

# iSeM

## Intelligent Service Matchmaker

Version 1.0

### User Manual

2010-04-20

Matthias Klusch, Patrick Kapahnke



Deutsches Forschungsinstitut für Künstliche Intelligenz DFKI GmbH  
Stuhlsatzenhausweg 3, 66123 Saarbrücken

Copyright ©: DFKI, 2010, All Rights Reserved

## Table of Contents

<b>1. Introduction .....</b>	<b>- 2 -</b>
<b>2. Command Line Interface Tool.....</b>	<b>- 3 -</b>
2.1. How to use iSeM from the CLI.....	- 3 -
2.1.1. Service Matching.....	- 4 -
2.1.2. Training .....	- 5 -
2.1.3. Miscellaneous.....	- 5 -
<b>3. iSeM API .....</b>	<b>- 6 -</b>
3.1. S2M2-DE .....	- 6 -
3.2. How to use iSeM in own applications.....	- 7 -
<b>4. iSeM as plug-in for SME<sup>2</sup> .....</b>	<b>- 8 -</b>
<b>5. Contact and Copyright .....</b>	<b>- 9 -</b>
<b>References .....</b>	<b>- 10 -</b>

## 1. Introduction

This document provides information on how to use the hybrid adaptive matchmaker iSeM 1.0. In the current version, it is available via a Java API, as command line interface tool and as plug-in for the Semantic Matchmaker Evaluation Environment SME<sup>2</sup>. A convenient graphical user interface is subject to future work.

iSeM 1.0 performs adaptive and hybrid semantic service selection exploiting the full semantic OWL-S<sup>1</sup> profile of service offers. In particular, it complements its strict logical signature matching with approximated reasoning based on logical concept abduction and contraction (see [2], [3]) together with information-theoretic similarity and evidential coherence-based valuation, as well as non-logic-based approximated matching based on text similarity measures, originally used for pure textual comparison of documents in the context of information retrieval, and structural comparison of logical definitions. Besides that, it performs logical specification plug-in matching of service preconditions and effects described in PDDL<sup>2</sup> using  $\theta$ -subsumption (see [4]). Thus, iSeM resembles a full IOPE (inputs, outputs, preconditions and effects) matchmaker.

The optimal aggregation strategy of the logical and non-logical aspects of a matching process described above is learned off-line by means of a binary SVM-based service relevance classifier which also performs result ranking. Evidential coherence-based feature space pruning is applied to reduce training and query response time and to increase performance with respect to ranking precision.

For more detailed information on iSeM 1.0, we refer the interested reader to [1].

iSeM is publicly available at [semwebcentral.org](http://semwebcentral.org):

<http://projects.semwebcentral.org/projects/isem>

As reasoner for approximated logic-based signature matching, iSeM currently uses MaMaS-tng developed by the SisInf Research Group (Politecnico di Bari, Italy, see [2]), which is available at <http://dee227.poliba.it:8080/MAMAS-tng/DIG> via DIG (DL Implementation Group) interface. The project page of MaMaS-tng is:

<http://sisinflab.poliba.it/MAMAS-tng/>

For future releases, it is planned to provide support for approximated logic-based matching in the complete scope of OWL-DL.

iSeM 1.0 requires a Java 1.6 (or higher) runtime environment.

---

<sup>1</sup> Please note: The ideas behind iSeM are not restricted to OWL-S. Support for other formalisms may be added with moderate effort using the API.

<sup>2</sup> Since PDDL allows first-order formulas to describe P/E, many other formalisms that allow a semantics-preserving translation are covered in principle. Please note: currently, only Horn-like clauses are handled in the implementation.

## 2. Command Line Interface Tool

iSeM 1.0 provides a Java-based command line interface (CLI) tool. Currently, only the fully-fledged variant of iSeM including all basic matching filters as described in [1] is available this way. To access reduced variants or to setup arbitrary feature spaces, the API must be used directly. It is planned to enable iSeM configuration using XML and a graphical user interface for future releases. The iSeM CLI tool can be accessed using the following base command:

```
java -jar isem.jar
```

Without further parameters, the application will print its usage message and quit.

```
Welcome to iSeM v1.0 (Intelligent Service Matchmaker).
(c) DFKI GmbH 2010

Usage: java -jar isem.jar [-x <testcollection>] [-c <request(s)>
<offer(s)>+] [-o <output>] [-m <model>] [-t <trainingset>] [-v] [-h]
```

In Windows environments, the batch files `isem.bat` or `isem_moremem.bat` can be used as abbreviation. The latter starts iSeM with the `-Xmx1024m` option to provide more main memory.

The following sections describe the parameters and how to setup iSeM for service matching and training.

### 2.1. How to use iSeM from the CLI

Starting iSeM from the CLI using parameter `-h` prints the available set of options including a short description:

```
java -jar isem.jar -h
```

```
-x: Experiment setup using XML test collection from SME2. -h, -x and -t
are exclusive.
<testcollection>: XML format for test collection as in SME2

-c: Experiment setup as subsequent arguments in command line interface. -
h, -x and -t are exclusive.
<request(s)>: URL or path to OWL-S request document or path to directory
of (multiple) OWL-S request documents.
<offer(s)>+: URL or path to OWL-S offer document or path to directory of
(multiple) OWL-S offer documents. Multiple occurrences in arbitrary
combination allowed.

-o: Write result rankings as XML to specified location. If this option is
not set, output will be printed to the console.
<output>: Output XML file.

-m: Set the SVM model file to use or the create. "default_model.xml" is
used by default.
<model>: Model XML file location.
```

-t: Sets iSeM to training mode. Provided test collection XML is used for supervised training. -h, -x and -t are exclusive.  
<trainingset>: Training set to train iSeM. XML format for binary relevance as in SME2.

-v: Run in verbose mode.

-h: Displays this help message.

The following sections describe the available options in more detail.

### 2.1.1. Service Matching

- **-x:** Sets up iSeM for matching given a set of service offers and requests in XML format as used by the Semantic Matchmaker Evaluation Environment SME<sup>2</sup>. This option is especially useful for experiments based on OWLS-TC<sup>3</sup>, which provides a XML file in the required format. A sample call to iSeM using this feature might look as follows:

```
java -jar isem.jar -x owls-tc3.xml
```

- **-c:** This variant enables custom experiment setup in terms of service offers and requests directly via the CLI. It requires exactly one parameter specifying the request document(s) and an arbitrary number of parameters to point to service offers. A request or offer parameter may be one of the following (every combination is allowed):
  - a file system path pointing to a single OWL-S document,
  - a file system path pointing to a directory containing (only) OWL-S documents; this may also be used for the parameter specifying requests to setup experiments with multiple queries,
  - a URL pointing to a single OWL-S document.

A sample call for a request located at the local hard drive and a mix of locally stored and online service offer descriptions may look as follows:

```
java -jar isem.jar -c requests/request.owl  
offers/offer1.owl http://www.example.org/example.owl  
offers/offer2.owl
```

- **-o:** This option enables output of result ranking(s) as XML file for post-processing using arbitrary XML tools. It may be used in combination with service matching options -x or -c described above. If it is not used, iSeM will output its result ranking(s) directly to the console (as pure text without XML formatting).
- **-m:** iSeM uses a Support Vector Machine (SVM) for the adaption of aggregation strategies for its various matching aspects. This option enables the re-use of previously trained models (SVM-specific parameters such as support vectors, error parameter C, RBF Kernel parameter  $\gamma$ , etc.) for matching, i.e. the application of a previously computed aggregation strategy. If this option is not set, the default model

---

<sup>3</sup> Available at <http://projects.semwebcentral.org/projects/owls-tc>

default\_model.xml will be used, which is provided with the iSeM 1.0 distribution. It was generated using 10% of OWLS-TC as training set. An example for using a model other than the default model might be as follows:

```
java -jar isem.jar -x owls-tc3.xml -m my_model.xml
```

### 2.1.2. Training

- **-t:** Sets up iSeM for training mode. For training, the set of samples available in the provided XML file parameter will be used. This XML file must be formatted as SME<sup>2</sup> test collection document. The classification of samples as positive or negative is based on the binary relevance assessments provided in the file. Every sample explicitly stated to be relevant is considered as positive, the remaining samples are negative. Please note: the computed model will be saved at the location specified using the **-m** option. If **-m** is not used, the iSeM CLI tool will use the default location to save its result, thus it will overwrite the original default model. Please make sure to set up the parameters correctly before executing the command to avoid overwriting models that should not be overwritten! Please have a look at the next point for a sample call.
- **-m:** For the training case, this option specifies the output location of the computed aggregation strategy. The following example shows how to train iSeM given a training set and output location:

```
java -jar isem.jar -t training_set.xml -m my_model.xml
```

### 2.1.3. Miscellaneous

- **-v:** Runs iSeM in verbose mode. This may be used in every combination with other options to display the current processing state.
- **-h:** Displays the help message text explaining the parameters.

Please note: The main options for matching and training **-x**, **-c** and **-t** are exclusive. Exactly one of these operations may be performed by iSeM during a single call.

### 3. iSeM API

The following sections provide some basic information about the iSeM API and the S2M2-DE used to implement it. It is not considered to be exhaustive yet and the APIs may be subject to change for future releases. As a more concrete source of information regarding the API, the distribution also provides Javadocs in the `doc/ javadoc` folder.

#### 3.1. S2M2-DE

The Semantic Service MatchMaker Development Environment S2M2-DE v0.1 is used as basis for the iSeM 1.0 implementation. The current status of S2M2-DE is to be considered as preliminary. Various additions and refinements are planned for future versions and it will also be provided as distinct project at [semwebcentral.org](http://semwebcentral.org). Currently, it is delivered with the iSeM distribution only.

The intended architecture of S2M2-DE is sketched in Figure 1. The XML configuration ability is currently not implemented. iSeM implements OWL-S parsing capabilities for the *Service Information Extractor* component, uses a wide range of pre-defined matching expressions as well as new expressions introduced specifically for iSeM for the *Matching Expression Evaluator* to resemble the various matching filters introduced in [1] and applies a simple top-down ranking based on the SVM aggregation (distance from optimally separating hyperplane) in the *Ranking Processor*.

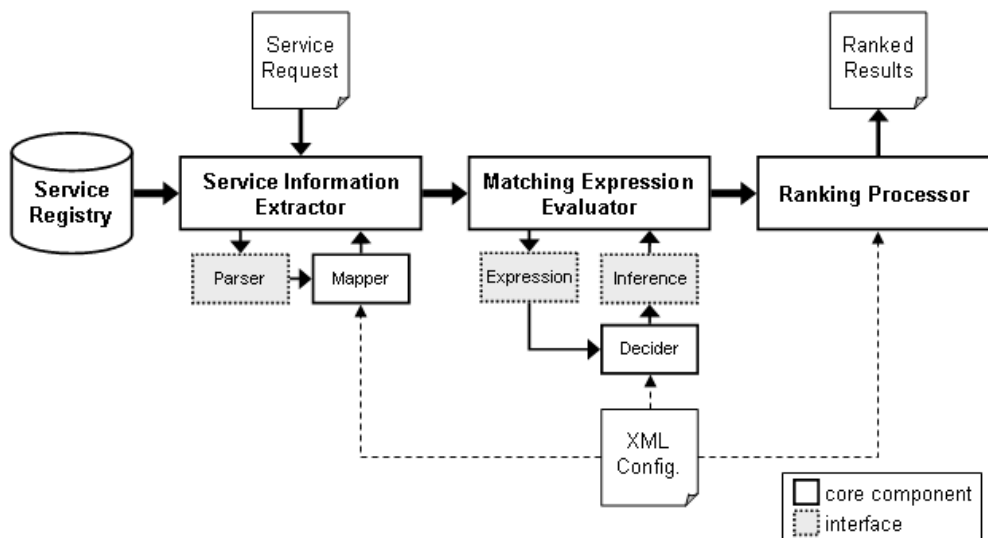


Figure 1 S2M2-DE architecture

In the following, a short overview of the different Java packages provided by S2M2-DE is given.

- **de.dfki.s2m2de:** Main package providing a flexible and configurable matching engine and matching result representation.
- **de.dfki.s2m2de.service:** Implements a basic service registry and contains basic interfaces for services and functional/non-functional parameters. It also contains

specific sub-packages for OWL-S and preconditions and effects in PDDL utilizing OWLS-API<sup>4</sup> and ANTLR<sup>5</sup> 3.2 for parsing.

- **de.dfki.s2m2de.expression:** Contains interfaces and basic implementations for matching filter expressions. It provides a wide range of expressions to construct conjunctions/disjunctions, partial matching result aggregation on service level, numerical operators etc. This package heavily relies on Java generics to allow re-use of filter expressions in different contexts. It also contains various sub-packages for more specialized features such as bipartite graph matching or the SVM as used for example in iSeM.
- **de.dfki.s2m2de.inference:** Defines a set of interfaces for inference engines and tasks to adopt by matchmaker implementations. More specific interfaces suited for DL reasoning and first-order logic theorem-proving are provided in sub-packages. There are also two sample implementations based on Pellet<sup>6</sup> and OWL-API<sup>7</sup> for OWL-DL reasoning and  $\theta$ -subsumption implemented in Prolog using tuProlog<sup>8</sup> for (approximation of) theorem-proving. The inference tasks provided by such implementations are also considered as matching expressions and can be used directly in more complex filter definitions.
- **de.dfki.s2m2de.ranking:** This package contains an interface and basic implementations of ranking strategies.
- **Miscellaneous:** Additionally, S2M2-DE v0.1 contains some other packages, which this manual will not describe in more detail. For example, there exists a package for text similarity measures and a sample implementation of the Cosine measure on the Vector Space Model using Apache Lucene<sup>9</sup> as term index. Sample implementations and a factory class for OWLS-MX-like (one of the predecessors of iSeM, see [5] for details) configurations of S2M2-DE are also provided.

### 3.2. How to use iSeM in own applications

The iSeM 1.0 implementation can be found in the `de.dfki.isem` package. It implements (among others) a specific DL reasoner variant utilizing Pellet and MaMaS-tng as well as a specific variant of SVM aggregation based on evidential coherence-based feature space pruning.

The `IseMFactory` class provides a convenient entry-point to different iSeM setups that can be used to automatically initialize ready-to-use iSeM matchmakers or specific parts of a setup. If other variants are required, we refer the interested developer to the source code of the `create...Filter()`, `create...Setup()` and `create...Matcher()` methods as examples on how to use iSeM and S2M2-DE to procure own matching filter definitions and feature spaces.

---

<sup>4</sup> <http://on.cs.unibas.ch/owl-s-api/>

<sup>5</sup> <http://www.antlr.org/>

<sup>6</sup> <http://clarkparsia.com/pellet/>

<sup>7</sup> <http://owlapi.sourceforge.net/>

<sup>8</sup> <http://sourceforge.net/projects/tuprolog/>

<sup>9</sup> <http://lucene.apache.org/>



#### 4. iSeM as plug-in for SME<sup>2</sup>

The iSeM 1.0 distribution comes with a set of plug-ins for the Semantic Matchmaker Evaluation Environment SME<sup>2</sup>, which is used in the annual Semantic Service Selection contest S3<sup>10</sup>. It is publicly available at [semwebcentral.org](http://semwebcentral.org):

`http://projects.semwebcentral.org/projects/sme2`

To use iSeM as plug-in(s) for SME<sup>2</sup>, please follow the installation instructions provided with the evaluation environment to setup a test collection. To add the iSeM plug-ins to SME<sup>2</sup>, please copy the content of the `sme2` folder (including the `isem` directory) to the corresponding `sme2` (main) folder of SME<sup>2</sup>. To use SME<sup>2</sup> and iSeM, please follow the instructions of the SME<sup>2</sup> manual.

Please note: The SME<sup>2</sup> plug-ins are implemented to automatically train the aggregation strategy based on 2920 (= 10%) sample service offer/request pairs of OWLS-TC3 (which should be provided to SME<sup>2</sup> at `testcollections/owls-tc3`, following the test collection installation instructions). The resulting model file is automatically saved and may be re-used for the CLI tool. The training process may take some time depending on the used settings.

The source code of the plug-ins can be found in the `de.dfki.isem.sme2` package.

---

<sup>10</sup> <http://www-ags.dfki.uni-sb.de/~klusch/s3/>

## **5. Contact and Copyright**

The Intelligent Service Matchmaker iSeM was developed at the German Research Center for Artificial Intelligence DFKI GmbH (<http://www.dfki.de>) in Saarbrücken, Germany. This work was sponsored by the project ISReal (BMBF, 01/2009 - 12/2011)).

Copyright ©: DFKI, 2010, All Rights Reserved.

For bug reports, other technical problems and feature requests please contact Patrick Kapahnke: [patrick.kapahnke@dfki.de](mailto:patrick.kapahnke@dfki.de).

For general scientific inquiries please contact PD Dr. Matthias Klusch: [klusch@dfki.de](mailto:klusch@dfki.de).

## References

- [1] Klusch M., Kapahnke P. (2010): *iSeM: Approximated Reasoning for Adaptive Hybrid Selection of Semantic Services*. Proceedings of the 7<sup>th</sup> Extended Semantic Web Conference (ESWC), Heraklion, Greece.
- [2] Di Noia T., Di Sciascio E., Donini F.M., Mongiello M. (2004): *A system for principled Matchmaking in an electronic marketplace*. International Journal of Electronic Commerce, 8(4):9-37.
- [3] Di Noia T., Di Sciascio E., Donini F.M. (2009): *A Tableaux-based Calculus for Abduction in Expressive Description Logics: Preliminary Results*. Proceedings of the 22<sup>nd</sup> International Workshop on Description Logics (DL).
- [4] Scheffer T., Herbrich R., Wysotzki F. (1996): *Efficient theta-Subsumption based on Graph Algorithms*. Lecture Notes In Computer Science, 1314, Springer.
- [5] Klusch M., Fries B., Sycara K. (2009): *OWLS-MX: A Hybrid Semantic Web Service Matchmaker for OWL-S Services*. Web Semantics, 7(2), Elsevier.