
+getUndo():JButton
+getSkipToNext():JButton
+getRemaining():JButton
+reset():void
+getTextPanel():JPanel
- helpAboutDialog
HelpAboutDialog
<< create >>+HelpAboutDialog(frame:JFrame,imagePath:String):HelpAboutDialog
+getHelpDialog():JDialog
+getAboutDialog():JDialog
- algorithm
SamboAlgorithm
<< create >>+SamboAlgorithm():SamboAlgorithm
+openLeftOntology(path:String):void
+openRightOntology(path:String):void
+getRoot():DefaultMutableTreeNode
-ontologyHierarchyTree(mc:MClass):DefaultMutableTreeNode
+slotSuggestionMerge():void
+classSuggestionMerge():void
+relationSuggestionMerge():void
+markedSlotSuggestion(slotIndex:int,newName1:String,newName2:String,action:String):void
+markedClassSuggestion(classIndex:int,newName1:String,newName2:String,action:String):void
+markedRelationSuggestion(index:int,newName1:String,newName2:String,action:String):void
+MergeOntologies(language:String):void
+saveMergedOntology(path:String):void
+rejectSuggestion(index:int,type:String,newName1:String,newName2:String):void
+getAlgorithms():String[]
+setAlgorithm(chosenAlgorithm:String):void
+getSlots():ArrayList
+getClasses():ArrayList
+getRelations():ArrayList
+getMerger():MergeManager
+clearMergedOntology():void
+clearLeftOntology():void
+clearRightOntology():void
- renderer
MyRenderer
<< create >>+MyRenderer(icon:Icon):MyRenderer
+getTreeCellRendererComponent(tree:JTree,value:Object,sel:boolean,expanded:boolean,leaf:boolean,row:int,hasFocus:boolean):Component
-setIconAndToolTip(obj:Object[]):void

Appendix B: JavaDocs

	Package
Class	**Tree**

Index **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class Sambo

java.lang.Object
  └ **Sambo**

---

public class **Sambo**
extends java.lang.Object

The main class in this project.

Constructor Summary

Sambo()

Method Summary

static void	main(java.lang.String[] arg)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Sambo

public **Sambo**()

Method Detail

main

public static void **main**(java.lang.String[] arg)

	Package
Class	**Tree**

Index **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Package

Class Tree Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES All Classes

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class SamboDataHandler

```
java.lang.Object
 └ java.awt.Component
    └ java.awt.Container
       └ java.awt.Window
          └ java.awt.Frame
             └ javax.swing.JFrame
                └ SamboDataHandler
```

All Implemented Interfaces:

javax.accessibility.Accessible, java.awt.event.ActionListener, java.awt.event.ComponentListener, java.util.EventListener, java.awt.image.ImageObserver, java.awt.MenuContainer, javax.swing.RootPaneContainer, java.io.Serializable, javax.swing.WindowConstants, java.awt.event.WindowListener

public class **SamboDataHandler**
extends javax.swing.JFrame
implements java.awt.event.ActionListener, java.awt.event.ComponentListener, java.awt.event.WindowListener

SamboDataHandler creates the window of the program and handles all actions generated by the user.

See Also:

Serialized Form

Nested Class Summary

Nested classes inherited from class javax.swing.JFrame
javax.swing.JFrame.AccessibleJFrame

Nested classes inherited from class java.awt.Frame
java.awt.Frame.AccessibleAWTFrame

Nested classes inherited from class java.awt.Window
java.awt.Window.AccessibleAWTWindow

Nested classes inherited from class java.awt.Container
java.awt.Container.AccessibleAWTContainer

Nested classes inherited from class java.awt.Component
java.awt.Component.AccessibleAWTComponent, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy

Field Summary

Fields inherited from class javax.swing.JFrame
accessibleContext, EXIT_ON_CLOSE, rootPane, rootPaneCheckingEnabled

Fields inherited from class java.awt.Frame
CROSSHAIR_CURSOR, DEFAULT_CURSOR, E_RESIZE_CURSOR, HAND_CURSOR, ICONIFIED, MAXIMIZED_BOTH, MAXIMIZED_HORIZ, MAXIMIZED_VERT, MOVE_CURSOR, N_RESIZE_CURSOR, NE_RESIZE_CURSOR, NORMAL, NW_RESIZE_CURSOR, S_RESIZE_CURSOR, SE_RESIZE_CURSOR, SW_RESIZE_CURSOR, TEXT_CURSOR, W_RESIZE_CURSOR, WAIT_CURSOR

Fields inherited from class java.awt.Component
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface javax.swing.WindowConstants
DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, HIDE_ON_CLOSE

Fields inherited from interface java.awt.image.ImageObserver
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

SamboDataHandler() The Constructor of the class.

Method Summary

void	**actionPerformed**(java.awt.event.ActionEvent event) Performs the acquired action from the user.
void	**componentHidden**(java.awt.event.ComponentEvent event)
void	**componentMoved**(java.awt.event.ComponentEvent event)
void	**componentResized**(java.awt.event.ComponentEvent event) Repaints the whole window when the window is resized.
void	**componentShown**(java.awt.event.ComponentEvent event)
void	**windowActivated**(java.awt.event.WindowEvent event)
void	**windowClosed**(java.awt.event.WindowEvent event)
void	**windowClosing**(java.awt.event.WindowEvent event) Checks if the merged ontology is saved before exiting the program.
void	**windowDeactivated**(java.awt.event.WindowEvent event)

void	**windowDeiconified**(java.awt.event.WindowEvent event)
void	**windowIconified**(java.awt.event.WindowEvent event)
void	**windowOpened**(java.awt.event.WindowEvent event)

Methods inherited from class javax.swing.JFrame
addImpl, createRootPane, frameInit, getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getJMenuBar, getLayeredPane, getRootPane, isDefaultLookAndFeelDecorated, isRootPaneCheckingEnabled, paramString, processWindowEvent, remove, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setJMenuBar, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, update

Methods inherited from class java.awt.Frame
addNotify, finalize, getCursorType, getExtendedState, getFrames, getIconImage, getMaximizedBounds, getMenuBar, getState, getTitle, isResizable, isUndecorated, remove, removeNotify, setCursor, setExtendedState, setIconImage, setMaximizedBounds, setMenuBar, setResizable, setState, setTitle, setUndecorated

Methods inherited from class java.awt.Window
addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, dispose, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getInputContext, getListeners, getLocale, getMostRecentFocusOwner, getOwnedWindows, getOwner, getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners, getWindowStateListeners, hide, isActive, isFocusableWindow, isFocusCycleRoot, isFocused, isShowing, pack, postEvent, processEvent, processWindowFocusEvent, processWindowStateEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, setCursor, setFocusableWindowState, setFocusCycleRoot, setLocationRelativeTo, show, toBack, toFront

Methods inherited from class java.awt.Container
add, add, add, add, add, addContainerListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getContainerListeners, getFocusTraversalPolicy, getInsets, getLayout, getMaximumSize, getMinimumSize, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, processContainerEvent, remove, removeAll, removeContainerListener, setFocusTraversalKeys, setFocusTraversalPolicy, setFont, transferFocusBackward, transferFocusDownCycle, validate, validateTree

Methods inherited from class java.awt.Component
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, createVolatileImage,

createVolatileImage, disable, disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, getBackground, getBounds, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocation, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMouseWheelListeners, getName, getParent, getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isOpaque, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, prepareImage, prepareImage, printAll, processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, processMouseWheelEvent, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setName, setSize, setSize, setVisible, show, size, toString, transferFocus, transferFocusUpCycle

Methods inherited from class java.lang.Object
clone, equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Methods inherited from interface java.awt.MenuContainer
getFont, postEvent

Constructor Detail

SamboDataHandler

public **SamboDataHandler**()

The Constructor of the class.

Method Detail

actionPerformed

public void **actionPerformed**(java.awt.event.ActionEvent event)

Performs the acquired action from the user. The acquired action will be one of the following: Undo slot suggestion, Undo class suggestion, Undo relation suggestion, Begin to merge, Process slot suggestion, Process class suggestion, Process relation suggestion, Merge, Reject slot suggestion, Reject class suggestion, Reject relation suggestion, Merge remaining suggestions, Skip remaining suggestions, Expand left tree, Expand right tree and Expand merged tree.

Specified by:

actionPerformed in interface java.awt.event.ActionListener

---

componentHidden

public void **componentHidden**(java.awt.event.ComponentEvent event)

Specified by:
componentHidden in interface java.awt.event.ComponentListener

---

componentMoved

public void **componentMoved**(java.awt.event.ComponentEvent event)

Specified by:
componentMoved in interface java.awt.event.ComponentListener

---

componentResized

public void **componentResized**(java.awt.event.ComponentEvent event)

Repaints the whole window when the window is resized.

Specified by:
componentResized in interface java.awt.event.ComponentListener

---

componentShown

public void **componentShown**(java.awt.event.ComponentEvent event)

Specified by:
componentShown in interface java.awt.event.ComponentListener

---

windowActivated

public void **windowActivated**(java.awt.event.WindowEvent event)

Specified by:
windowActivated in interface java.awt.event.WindowListener

---

windowClosed

public void **windowClosed**(java.awt.event.WindowEvent event)

Specified by:
windowClosed in interface java.awt.event.WindowListener

---

windowClosing

public void **windowClosing**(java.awt.event.WindowEvent event)

Checks if the merged ontology is saved before exiting the program. If not saved then a dialog box asking to save the merged ontology will appear.

Specified by:
windowClosing in interface java.awt.event.WindowListener

---

windowDeactivated

public void **windowDeactivated**(java.awt.event.WindowEvent event)

Specified by:
windowDeactivated in interface java.awt.event.WindowListener

---

windowDeiconified

public void **windowDeiconified**(java.awt.event.WindowEvent event)

Specified by:
windowDeiconified in interface java.awt.event.WindowListener

---

windowIconified

public void **windowIconified**(java.awt.event.WindowEvent event)

Specified by:
windowIconified in interface java.awt.event.WindowListener

---

windowOpened

public void **windowOpened**(java.awt.event.WindowEvent event)

Specified by:
windowOpened in interface java.awt.event.WindowListener

Package
Class **Tree** **Index** **Help**

PREV CLASS NEXT CLASS **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Package

Class **Tree** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class SamboAlgorithm

java.lang.Object
└ **SamboAlgorithm**

public class **SamboAlgorithm**
extends java.lang.Object

The class SamboAlgorithm implements all algorithms to open, merge, suggestions, save, etc.

Constructor Summary

SamboAlgorithm() The constructor of the class.

Method Summary

void	**classSuggestionMerge**() Gets and puts all the class suggestions to merge, into a list.
void	**clearLeftOntology**() Clears the memory of the first ontology so that a new ontology can be opened.
void	**clearMergedOntology**() Clears the memory of the merged ontology so that a new merged ontology can be created.
void	**clearRightOntology**() Clears the memory of the second ontology so that a new ontology can be opened.
java.lang.String[]	**getAlgorithms**() Returns a list of all the algorithms that can be chosen as the algorithm that finds the suggestions.
java.util.ArrayList	**getClasses**() Returns a list of all class suggestions.
sambo.Merger.MergeManager	**getMerger**() Returns the MergeManager that is used.
java.util.ArrayList	**getRelations**() Returns a list of all relation suggestions.
javax.swing.tree.DefaultMutableTreeNode	**getRoot**() Returns the root node with all of its subnodes and leafs of the current tree.
java.util.ArrayList	**getSlots**() Returns a list of all slot suggestions.
void	**markedClassSuggestion**(int classIndex,

	java.lang.String newName1, java.lang.String newName2, java.lang.String action) Checks if the names of the current class suggestion has been changed, which action to process and then processes the current class suggestion.
void	**markedRelationSuggestion**(int index, java.lang.String newName1, java.lang.String newName2, java.lang.String action) Checks if the names of the current relation suggestion has been changed, which action to process and then processes the current relation suggestion.
void	**markedSlotSuggestion**(int slotIndex, java.lang.String newName1, java.lang.String newName2, java.lang.String action) Checks if the names of the current slot suggestion has been changed, which action to process and then processes the current slot suggestion.
void	**MergeOntologies**(java.lang.String language) Merges the two ontologies acoording to the chosen suggestions.
void	**openLeftOntology**(java.lang.String path) Opens the first ontology.
void	**openRightOntology**(java.lang.String path) Opens the second ontology.
void	**rejectSuggestion**(int index, java.lang.String type) Rejects the current suggestion.
void	**relationSuggestionMerge**() Gets and puts all the relation suggestions to merge, into a list.
void	**saveMergedOntology**(java.lang.String path) Saves the merged ontology.
void	**setAlgorithm**(java.lang.String chosenAlgorithm) Sets the algorithm that has been chosen to find the suggestions.
void	**slotSuggestionMerge**() Gets and puts all the slot suggestions to merge, into a list.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

SamboAlgorithm

public **SamboAlgorithm**()

The constructor of the class.

Method Detail

classSuggestionMerge

public void **classSuggestionMerge**()

Gets and puts all the class suggestions to merge, into a list.

---

clearLeftOntology

public void **clearLeftOntology**()

Clears the memory of the first ontology so that a new ontology can be opened.

---

clearMergedOntology

public void **clearMergedOntology**()

Clears the memory of the merged ontology so that a new merged ontology can be created.

---

clearRightOntology

public void **clearRightOntology**()

Clears the memory of the second ontology so that a new ontology can be opened.

---

getAlgorithms

public java.lang.String[] **getAlgorithms**()

Returns a list of all the algorithms that can be chosen as the algorithm that finds the suggestions.

---

getClasses

public java.util.ArrayList **getClasses**()

Returns a list of all class suggestions.

---

getMerger

public sambo.Merger.MergeManager **getMerger**()

Returns the MergeManager that is used.

---

getRelations

public java.util.ArrayList **getRelations**()

Returns a list of all relation suggestions.

---

getRoot

public javax.swing.tree.DefaultMutableTreeNode **getRoot**()

Returns the root node with all of its subnodes and leafs of the current tree.

---

getSlots

public java.util.ArrayList **getSlots**()

Returns a list of all slot suggestions.

---

markedClassSuggestion

public void **markedClassSuggestion**(int classIndex,
                                   java.lang.String newName1,
                                   java.lang.String newName2,

```
                        java.lang.String action)
```

Checks if the names of the current class suggestion has been changed, which action to process and then processes the current class suggestion.

markedRelationSuggestion

```
public void markedRelationSuggestion(int index,
                                     java.lang.String newName1,
                                     java.lang.String newName2,
                                     java.lang.String action)
```

Checks if the names of the current relation suggestion has been changed, which action to process and then processes the current relation suggestion.

markedSlotSuggestion

```
public void markedSlotSuggestion(int slotIndex,
                                 java.lang.String newName1,
                                 java.lang.String newName2,
                                 java.lang.String action)
```

Checks if the names of the current slot suggestion has been changed, which action to process and then processes the current slot suggestion.

MergeOntologies

```
public void MergeOntologies(java.lang.String language)
```

Merges the two ontologies acoording to the chosen suggestions. The new merged ontology will be created in the language specified by the parameter "language".

openLeftOntology

```
public void openLeftOntology(java.lang.String path)
```

Opens the first ontology. It is called "openLeftOntoly" because the first ontology is always shown at the left side in the system.

openRightOntology

```
public void openRightOntology(java.lang.String path)
```

Opens the second ontology. It is called "openRightOntoly" because the second ontology is always shown at the right side in the system.

rejectSuggestion

```
public void rejectSuggestion(int index,
                             java.lang.String type)
```

Rejects the current suggestion.

relationSuggestionMerge

```
public void relationSuggestionMerge()
```

Gets and puts all the relation suggestions to merge, into a list.

saveMergedOntology

```
public void saveMergedOntology(java.lang.String path)
```

Saves the merged ontology.

setAlgorithm

public void **setAlgorithm**(java.lang.String chosenAlgorithm)

Sets the algorithm that has been chosen to find the suggestions.

---

slotSuggestionMerge

public void **slotSuggestionMerge**()

Gets and puts all the slot suggestions to merge, into a list.

Package
Class **Tree** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Package
Class Tree Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES All Classes
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class NameFilter

```
java.lang.Object
 └ javax.swing.filechooser.FileFilter
    └ NameFilter
```

public class **NameFilter**
extends javax.swing.filechooser.FileFilter

The class NameFilter creates suffixes that will only be choosable when saving or opening files.

Constructor Summary
NameFilter(java.lang.String suffix) The constructor of the class.

Method Summary

boolean	**accept**(java.io.File file) Returns TRUE if the file is a directory or a file with the right suffix else returns FALSE.
java.lang.String	**getDescription**() Returns the description of the file.
java.lang.String	**getSuffix**() Returns the suffix of the file.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

NameFilter

public **NameFilter**(java.lang.String suffix)

The constructor of the class.

Method Detail

accept

public boolean **accept**(java.io.File file)

Returns TRUE if the file is a directory or a file with the right suffix else returns FALSE.

getDescription

public java.lang.String **getDescription**()

Returns the description of the file.

---

getSuffix

public java.lang.String **getSuffix**()

Returns the suffix of the file.

Package
Class **Tree** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Package
Class Tree Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES All Classes

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class MyRenderer

```
java.lang.Object
 └ java.awt.Component
     └ java.awt.Container
         └ javax.swing.JComponent
             └ javax.swing.JLabel
                 └ javax.swing.tree.DefaultTreeCellRenderer
                     └ MyRenderer
```

All Implemented Interfaces:

javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable, javax.swing.SwingConstants, javax.swing.tree.TreeCellRenderer

public class **MyRenderer**
extends javax.swing.tree.DefaultTreeCellRenderer

The class MyRenderer renders the trees that shows the ontologies with colors. It also puts the two icons of CLASS and PROPERTY.

See Also:

Serialized Form

Nested Class Summary

Nested classes inherited from class javax.swing.JLabel
javax.swing.JLabel.AccessibleJLabel

Nested classes inherited from class javax.swing.JComponent
javax.swing.JComponent.AccessibleJComponent

Nested classes inherited from class java.awt.Container
java.awt.Container.AccessibleAWTContainer

Nested classes inherited from class java.awt.Component
java.awt.Component.AccessibleAWTComponent, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy

Field Summary

Fields inherited from class javax.swing.tree.DefaultTreeCellRenderer
backgroundNonSelectionColor, backgroundSelectionColor, borderSelectionColor, closedIcon, hasFocus, leafIcon, openIcon, selected, textNonSelectionColor, textSelectionColor

Fields inherited from class javax.swing.JLabel
labelFor

Fields inherited from class javax.swing.JComponent
accessibleContext, listenerList, TOOL_TIP_TEXT_KEY, ui, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW

Fields inherited from class java.awt.Component
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface javax.swing.SwingConstants
BOTTOM, CENTER, EAST, HORIZONTAL, LEADING, LEFT, NEXT, NORTH, NORTH_EAST, NORTH_WEST, PREVIOUS, RIGHT, SOUTH, SOUTH_EAST, SOUTH_WEST, TOP, TRAILING, VERTICAL, WEST

Fields inherited from interface java.awt.image.ImageObserver
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

MyRenderer(javax.swing.Icon icon) The constructor of the class.	

Method Summary

java.awt.Component	**getTreeCellRendererComponent**(javax.swing.JTree tree, java.lang.Object value, boolean sel, boolean expanded, boolean leaf, int row, boolean hasFocus) Returns the current component in the tree which has been rendered.

Methods inherited from class javax.swing.tree.DefaultTreeCellRenderer
firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getBackgroundNonSelectionColor, getBackgroundSelectionColor, getBorderSelectionColor, getClosedIcon, getDefaultClosedIcon, getDefaultLeafIcon, getDefaultOpenIcon, getFont, getLeafIcon, getOpenIcon, getPreferredSize, getTextNonSelectionColor, getTextSelectionColor, paint, repaint, repaint, revalidate, setBackground, setBackgroundNonSelectionColor, setBackgroundSelectionColor, setBorderSelectionColor, setClosedIcon, setFont, setLeafIcon, setOpenIcon, setTextNonSelectionColor, setTextSelectionColor, validate

Methods inherited from class javax.swing.JLabel
checkHorizontalKey, checkVerticalKey, getAccessibleContext, getDisabledIcon, getDisplayedMnemonic, getDisplayedMnemonicIndex, getHorizontalAlignment, getHorizontalTextPosition, getIcon, getIconTextGap, getLabelFor, getText, getUI, getUIClassID, getVerticalAlignment, getVerticalTextPosition, imageUpdate, paramString, setDisabledIcon, setDisplayedMnemonic, setDisplayedMnemonic, setDisplayedMnemonicIndex, setHorizontalAlignment,

setHorizontalTextPosition, setIcon, setIconTextGap, setLabelFor, setText, setUI, setVerticalAlignment, setVerticalTextPosition, updateUI

Methods inherited from class javax.swing.JComponent
addAncestorListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentGraphics, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getGraphics, getHeight, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPropertyChangeListeners, getPropertyChangeListeners, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getWidth, getVisibleRect, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isPreferredSizeSet, isRequestFocusEnabled, isValidateRoot, paintBorder, paintChildren, paintComponent, paintImmediately, paintImmediately, print, printAll, printBorder, printChildren, printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, removeVetoableChangeListener, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resetKeyboardActions, reshape, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBorder, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setForeground, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update

Methods inherited from class java.awt.Container
add, add, add, add, add, addContainerListener, addImpl, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent, processEvent, remove, remove, removeAll, removeContainerListener, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setLayout, transferFocusBackward, transferFocusDownCycle, validateTree

Methods inherited from class java.awt.Component
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents, contains, createImage, createImage, createVolatileImage, createVolatileImage, disableEvents, dispatchEvent, enable, enableEvents, enableInputMethods, getBackground, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeysEnabled, getFontMetrics, getForeground, getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMouseWheelListeners, getName, getParent, getPeer, getSize, getToolkit, getTreeLock, gotFocus, handleEvent, hasFocus, hide, inside, isBackgroundSet, isCursorSet,

isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, postEvent, prepareImage, prepareImage, processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processMouseEvent, processMouseWheelEvent, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, repaint, repaint, repaint, resize, resize, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale, setLocation, setLocation, setName, setSize, setSize, show, show, size, toString, transferFocus, transferFocusUpCycle

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

MyRenderer

```
public MyRenderer(javax.swing.Icon icon)
```

The constructor of the class.

Method Detail

getTreeCellRendererComponent

```
public java.awt.Component getTreeCellRendererComponent(javax.swing.JTree tree,
                                                       java.lang.Object value,
                                                       boolean sel,
                                                       boolean expanded,
                                                       boolean leaf,
                                                       int row,
                                                       boolean hasFocus)
```

Returns the current component in the tree which has been rendered.

Package
Class Tree Index Help
PREV CLASS NEXT CLASS FRAMES NO FRAMES All Classes
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Package
Class Tree Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES All Classes
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class MergeSuggestion

```
java.lang.Object
  └ java.awt.Component
      └ java.awt.Container
          └ javax.swing.JComponent
              └ javax.swing.JPanel
                  └ MergeSuggestion
```

All Implemented Interfaces:

javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable

---

public class **MergeSuggestion**
extends javax.swing.JPanel

MergeSuggestion is the dialog that shows the slots, classes etc. suggestions, one by one.

See Also:

Serialized Form

Nested Class Summary

Nested classes inherited from class javax.swing.JPanel
javax.swing.JPanel.AccessibleJPanel

Nested classes inherited from class javax.swing.JComponent
javax.swing.JComponent.AccessibleJComponent

Nested classes inherited from class java.awt.Container
java.awt.Container.AccessibleAWTContainer

Nested classes inherited from class java.awt.Component
java.awt.Component.AccessibleAWTComponent, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy

Field Summary

Fields inherited from class javax.swing.JComponent
accessibleContext, listenerList, TOOL_TIP_TEXT_KEY, ui, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW

Fields inherited from class java.awt.Component
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface java.awt.image.ImageObserver
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

MergeSuggestion() The constructor of MergeSuggestion.

Method Summary

void	**algorithmDialog**(java.lang.String[] algorithm) algorithmDialog creates and shows the dialog box of existing algorithms which the user must choose from before begining the merge process.
void	**classSuggestionDialog**(sambo.util.Pair cpair) classSuggestionDialog creates and shows the dialog box of class suggestions one by one.
javax.swing.JButton	**getNext**() getNext returns the button that merges the current suggestion.
javax.swing.JButton	**getRemaining**() getRemaining returns the button that either merges the remaining CLASS suggestions or skips the remaining RELATION suggestions.
javax.swing.JButton	**getSkipToNext**() getSkipToNext returns the button that skips the current suggestion.
javax.swing.JPanel	**getTextPanel**() getTextPanel returns the panel in which the two boxes that changes the names of the current suggested pair is included.
javax.swing.JButton	**getUndo**() getUndo returns the button that undoes the previously merged suggestion.
void	**languageSuggestionDialog**() languageSuggestionDialog creates and shows the dialog box of supported languages.
void	**relationSuggestionDialog**(sambo.util.Pair rpair) relationSuggestionDialog creates and shows the dialog box of relation suggestions one by one.
void	**reset**() This method resets the dialog in which suggestions are showned.
void	**slotSuggestionDialog**(sambo.util.Pair spair) slotSuggestionDialog creates and shows the dialog box of slot suggestions one by one.

Methods inherited from class javax.swing.JPanel
getAccessibleContext, getUI, getUIClassID, paramString, setUI, updateUI

Methods inherited from class javax.swing.JComponent
addAncestorListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentGraphics, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getGraphics, getHeight, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getWidth, getVisibleRect, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isPreferredSizeSet, isRequestFocusEnabled, isValidateRoot, paint, paintBorder, paintChildren, paintComponent, paintImmediately, paintImmediately, print, printAll, printBorder, printChildren, printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFont, setForeground, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update

Methods inherited from class java.awt.Container
add, add, add, add, add, addContainerListener, addImpl, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent, processEvent, remove, remove, removeAll, removeContainerListener, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setLayout, transferFocusBackward, transferFocusDownCycle, validate, validateTree

Methods inherited from class java.awt.Component
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents, contains, createImage, createImage, createVolatileImage, createVolatileImage, disableEvents, dispatchEvent, enable, enableEvents, enableInputMethods, getBackground, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMouseWheelListeners, getName, getParent, getPeer, getSize, getToolkit, getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner,

isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, postEvent, prepareImage, prepareImage, processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processMouseEvent, processMouseWheelEvent, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, repaint, repaint, repaint, resize, resize, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale, setLocation, setLocation, setName, setSize, setSize, show, show, size, toString, transferFocus, transferFocusUpCycle

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

MergeSuggestion

public **MergeSuggestion**()

The constructor of MergeSuggestion.

Method Detail

algorithmDialog

public void **algorithmDialog**(java.lang.String[] algorithm)

algorithmDialog creates and shows the dialog box of existing algorithms which the user must choose from before begining the merge process.

---

classSuggestionDialog

public void **classSuggestionDialog**(sambo.util.Pair cpair)

classSuggestionDialog creates and shows the dialog box of class suggestions one by one.

---

getNext

public javax.swing.JButton **getNext**()

getNext returns the button that merges the current suggestion.

---

getRemaining

public javax.swing.JButton **getRemaining**()

getRemaining returns the button that either merges the remaining CLASS suggestions or skips the remaining RELATION suggestions.

---

getSkipToNext

public javax.swing.JButton **getSkipToNext**()

getSkipToNext returns the button that skips the current suggestion.

---

getTextPanel

public javax.swing.JPanel **getTextPanel**()

getTextPanel returns the panel in which the two boxes that changes the names of the current suggested pair is included.

getUndo

public javax.swing.JButton **getUndo**()

getUndo returns the button that undoes the previously merged suggestion.

languageSuggestionDialog

public void **languageSuggestionDialog**()

languageSuggestionDialog creates and shows the dialog box of supported languages.

relationSuggestionDialog

public void **relationSuggestionDialog**(sambo.util.Pair rpair)

relationSuggestionDialog creates and shows the dialog box of relation suggestions one by one.

reset

public void **reset**()

This method resets the dialog in which suggestions are showned.

slotSuggestionDialog

public void **slotSuggestionDialog**(sambo.util.Pair spair)

slotSuggestionDialog creates and shows the dialog box of slot suggestions one by one.

Package
Class **Tree** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Package
Class Tree Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES All Classes
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class HelpAboutDialog

```
java.lang.Object
 └ java.awt.Component
    └ java.awt.Container
       └ java.awt.Window
          └ java.awt.Dialog
             └ javax.swing.JDialog
                └ HelpAboutDialog
```

All Implemented Interfaces:

javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, javax.swing.RootPaneContainer, java.io.Serializable, javax.swing.WindowConstants

public class **HelpAboutDialog**
extends javax.swing.JDialog

HelpAboutDialog creates the dialog boxes for the about option and the help option in the menubar. It also shows the requested dialog box.

See Also:

Serialized Form

Nested Class Summary

Nested classes inherited from class javax.swing.JDialog
javax.swing.JDialog.AccessibleJDialog

Nested classes inherited from class java.awt.Dialog
java.awt.Dialog.AccessibleAWTDialog

Nested classes inherited from class java.awt.Window
java.awt.Window.AccessibleAWTWindow

Nested classes inherited from class java.awt.Container
java.awt.Container.AccessibleAWTContainer

Nested classes inherited from class java.awt.Component
java.awt.Component.AccessibleAWTComponent, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy

Field Summary

Fields inherited from class javax.swing.JDialog
accessibleContext, rootPane, rootPaneCheckingEnabled

Fields inherited from class java.awt.Component
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface javax.swing.WindowConstants
DISPOSE_ON_CLOSE, DO_NOTHING_ON_CLOSE, EXIT_ON_CLOSE, HIDE_ON_CLOSE

Fields inherited from interface java.awt.image.ImageObserver
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

HelpAboutDialog(javax.swing.JFrame frame, java.lang.String imagePath) The constructor of the class.

Method Summary

javax.swing.JDialog	**getAboutDialog**() getAboutDialog returns the about dialog box when it is requested.
javax.swing.JDialog	**getHelpDialog**() getHelpDialog returns the help dialog box when it is requested.

Methods inherited from class javax.swing.JDialog
addImpl, createRootPane, dialogInit, getAccessibleContext, getContentPane, getDefaultCloseOperation, getGlassPane, getJMenuBar, getLayeredPane, getRootPane, isDefaultLookAndFeelDecorated, isRootPaneCheckingEnabled, paramString, processWindowEvent, remove, setContentPane, setDefaultCloseOperation, setDefaultLookAndFeelDecorated, setGlassPane, setJMenuBar, setLayeredPane, setLayout, setRootPane, setRootPaneCheckingEnabled, update

Methods inherited from class java.awt.Dialog
addNotify, dispose, getTitle, hide, isModal, isResizable, isUndecorated, setModal, setResizable, setTitle, setUndecorated, show

Methods inherited from class java.awt.Window
addPropertyChangeListener, addPropertyChangeListener, addWindowFocusListener, addWindowListener, addWindowStateListener, applyResourceBundle, applyResourceBundle, createBufferStrategy, createBufferStrategy, finalize, getBufferStrategy, getFocusableWindowState, getFocusCycleRootAncestor, getFocusOwner, getFocusTraversalKeys, getGraphicsConfiguration, getInputContext, getListeners, getLocale, getMostRecentFocusOwner, getOwnedWindows, getOwner, getToolkit, getWarningString, getWindowFocusListeners, getWindowListeners, getWindowStateListeners, isActive, isFocusableWindow, isFocusCycleRoot, isFocused, isShowing, pack, postEvent, processEvent, processWindowFocusEvent, processWindowStateEvent, removeWindowFocusListener, removeWindowListener, removeWindowStateListener, setCursor, setFocusableWindowState, setFocusCycleRoot, setLocationRelativeTo, toBack, toFront

Methods inherited from class java.awt.Container
add, add, add, add, add, addContainerListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getAlignmentX, getAlignmentY, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getContainerListeners, getFocusTraversalPolicy, getInsets, getLayout, getMaximumSize, getMinimumSize, getPreferredSize, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paint, paintComponents, preferredSize, print, printComponents, processContainerEvent, remove, removeAll, removeContainerListener, removeNotify, setFocusTraversalKeys, setFocusTraversalPolicy, setFont, transferFocusBackward, transferFocusDownCycle, validate, validateTree

Methods inherited from class java.awt.Component
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, createVolatileImage, createVolatileImage, disable, disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, getBackground, getBounds, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocation, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMouseWheelListeners, getName, getParent, getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getSize, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isDoubleBuffered, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isOpaque, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, prepareImage, prepareImage, printAll, processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, processMouseWheelEvent, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setDropTarget, setEnabled, setFocusable, setFocusTraversalKeysEnabled, setForeground, setIgnoreRepaint, setLocale, setLocation, setLocation, setName, setSize, setSize, setVisible, show, size, toString, transferFocus, transferFocusUpCycle

Methods inherited from class java.lang.Object
clone, equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

HelpAboutDialog

```
public HelpAboutDialog(javax.swing.JFrame frame,
                       java.lang.String imagePath)
```

The constructor of the class.

Method Detail

getAboutDialog

public javax.swing.JDialog **getAboutDialog**()

getAboutDialog returns the about dialog box when it is requested.

getHelpDialog

public javax.swing.JDialog **getHelpDialog**()

getHelpDialog returns the help dialog box when it is requested.

Package

Class **Tree** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Package

Class **Tree** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class FileExistException

java.lang.Object
└ java.lang.Throwable
  └ java.lang.Exception
    └ **FileExistException**

All Implemented Interfaces:
java.io.Serializable

public class **FileExistException**
extends java.lang.Exception

FileExistException throws an exception when a user tries to open a file that doesn't exist or when a user tries to save a file that already exists.

See Also:
Serialized Form

Constructor Summary

FileExistException(java.lang.String s1) The first constructor of the class.
FileExistException(java.lang.String s1, java.lang.Throwable throwable) The third constructor of the class.
FileExistException(java.lang.Throwable throwable) The second constructor of the class.

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

FileExistException

public **FileExistException**(java.lang.String s1)

The first constructor of the class.

FileExistException

public **FileExistException**(java.lang.String s1,
                          java.lang.Throwable throwable)

The third constructor of the class.

FileExistException

public **FileExistException**(java.lang.Throwable throwable)

The second constructor of the class.

Package

Class **Tree** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Class FileDialog

java.lang.Object
└ java.awt.Component
  └ java.awt.Container
    └ javax.swing.JComponent
      └ javax.swing.JFileChooser
        └ **FileDialog**

All Implemented Interfaces:

javax.accessibility.Accessible, java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable

---

public class **FileDialog**
extends javax.swing.JFileChooser

The class FileDialog creates a dialog for opening a file and a another dialog for saving a file.

See Also:

Serialized Form

Nested Class Summary

Nested classes inherited from class javax.swing.JFileChooser
javax.swing.JFileChooser.AccessibleJFileChooser

Nested classes inherited from class javax.swing.JComponent
javax.swing.JComponent.AccessibleJComponent

Nested classes inherited from class java.awt.Container
java.awt.Container.AccessibleAWTContainer

Nested classes inherited from class java.awt.Component
java.awt.Component.AccessibleAWTComponent, java.awt.Component.BltBufferStrategy, java.awt.Component.FlipBufferStrategy

Field Summary

Fields inherited from class javax.swing.JFileChooser
ACCEPT_ALL_FILE_FILTER_USED_CHANGED_PROPERTY, accessibleContext, ACCESSORY_CHANGED_PROPERTY, APPROVE_BUTTON_MNEMONIC_CHANGED_PROPERTY, APPROVE_BUTTON_TEXT_CHANGED_PROPERTY, APPROVE_BUTTON_TOOL_TIP_TEXT_CHANGED_PROPERTY, APPROVE_OPTION,

APPROVE_SELECTION, CANCEL_OPTION, CANCEL_SELECTION, CHOOSABLE_FILE_FILTER_CHANGED_PROPERTY, CONTROL_BUTTONS_ARE_SHOWN_CHANGED_PROPERTY, CUSTOM_DIALOG, DIALOG_TITLE_CHANGED_PROPERTY, DIALOG_TYPE_CHANGED_PROPERTY, DIRECTORIES_ONLY, DIRECTORY_CHANGED_PROPERTY, ERROR_OPTION, FILE_FILTER_CHANGED_PROPERTY, FILE_HIDING_CHANGED_PROPERTY, FILE_SELECTION_MODE_CHANGED_PROPERTY, FILE_SYSTEM_VIEW_CHANGED_PROPERTY, FILE_VIEW_CHANGED_PROPERTY, FILES_AND_DIRECTORIES, FILES_ONLY, MULTI_SELECTION_ENABLED_CHANGED_PROPERTY, OPEN_DIALOG, SAVE_DIALOG, SELECTED_FILE_CHANGED_PROPERTY, SELECTED_FILES_CHANGED_PROPERTY

Fields inherited from class javax.swing.JComponent
listenerList, TOOL_TIP_TEXT_KEY, ui, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW

Fields inherited from class java.awt.Component
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface java.awt.image.ImageObserver
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

FileDialog() The constructor of the class.

Method Summary

java.lang.String	**getFilePath**(javax.swing.JFrame frame, java.awt.Container c) Checks if the requested file exists, if it exists, opens it and returns the path of the file.
java.lang.String	**getOpenFileName**() Returns the name of the opened file.
java.lang.String	**getSaveFileName**() Returns the name of the saved file.
java.lang.String	**getSaveFilePath**() Returns the path of the saved file.
java.lang.String	**saveDialog**(javax.swing.JFrame frame) Checks if the requested file already exists, if it exists, throws an exception.
void	**setFilter**(java.lang.String language, boolean open) Sets the file filter depending on the parameter language and the parameter open so that the right file filter is choosable when saving or opening a file.
void	**setSave**(boolean save) Sets that the file has been saved before.

Methods inherited from class javax.swing.JFileChooser
accept, addActionListener, addChoosableFileFilter, approveSelection, cancelSelection,

changeToParentDirectory, createDialog, ensureFileIsVisible, fireActionPerformed, getAcceptAllFileFilter, getAccessibleContext, getAccessory, getActionListeners, getApproveButtonMnemonic, getApproveButtonText, getApproveButtonToolTipText, getChoosableFileFilters, getControlButtonsAreShown, getCurrentDirectory, getDescription, getDialogTitle, getDialogType, getDragEnabled, getFileFilter, getFileSelectionMode, getFileSystemView, getFileView, getIcon, getName, getSelectedFile, getSelectedFiles, getTypeDescription, getUI, getUIClassID, isAcceptAllFileFilterUsed, isDirectorySelectionEnabled, isFileHidingEnabled, isFileSelectionEnabled, isMultiSelectionEnabled, isTraversable, paramString, removeActionListener, removeChoosableFileFilter, rescanCurrentDirectory, resetChoosableFileFilters, setAcceptAllFileFilterUsed, setAccessory, setApproveButtonMnemonic, setApproveButtonMnemonic, setApproveButtonText, setApproveButtonToolTipText, setControlButtonsAreShown, setCurrentDirectory, setDialogTitle, setDialogType, setDragEnabled, setFileFilter, setFileHidingEnabled, setFileSelectionMode, setFileSystemView, setFileView, setMultiSelectionEnabled, setSelectedFile, setSelectedFiles, setup, showDialog, showOpenDialog, showSaveDialog, updateUI

Methods inherited from class javax.swing.JComponent
addAncestorListener, addNotify, addPropertyChangeListener, addPropertyChangeListener, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, fireVetoableChange, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentGraphics, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getGraphics, getHeight, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPreferredSize, getPropertyChangeListeners, getPropertyChangeListeners, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getWidth, getVisibleRect, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isMaximumSizeSet, isMinimumSizeSet, isOpaque, isOptimizedDrawingEnabled, isPaintingTile, isPreferredSizeSet, isRequestFocusEnabled, isValidateRoot, paint, paintBorder, paintChildren, paintComponent, paintImmediately, paintImmediately, print, printAll, printBorder, printChildren, printComponent, processComponentKeyEvent, processKeyBinding, processKeyEvent, processMouseMotionEvent, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFont, setForeground, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setUI, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update

Methods inherited from class java.awt.Container
add, add, add, add, add, addContainerListener, addImpl, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paintComponents, preferredSize, printComponents, processContainerEvent, processEvent, remove, remove, removeAll, removeContainerListener, setFocusCycleRoot, setFocusTraversalKeys, setFocusTraversalPolicy, setLayout, transferFocusBackward, transferFocusDownCycle, validate, validateTree

Methods inherited from class java.awt.Component
action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, coalesceEvents, contains, createImage, createImage, createVolatileImage, createVolatileImage, disableEvents, dispatchEvent, enable, enableEvents, enableInputMethods, getBackground, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getFontMetrics, getForeground, getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMouseWheelListeners, getName, getParent, getPeer, getSize, getToolkit, getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, postEvent, prepareImage, prepareImage, processComponentEvent, processFocusEvent, processHierarchyBoundsEvent, processHierarchyEvent, processInputMethodEvent, processMouseEvent, processMouseWheelEvent, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, repaint, repaint, repaint, resize, resize, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale, setLocation, setLocation, setName, setSize, setSize, show, show, size, toString, transferFocus, transferFocusUpCycle

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

FileDialog

public **FileDialog**()

The constructor of the class.

Method Detail

getFilePath

```
public java.lang.String getFilePath(javax.swing.JFrame frame,
                                    java.awt.Container c)
                             throws FileExistException
```

Checks if the requested file exists, if it exists, opens it and returns the path of the file. If it does not exist, throws an exception.

Throws:

FileExistException

getOpenFileName

public java.lang.String **getOpenFileName**()

Returns the name of the opened file.

getSaveFileName

public java.lang.String **getSaveFileName**()

Returns the name of the saved file.

---

getSaveFilePath

public java.lang.String **getSaveFilePath**()

Returns the path of the saved file.

---

saveDialog

public java.lang.String **saveDialog**(javax.swing.JFrame frame)
throws FileExistException

Checks if the requested file already exists, if it exists, throws an exception. If it does not exist, create the file, saves it and returns the path of the saved file.

Throws:
FileExistException

---

setFilter

public void **setFilter**(java.lang.String language,
boolean open)

Sets the file filter depending on the parameter language and the parameter open so that the right file filter is choosable when saving or opening a file.

---

setSave

public void **setSave**(boolean save)

Sets that the file has been saved before.

Package
Class **Tree** **Index** **Help**

PREV CLASS **NEXT CLASS** **FRAMES** **NO FRAMES** **All Classes**

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Appendix C: User manual

The SAMBO system can be used on the following operating systems:

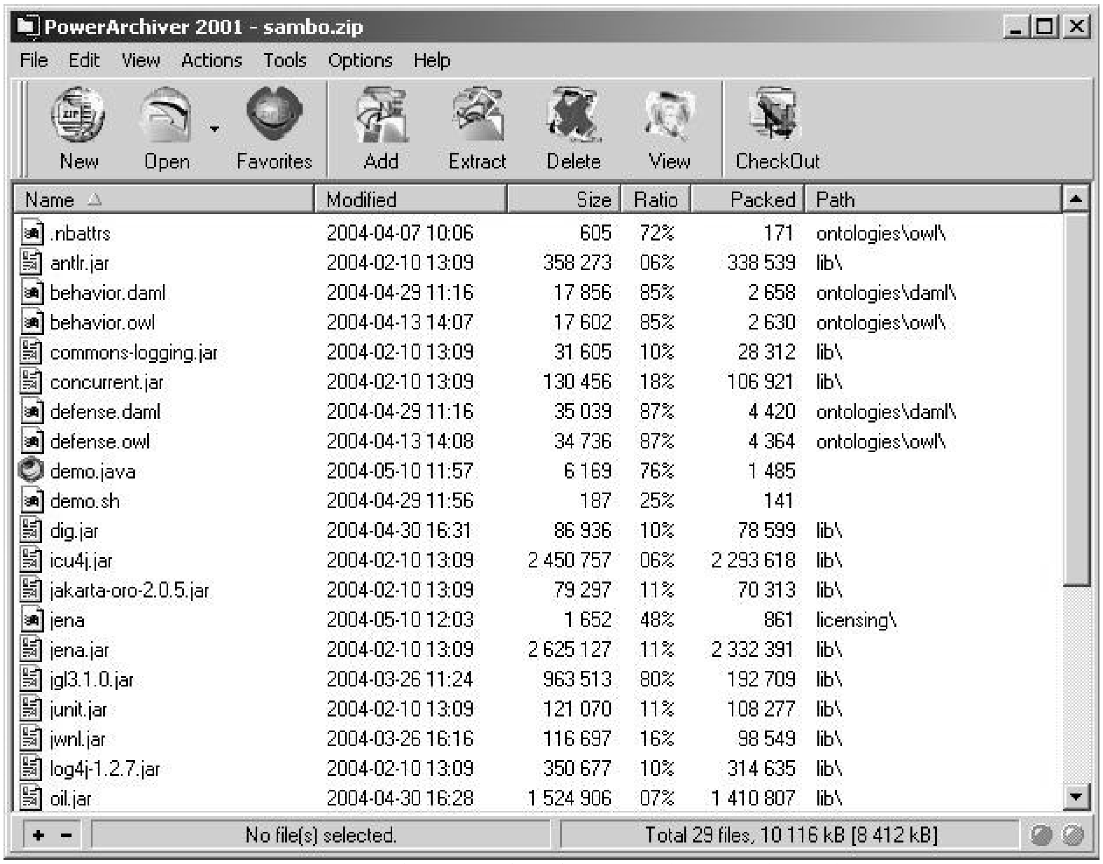
- Windows 95/98/98SE/2000/ME/XP
- UNIX

SAMBO must be installed on the local computer where a user will work with SAMBO and to do that the steps in the next section must be done depending on the computer's operating system.

An introduction of "how to use the system" will be covered in the next sections.

Unzip and install

Windows 95/98/98SE/2000/ME/XP: A program that can pack and unpack files must be installed on the local computer. Such program can be e.g. WinZIP, WinRAR, PowerArchiver, etc. (see figure 9)



Name	Modified	Size	Ratio	Packed	Path
.nbattrs	2004-04-07 10:06	605	72%	171	ontologies\owl\
antlr.jar	2004-02-10 13:09	358 273	06%	338 539	lib\
behavior.daml	2004-04-29 11:16	17 856	85%	2 658	ontologies\daml\
behavior.owl	2004-04-13 14:07	17 602	85%	2 630	ontologies\owl\
commons-logging.jar	2004-02-10 13:09	31 605	10%	28 312	lib\
concurrent.jar	2004-02-10 13:09	130 456	18%	106 921	lib\
defense.daml	2004-04-29 11:16	35 039	87%	4 420	ontologies\daml\
defense.owl	2004-04-13 14:08	34 736	87%	4 364	ontologies\owl\
demo.java	2004-05-10 11:57	6 169	76%	1 485	
demo.sh	2004-04-29 11:56	187	25%	141	
dig.jar	2004-04-30 16:31	86 936	10%	78 599	lib\
icu4j.jar	2004-02-10 13:09	2 450 757	06%	2 293 618	lib\
jakarta-oro-2.0.5.jar	2004-02-10 13:09	79 297	11%	70 313	lib\
jena	2004-05-10 12:03	1 652	48%	861	licensing\
jena.jar	2004-02-10 13:09	2 625 127	11%	2 332 391	lib\
jgl3.1.0.jar	2004-03-26 11:24	963 513	80%	192 709	lib\
junit.jar	2004-02-10 13:09	121 070	11%	108 277	lib\
jwnl.jar	2004-03-26 16:16	116 697	16%	98 549	lib\
log4j-1.2.7.jar	2004-02-10 13:09	350 677	10%	314 635	lib\
oil.jar	2004-04-30 16:28	1 524 906	07%	1 410 807	lib\

No file(s) selected.

Total 29 files, 10 116 kB [8 412 kB]

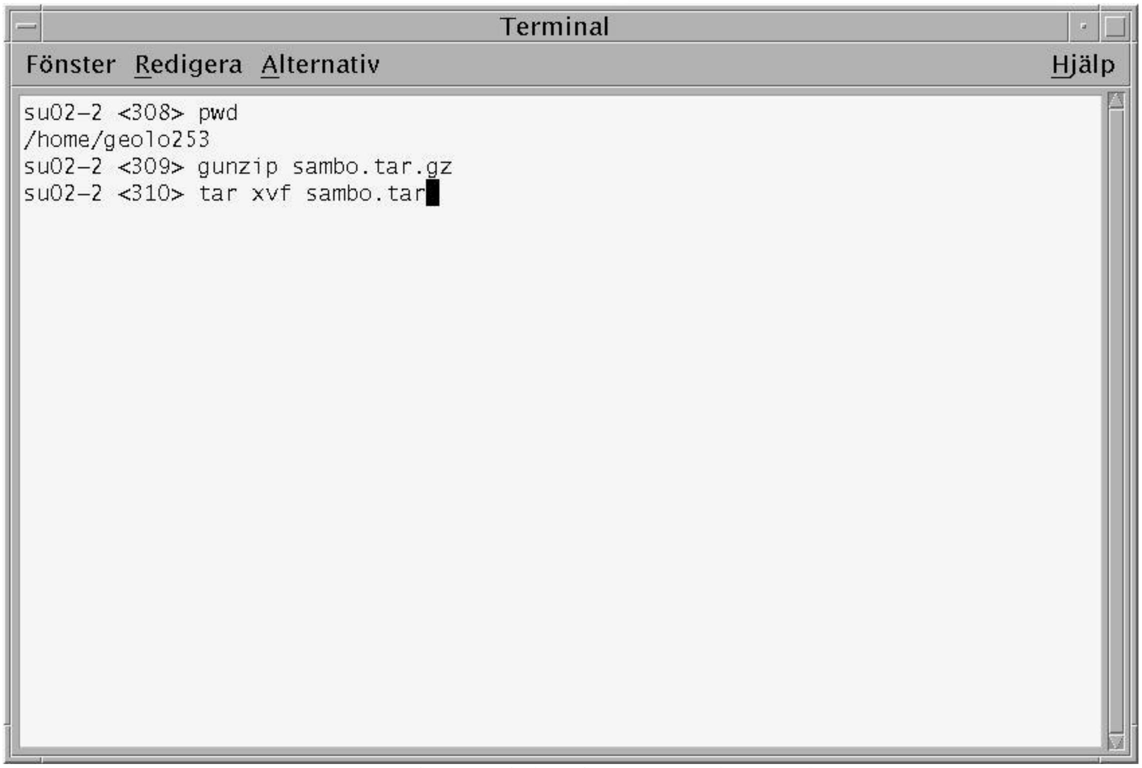
Figure 9. The user interface of a program that can pack/unpack

Download the file sambo.tar.gz and save it in the user's home directory.
Double left-click on the file sambo.tar.gz to unpack it. A window (like the one above) will appear. Click on the button that says, "Extract". A new smaller window will appear. Choose the user's home directory and then click on the "Extract" button. The unpacking process will start and when it is finished, close the program. The installation of SAMBO is done.

UNIX: Download the file sambo.tar.gz and save it in the user's home directory.
Open a terminal shell. Make sure that the current directory is the user's home directory. Type the following two commands at the prompt (see figure 10):

```
gunzip sambo.tar.gz

tar xvf sambo.tar
```



```
su02-2 <308> pwd
/home/geolo253
su02-2 <309> gunzip sambo.tar.gz
su02-2 <310> tar xvf sambo.tar
```

Figure 10. The terminal where the two commands are typed in

The unpacking process will start and when it is done close the terminal. The installation of SAMBO is done.

How to use the system

Windows 95/98/98SE/2000/ME/XP: To run SAMBO, the user must double left-click on the file "sambo.bat" (its location is in the directory where SAMBO is installed). The user can copy the shortcut "SAMBO.lnk" to the desktop to access SAMBO faster.

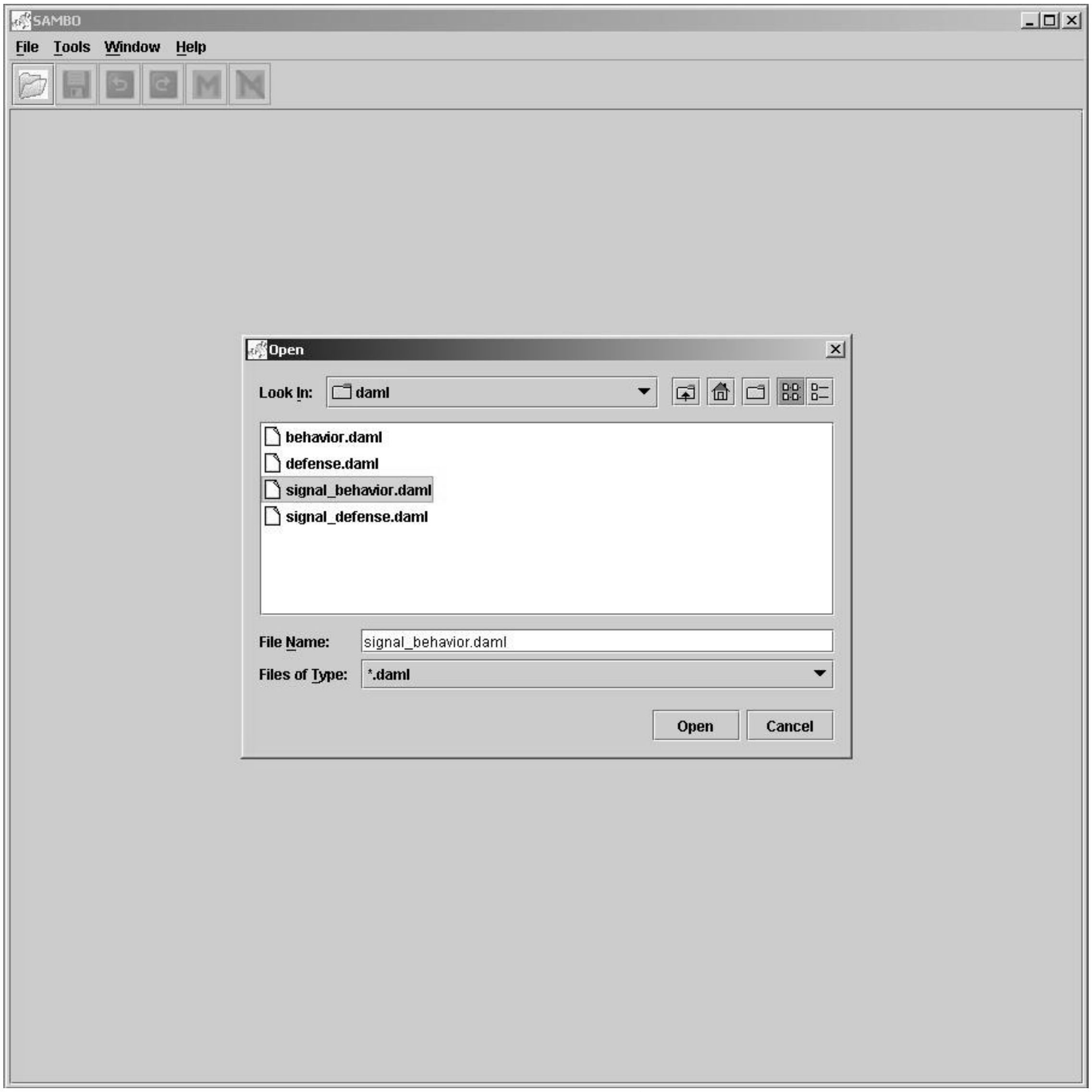
UNIX: To run SAMBO, the user must open a terminal shell, go to the directory of SAMBO and type the following command at the prompt:

sambo.sh &

Open ontology, there are three ways to open an ontology:

1. Click the icon 📂
2. Press the keyboard's keys Ctrl+O
3. From the menu File⇒Open

Choose the extension of the file to be visible and the file to open then click on the "Open" button. Do the same steps again to open another ontology (see figure 11).



Look In: daml

behavior.daml
defense.daml
signal_behavior.daml
signal_defense.daml

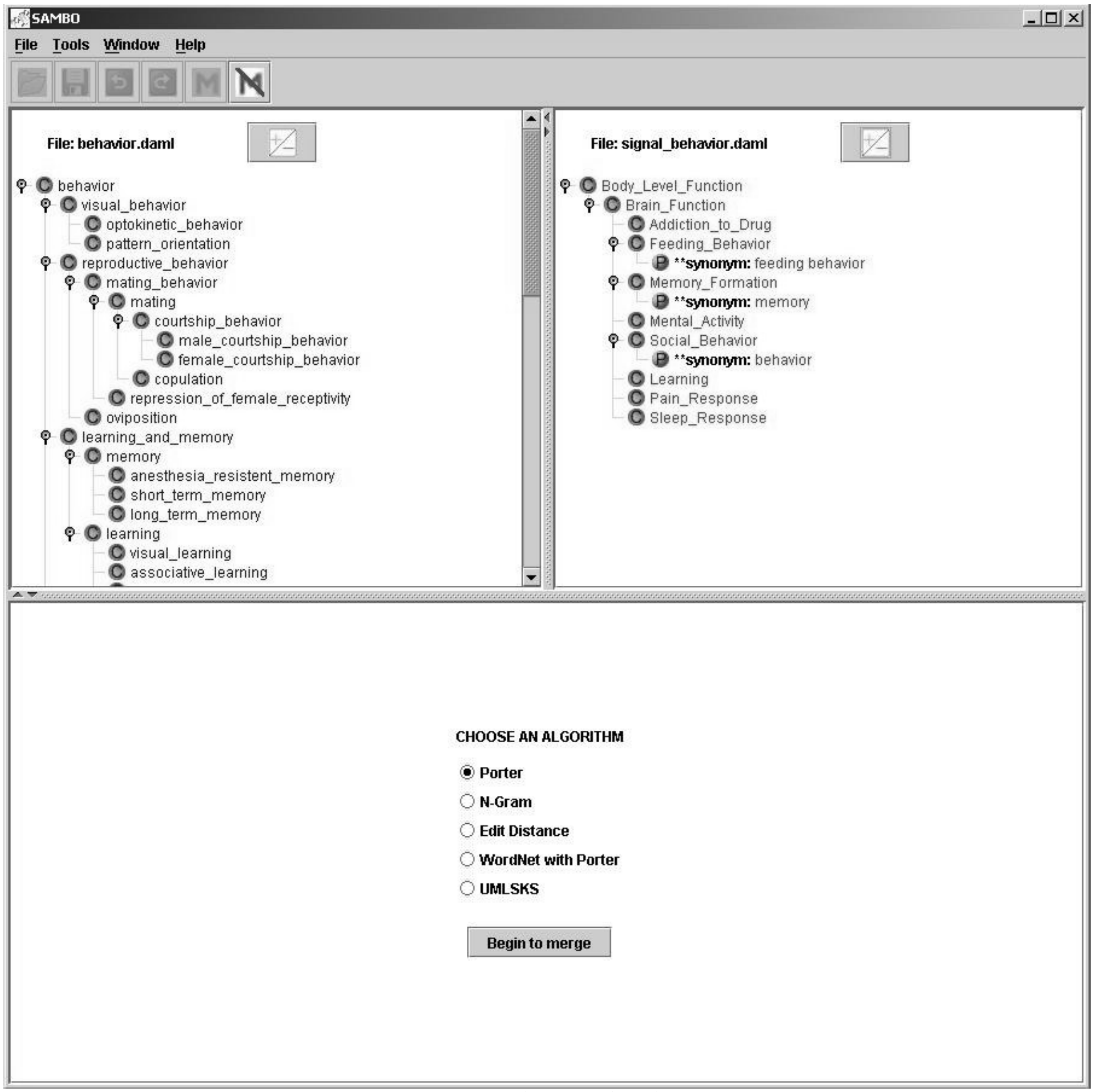
File Name: signal_behavior.daml
Files of Type: *.daml

Figure 11. Open an ontology

Merge ontology, after opening two ontologies choose one of the following ways to begin the merging process.

- Click the icon M
- Press the keyboard's keys Ctrl+M
- From the menu Tools⇒Merge

1. Choose an algorithm from the list as shown in figure 12 and click on the button “Begin to merge”.



CHOOSE AN ALGORITHM

● Porter
○ N-Gram
○ Edit Distance
○ WordNet with Porter
○ UMLSKS

Figure 12. Choose an algorithm

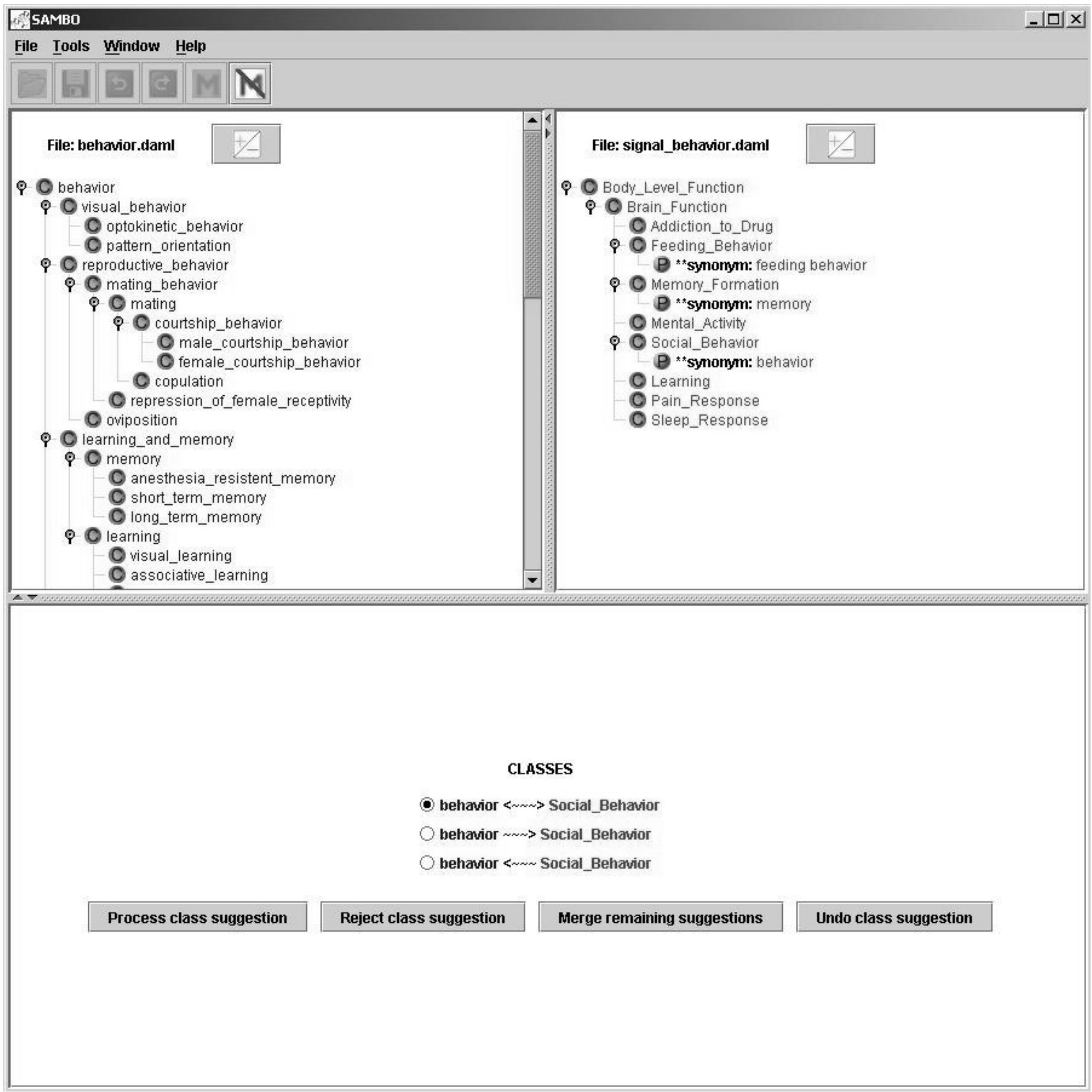
2. Choose the relation type for the current merge suggestion pair to merge,

   - A <~~~> B, Merge the two terms/subjects
   - A ~~~> B, Create a relation where term A is a parent of term B
   - A <~~~ B, Create a relation where term A is a child of term B

The colours indicate which ontology the current term comes from.
If the subjects’ names of the suggestion pair match then two small edit boxes will appear so that the user can change the name of one of the subjects.

3. Click on one of the following buttons (see figure 13):

   - “Process … suggestion” to process the current suggestion and go to the next suggestion
   - “Reject … suggestion” to reject the current suggestion and go to the next suggestion
   - “Undo … suggestion” to undo the previous suggestion and go to the previous suggestion
   - “Process remaining suggestions” to process all remaining suggestions (**only for classes**)
   - “Reject remaining suggestions” to reject all remaining suggestions (**only for relations**)



CLASSES

● behavior <~~~> Social_Behavior
○ behavior ~~~> Social_Behavior
○ behavior <~~~ Social_Behavior

Figure 13. The merging process

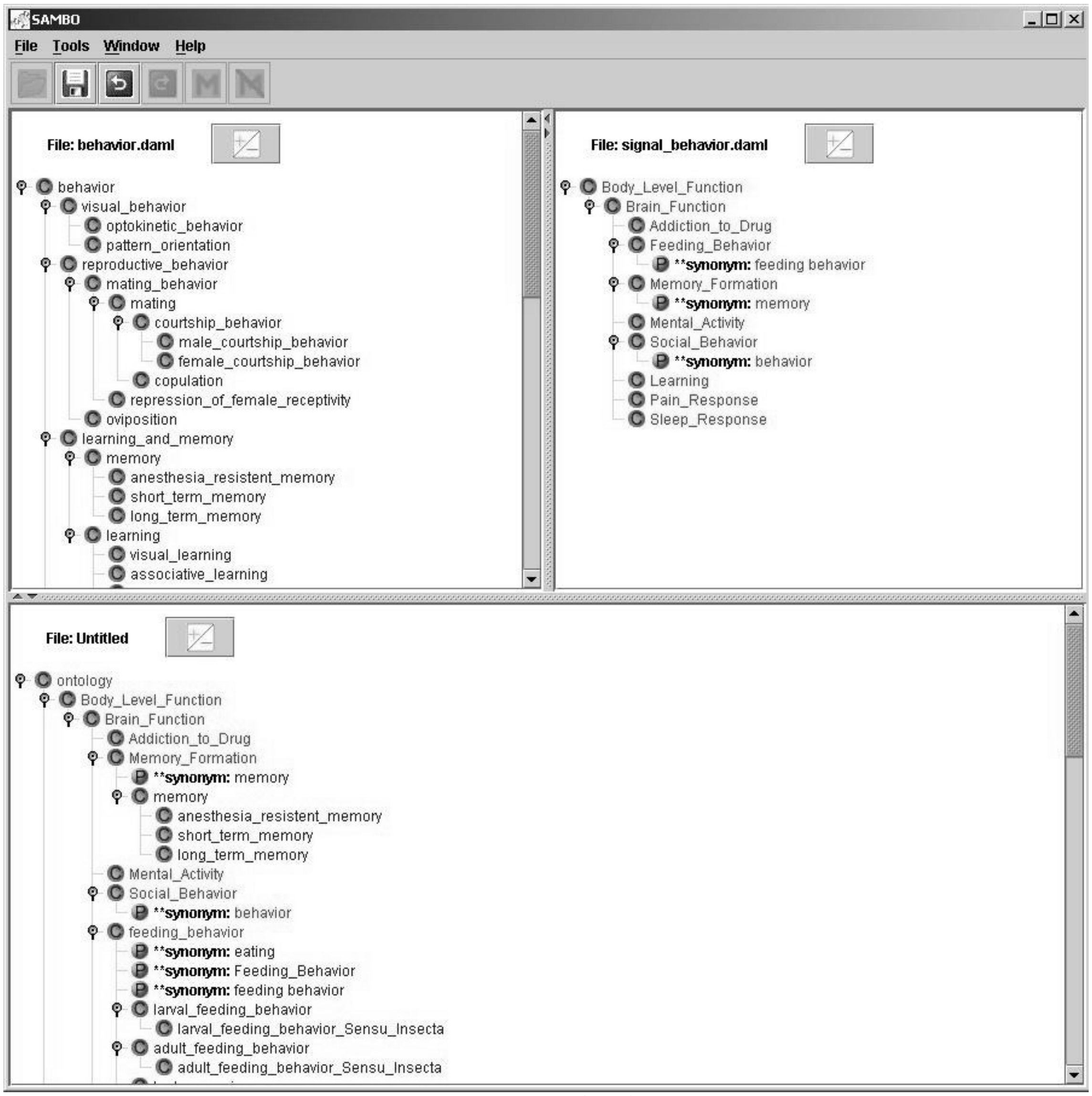
4. Note that during the merging process, the user can always abort the merging process at any step, by clicking on the icon M.

5. Choose a language which the merged ontology will be created in and click the button “Merge” to create the new ontology.

6. The result of a merged ontology is shown at the bottom window (see figure 14). The different colours indicate from which ontology the subject came from.

   - Blue colour – Ontology one (is the left ontology)
   - Red colour – Ontology two (is the right ontology)
   - Green colour – Merged ontology (is the bottom ontology)

   The icons in the tree structure, indicate that a node is of the type class “C” or of the type property “P”.



File: behavior.daml

```
behavior
  visual_behavior
    optokinetic_behavior
    pattern_orientation
  reproductive_behavior
    mating_behavior
      mating
        courtship_behavior
          male_courtship_behavior
          female_courtship_behavior
        copulation
      repression_of_female_receptivity
    oviposition
  learning_and_memory
    memory
      anesthesia_resistent_memory
      short_term_memory
      long_term_memory
    learning
      visual_learning
      associative_learning
```

File: signal_behavior.daml

```
Body_Level_Function
  Brain_Function
    Addiction_to_Drug
    Feeding_Behavior
      **synonym: feeding behavior
    Memory_Formation
      **synonym: memory
    Mental_Activity
    Social_Behavior
      **synonym: behavior
    Learning
    Pain_Response
    Sleep_Response
```

File: Untitled

```
ontology
  Body_Level_Function
    Brain_Function
      Addiction_to_Drug
      Memory_Formation
        **synonym: memory
        memory
          anesthesia_resistent_memory
          short_term_memory
          long_term_memory
      Mental_Activity
      Social_Behavior
        **synonym: behavior
      feeding_behavior
        **synonym: eating
        **synonym: Feeding_Behavior
        **synonym: feeding behavior
        larval_feeding_behavior
          larval_feeding_behavior_Sensu_Insecta
        adult_feeding_behavior
          adult_feeding_behavior_Sensu_Insecta
```

Figure 14. Merged ontology

Save the merged ontology, there are four ways to save a merged ontology:

1. Click the icon 💾
2. Press the keyboard's keys Ctrl+S
3. From the menu File⇒Save
4. From the menu File⇒Save As…

Choose a name for the merged ontology then click on the "Save" button to save the ontology.

Close the ontologies, there are two ways to close an ontology:

1. Press the keyboard's keys Ctrl+Shift+L, R or M (depending on which ontology to close, e.g. L for left ontology, R for right ontology and M for merged ontology)
2. From the menu File⇒Close⇒Close … Ontology

Exit the program SAMBO, there are two ways to exit the program:

1. Press the keyboard's keys Alt+F4
2. From the menu File⇒Exit

Undo the merged ontology, there are three ways to undo a merged ontology:

1. Click the icon ↶
2. Press the keyboard's keys Ctrl+Z
3. From the menu Tools⇒Undo Merged Ontology

Redo the merged ontology, there are three ways to redo a merged ontology:

1. Click the icon ↷
2. Press the keyboard's keys Ctrl+Shift+Z
3. From the menu Tools⇒Redo Merged Ontology

Showing the ontologies, the user can choose which ontology is to be shown by choosing:

- From the menu Window⇒Show Left Window
- From the menu Window⇒Show Right Window
- From the menu Window⇒Show Bottom Window
- From the menu Window⇒Show All Windows

Help menu, the user can get a quick help and a short information about SAMBO by choosing:

- From the menu Help⇒Help…
- From the menu Help⇒About…

LINKÖPINGS UNIVERSITET

Avdelning, Institution
Division, Department

Institutionen för datavetenskap
581 83 LINKÖPING

Datum
Date
2004-12-17

Språk
Language
Svenska/Swedish
X Engelska/English

Rapporttyp
Report category
Licentiatavhandling
X Examensarbete
C-uppsats
D-uppsats
Övrig rapport
____

ISBN

ISRN LITH-IDA-EX-ING--04/020--SE

Serietitel och serienummer
Title of series, numbering

ISSN

URL för elektronisk version
http://www.ep.liu.se/exjobb/ida/2004/dd-c/020/

Titel
Title

A user interface for the ontology merging tool SAMBO

Författare
Author

Bassam Abdulahad, Georgios Lounis

Sammanfattning
Abstract

Ontologies have become an important tool for representing data in a structured manner. Merging ontologies allows for the creation of ontologies that later can be composed into larger ontologies as well as for recognizing patterns and similarities between ontologies. Ontologies are being used nowadays in many areas, including bioinformatics. In this thesis, we present a desktop version of SAMBO, a system for merging ontologies that are represented in the languages OWL and DAML+OIL. The system has been developed in the programming language JAVA with JDK (Java Development Kit) 1.4.2. The user can open a file locally or from the network and can merge ontologies using suggestions generated by the SAMBO algorithm. SAMBO provides a user-friendly graphical interface, which guides the user through the merging process.

Nyckelord
Keyword

Merging, ontology, tool, bioinformatics