

# OI-Modeler

## User's Guide

*November 27<sup>th</sup>, 2002*



FZI

Forschungszentrum Informatik

at the University of Karlsruhe, Germany

---

# Content

<b>1. INTRODUCTION .....</b>	<b>4</b>
<b>2. THE OI-MODELER.....</b>	<b>4</b>
<b>2.1 The Menus .....</b>	<b>5</b>
2.1.1 The File menu: .....	5
2.1.2 The Edit Menu: .....	6
2.1.3 The View Menu: .....	7
2.1.4 The Procedures Menu:.....	7
<b>2.2 Language parameters .....</b>	<b>10</b>
<b>2.3 Search.....</b>	<b>10</b>
<b>2.4 The Clipboard.....</b>	<b>11</b>
<b>3. WORKING WITH OI-MODELS.....</b>	<b>12</b>
<b>3.1 Creating a new OI-Model.....</b>	<b>12</b>
<b>3.2 Including OI-Models .....</b>	<b>13</b>
<b>3.3 Creating concepts .....</b>	<b>14</b>
<b>3.4 Creating properties .....</b>	<b>14</b>
3.4.1 Properties from a concept: .....	14
3.4.2 Properties to a concept .....	15
<b>3.5 Creating instances .....</b>	<b>16</b>
<b>3.6 Sub-Concepts and Sub-Properties.....</b>	<b>17</b>
<b>3.7 Creating properties between instances. ....</b>	<b>17</b>
<b>3.8 The Context menus of concepts, instances and properties.....</b>	<b>18</b>
3.8.1 The Context menu of concepts: .....	18
3.8.2 The Context menu of Instances .....	18
3.8.3 The Context menu of Properties .....	19
3.8.4 The context menu between two concepts.....	19
3.8.5 The context menu between two instances.....	19
3.8.6 The context menu between two properties .....	20
3.8.7 The context menu between Concepts and Properties.....	20

---

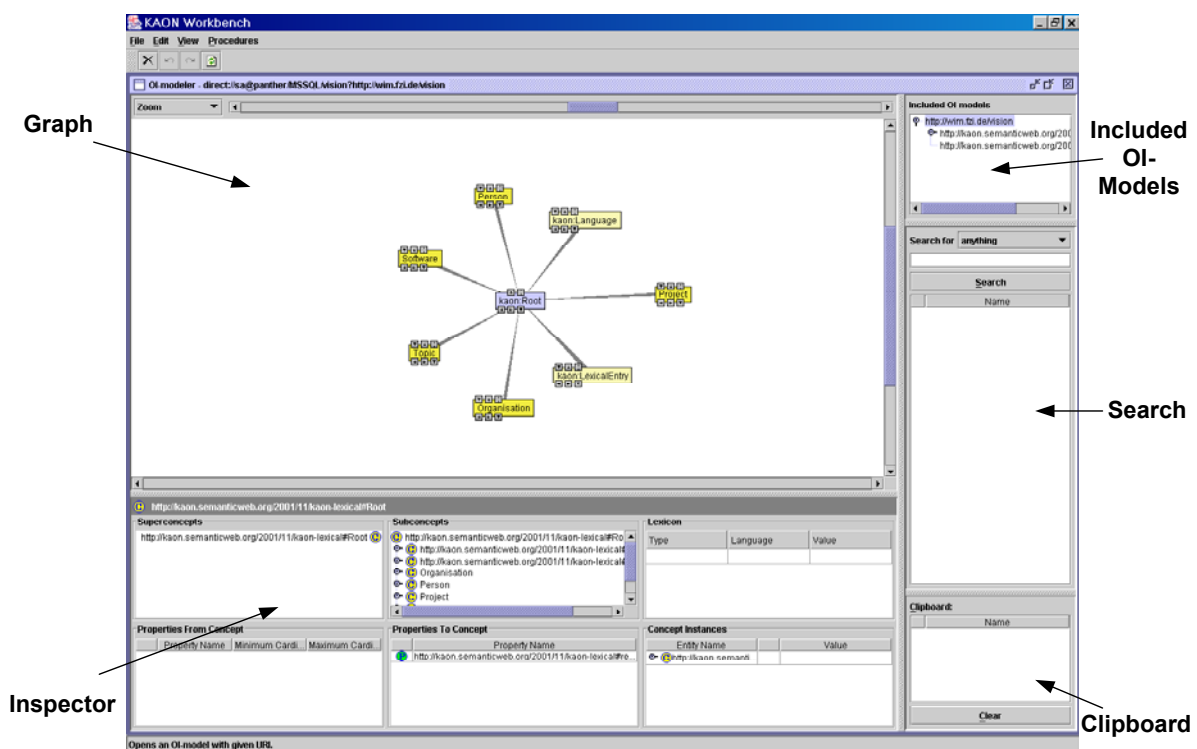
<b>3.9</b>	<b>The Inspector of concepts, properties and instances .....</b>	<b>21</b>
<b>3.10</b>	<b>The lexical layer .....</b>	<b>23</b>
	Modelling the Lexical Layer of Concepts, Attributes, Relations, and Instances.....	23
<b>4.</b>	<b>REFERENCES .....</b>	<b>25</b>

# 1. Introduction

OI-Modeler is a tool for ontology creation and maintenance. The goal of the tool is to allow scalability for editing large ontologies, as well as to incorporate some usability issues related to ontology management. The graph layout algorithms in OI-Modeler are based on an open-source [TouchGraph<sup>1</sup>](http://www.touchgraph.com) library.

## 2. The OI-Modeler

The following picture shows you the user interface of the OI-Modeler with its different windows:

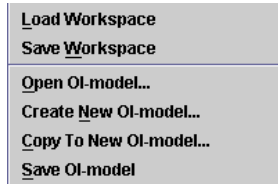


<sup>1</sup> <http://www.touchgraph.com>

---

## 2.1 The Menus

### 2.1.1 The File menu:

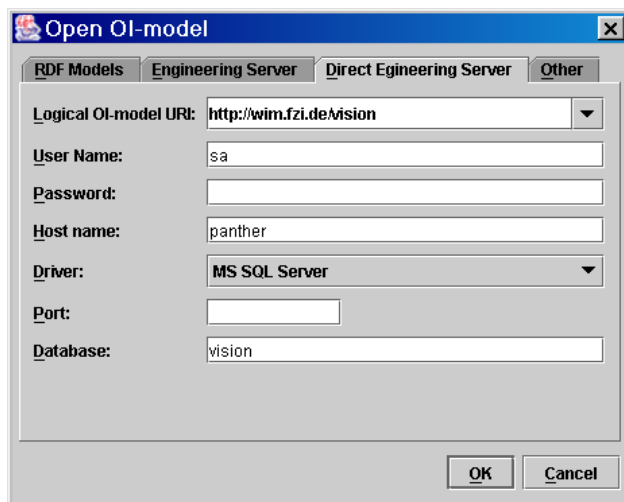


#### Functions:

- Load Workspace – Loads a workspace containing the state of all windows of your last saved session.
- Save Workspace - save a Kaon Workspace.
- .Open OI-Model

#### Open OI-Model:

To open an existing ontology. If you choose this function, another window will appear:



Now you can define where the ontology has to be stored: If you would like to save the ontology on your (local) drive(s), choose “RDF Models”, and then specify the physical URI<sup>2</sup>.

---

<sup>2</sup> Each OI-model has two URIs that uniquely identify it. The *physical URI* is the URI used to access the model. For example, if the model is located in `c:\temp\model.kaon` file, then the physical URI of the model will be `file:/c:/temp/model.kaon` (note that this is really the URI and not a simple file name). The structure of the model's physical URI is dependent on the KAON API implementation used, and it encodes all the information needed to obtain the model from some information source.

Each OI-model has a *logical URI*, which is independent from the physical one.

A model from the previous paragraph may have the logical URI <http://kaon.semanticweb.org/myModel.kaon>, although it is not loaded from the web. A Model's logical URI should be globally unique, whereas its physical URI typically isn't globally unique, and is often relative to the system which processes the model.

---

If the ontology should be stored on a local server, choose “Engineering Server” and then specify the servers name, the port and the logical URI.

If the ontology should be saved on a server, but you want to work directly over the Database, choose “Direct Engineering Server” and specify the logical URI and the database specifications (e.g. DB-Driver, the Database).

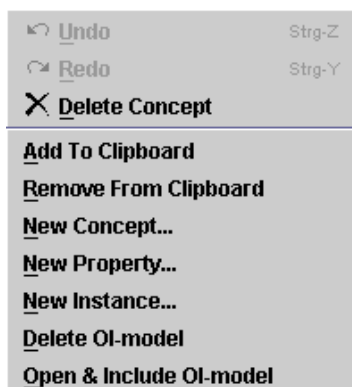
### Create a new OI-Model:

Function to create an empty OI-Model.

### Copy to new OI-Model:

With this option you can copy an existing OI-Model into a new one. This replica has the reference to the original OI-Model. If you choose this function, the same window as in the function “Open OI-Model” appears, and you can choose where to save the backup.

## 2.1.2 The Edit Menu:



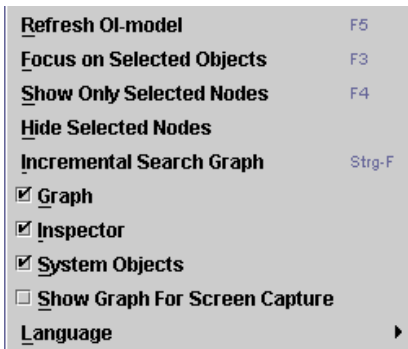
With this menu you can edit the ontology, for example add or delete **Concepts, Instances or Properties**, add nodes to the clipboard for copy/paste use, or open and include existing OI-Models into the current workspace (Details see point 3.2).

With the **Undo – Redo** function you are allowed to undo and redo your last work steps.

---

---

### 2.1.3 The View Menu:



In this menu you can find a refresh function (Shortcut F5) as well as functions for the nodes:

**Show Only Selected Nodes:** This function allows you to only display the nodes you selected (Shortcut F4; By pressing and holding the ctrl button you can select more than one node).

**Hide selected Nodes:** To hide the selected nodes.

**Incremental search graph:** With the function “Incremental Search Graph” you can search the graph for e.g. a keyword incrementally.

**Graph:** Function to display or hide the graph.

**Inspector:** Function to display or hide the inspector.

**System Objects:** An OI-Model consists of three Objects: Kaon-lexical # Root, Kaon-lexical # language and kaon-lexical # LexicalEntry. By switching of the “System Objects” you only visualize the Kaon-Root and so the ontology gets more clear because less concepts and instances are shown in the graph and the inspector.

**Language:** See 2.2 Language parameters.

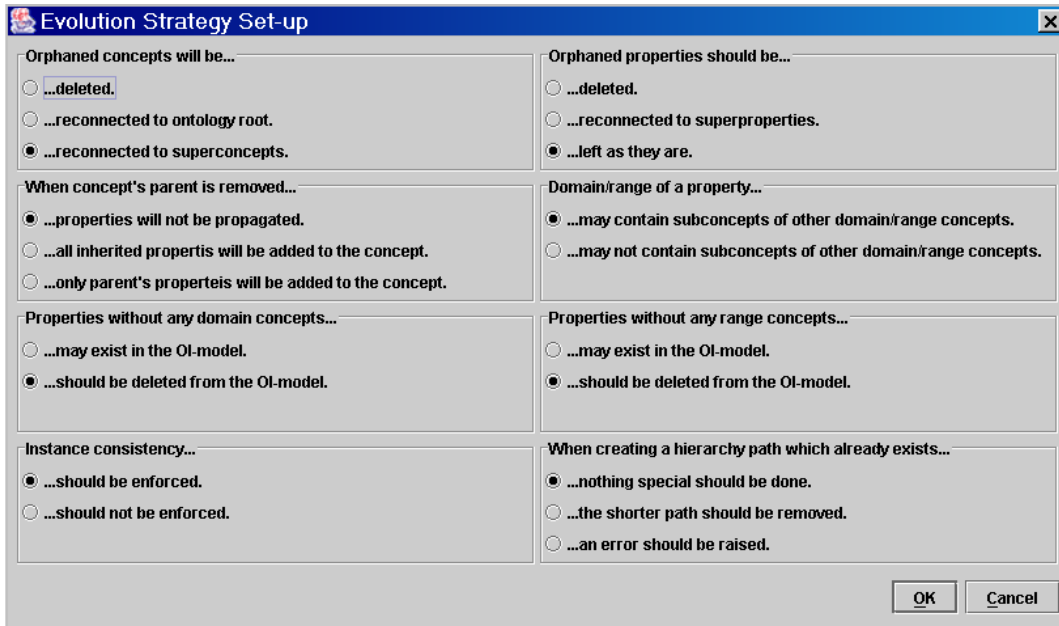
### 2.1.4 The Procedures Menu:



**Functions:**

With the function “**Set Look & Feel**” you can change the design of the user interface.

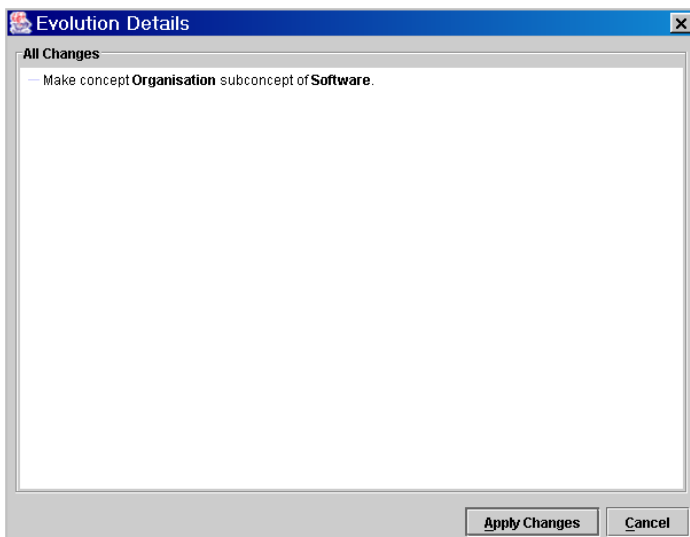
## Set-up Evolution Strategy:



Potentially, an ontology change might corrupt the instances, dependent ontologies as well as application programs running against the ontology and/or the data base. This option allows you to define the strategy how the OI-Modeler handles changes in the ontology, e.g. the deletion of concepts.

## Show Evolution Details:

This function allows you to switch the following window on or off:



This window appears before you apply a change in your OI-Model. It allows you to control the changes you applied.

---

## Create Evolution Log:

The evolution log saves all the evolution changes made during a work session.  
The following figures show an example on how the evolution log works:

The following code of a log file shows the Insertion of a new instance called "Knowledge Technologies":

```
<rdf:RDF xml:base="file:/Z:/VisionOntologies/log.kaon"
  xmlns:rdfs="&rdfs;"
  xmlns:kaon="&kaon;"
  xmlns:a="&a;"
  xmlns:rdf="&rdf;">
<a:LOG rdf:ID="1038388896109-65082069">
  <a:lastChange rdf:resource="#1038388991156-1151000064"/>
</a:LOG>
<a:AddEntity rdf:ID="1038388991146-1199738740"
  a:firstChangeInAGroup="true"
  a:has_referenceInstance="http://wim.fzi.de/vision#Knowledge-Technology"
  a:inOIModel="http://wim.fzi.de/vision"
  a:version="1"/>
<a:AddPropertyInstance rdf:ID="1038388991156-1151000064"
  a:has_referenceProperty="http://kaon.semanticweb.org/2001/11/kaon-lexical#inLanguage"
  a:has_referenceSourceInstance="http://wim.fzi.de/vision#1038388986470-1509637582"
  a:has_referenceTargetInstance="http://kaon.semanticweb.org/2001/11/kaon-lexical#en"
  a:inOIModel="http://wim.fzi.de/vision"
  a:version="1">
  <a:has_previousChange rdf:resource="#1038388991156-236494706"/>
</a:AddPropertyInstance>
<a:AddInstanceOf rdf:ID="1038388991156-1318064932"
  a:has_referenceConcept="http://wim.fzi.de/vision#Topic"
  a:has_referenceInstance="http://wim.fzi.de/vision#Knowledge-Technology"
  a:inOIModel="http://wim.fzi.de/vision"
  a:version="1">
  <a:has_previousChange rdf:resource="#1038388991146-1199738740"/>
</a:AddInstanceOf>
<a:AddEntity rdf:ID="1038388991156-1440264441"
  a:has_referenceInstance="http://wim.fzi.de/vision#1038388986470-1509637582"
  a:inOIModel="http://wim.fzi.de/vision"
  a:version="1">
  <a:has_previousChange rdf:resource="#1038388991156-1318064932"/>
</a:AddEntity>
<
```

And now the second step: The Deletion of the instance "Knowledge Technologies":

```
<a:RemovePropertyInstance rdf:ID="1038389363435-1077122541"
  a:has_referenceProperty="http://kaon.semanticweb.org/2001/11/kaon-lexical#inLanguage"
  a:has_referenceSourceInstance="http://wim.fzi.de/vision#1038388986470-1509637582"
  a:has_referenceTargetInstance="http://kaon.semanticweb.org/2001/11/kaon-lexical#en"
  a:inOIModel="http://wim.fzi.de/vision"
  a:version="2">
  <a:has_previousChange rdf:resource="#1038389363435-206021846"/>
</a:RemovePropertyInstance>
<a:RemoveInstanceOf rdf:ID="1038389363435-1145052166"
  a:has_referenceConcept="http://wim.fzi.de/vision#Topic"
  a:has_referenceInstance="http://wim.fzi.de/vision#Knowledge-Technology"
  a:inOIModel="http://wim.fzi.de/vision"
  a:version="2">
  <a:has_previousChange rdf:resource="#1038389363435-1981539391"/>
</a:RemoveInstanceOf>
<a:RemovePropertyInstance rdf:ID="1038389363435-1213234167"
  a:has_referenceProperty="http://kaon.semanticweb.org/2001/11/kaon-lexical#value"
  a:has_referenceSourceInstance="http://wim.fzi.de/vision#1038388986470-1509637582"
  a:has_referenceTargetObject="Knowledge Technology"
  a:inOIModel="http://wim.fzi.de/vision"
  a:version="2">
  <a:has_previousChange rdf:resource="#1038389363435-1077122541"/>
</a:RemovePropertyInstance>
```

---

## 2.2 Language parameters

Language parameters are to be found in the “View”-Menu. You can choose between English, German, French, Spanish, Arabic and Chinese.

## 2.3 Search

The Search function:

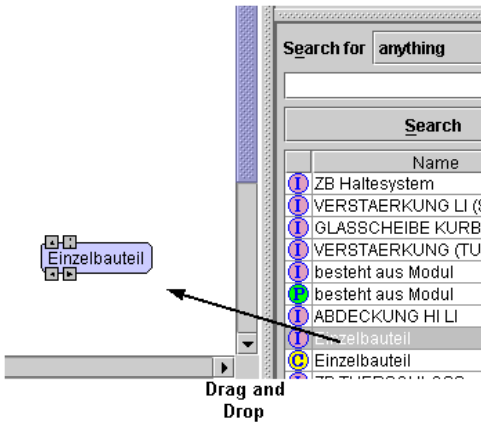


With the search function, you can easily find different named nodes.

It's possible to search for:

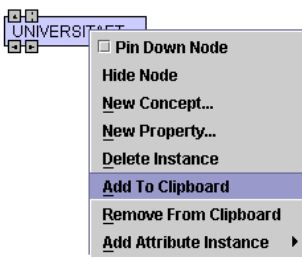
- anything: Every matching entity in the ontology will be displayed.
- concepts: Matching concepts will be displayed.
- instances: Matching instances will be displayed.
- properties: Matching properties will be displayed.

## The Paste function:

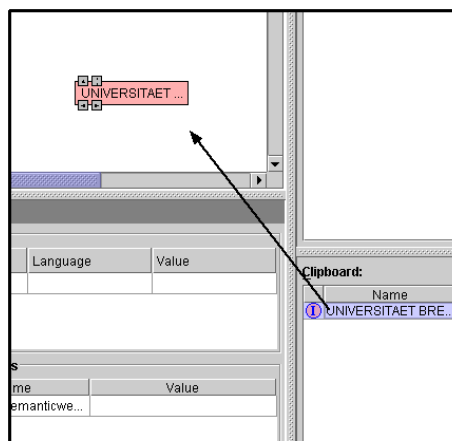


You can paste the selected node into the “Graph window” via drag & drop.

## 2.4 The Clipboard



The Clipboard is for copy & paste use. Just copy entities by clicking the right mouse button or by using the “view”-menu, and then choose the option “Add to Clipboard”. If you later need the entities once more just paste them back into the Ontology using drag and drop (see next pictures).



---

## 3. Working with OI-Models

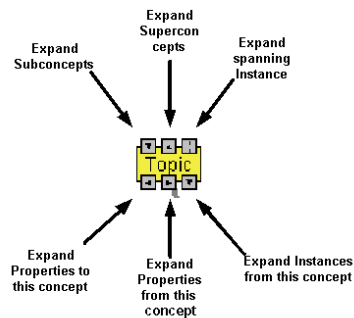
This section provides a detailed overview on how to work with OI-Models:

### 3.1 Creating a new OI-Model

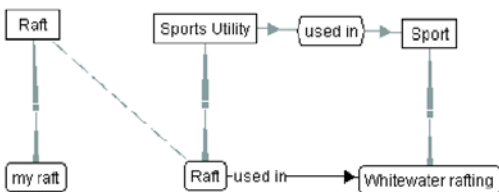
To work with an OI-Model, you can create a new ontology or open an existing one. From the file menu, select “Create New OI-Model” to create a new, empty ontology. To continue working on an existing ontology, select “Open OI-Model”. You can create, open and save ontologies on the local file system or an engineering server. To create a new OI-Model expand the “File”-Menu and click on “Create new OI-Model”. A Window pops up and you can define where to save the Ontology. (Details about the window see above at point 2.1 “The menus”).

The OI-Modeler provides different views on the Ontology and allows to inspect its components (concepts, instances, properties and lexicon). The graph in the upper section of the window shows the ontology entities and the connections between them.

Click on the arrows of an entity to expand its related entities:



**Expand spanning instances:** In real-world conceptual models, it is often unclear whether some element should be represented as a concept or as an instance. While developing a semantics-driven catalog system, relationships between sports utility types and individual sports utilities must be described. A possible conceptualization is to say that there is a Sports Utility concept representing the set of all types of sports utilities, with its elements representing particular type of sports utility (rafts, oars etc.) Assertions can be made about individual sport utility types (e.g. rafts are used for whitewater rafting). However, each type of sports utilities can be viewed as a set of individual instances (e.g. the raft in my garage is an instance in that set). Information about sports utility types is independent of information about particular instances (e.g. my raft can be broken). Hence, Sports Utility plays a dual role and can be interpreted as a concept and as an



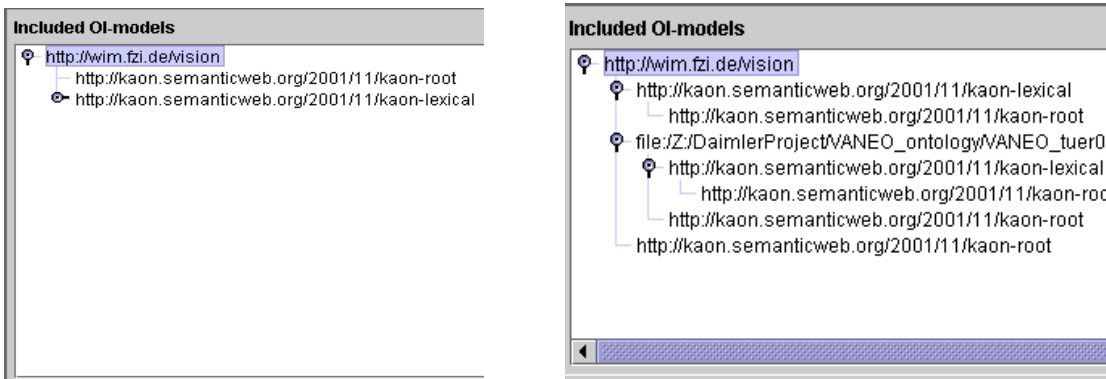
instance at the same time. Both interpretations of the Sports Utility are connected by the spanning object (the dashed line).

### 3.2 Including OI-Models

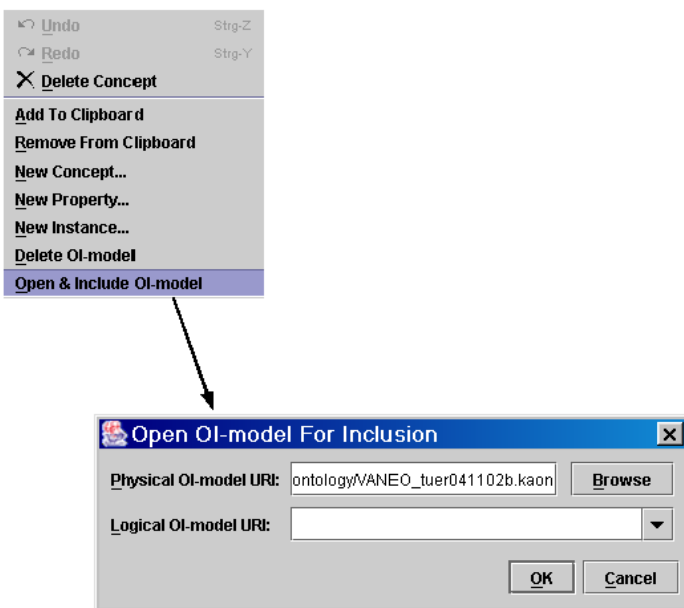
The OI-Modeler allows you to include ontologies. This means that you are able to combine two (or more) ontologies to one ontology.

An OI-Model always consists of two basic or system ontologies: The kaon-root and the kaon-lexical.

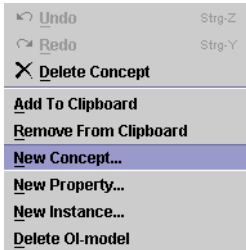
The following figures show the window “Included OI-models” before and after inclusion:



To include an OI-Model choose "open and include OI-Model" in the "Edit"-Menu. A new window opens and you can select the source of the OI-Model you want to include.



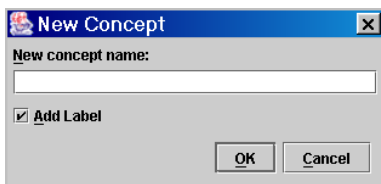
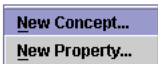
### 3.3 Creating concepts



There are three ways to create a concept:

1.) You can add new concepts by using the “Edit” menu (1<sup>st</sup> Picture), or 2.) by opening the context-menu (right mouse-button) in the graph window. (2<sup>nd</sup> picture), or 3.) by using the inspector.

(see point 3.8)

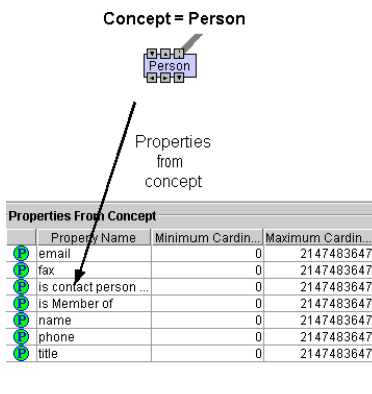


A new window appears and you can name the concept.

### 3.4 Creating properties

The procedure is almost the same as creating a concept, but the OI-Modeler differs between two kinds of properties:

#### 3.4.1 Properties from a concept:



These properties are relations to other concepts (instances) or attributes from this concept e.g. a description, long name, URL...

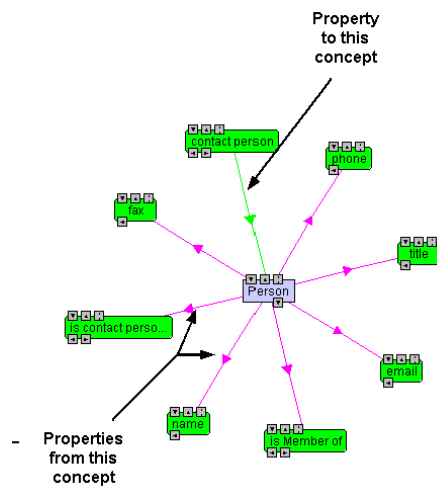
### 3.4.2 Properties to a concept

Property to concept

Property Name
<a href="http://kaon.semanticweb.org/2001/11/kaon-lexical#refere...">http://kaon.semanticweb.org/2001/11/kaon-lexical#refere...</a>
contact person

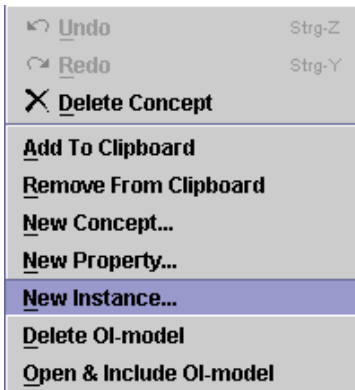
These properties are relations **from** other concepts **to** this concept. In this case the marked property is the inverse property to the one shown at point 3.3.1

In the following you can see the graph of a concept showing all its related properties:

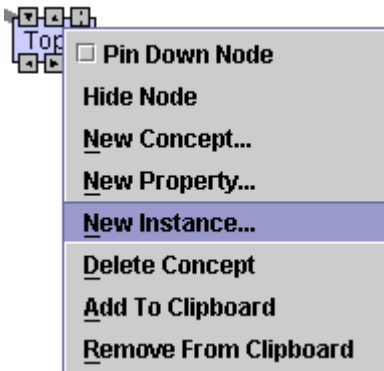


---

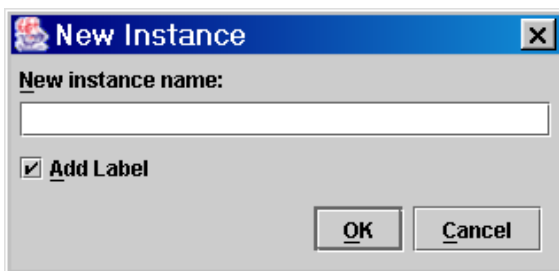
## 3.5 Creating instances



You can add an instance by using the “Edit”-menu (1<sup>st</sup> picture) or by using the context menu (right mouse button)(2<sup>nd</sup> picture). In both cases you first have to choose the concept you want to add the instance to.

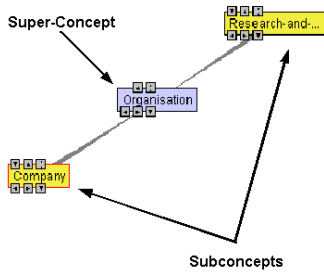


In both ways a new window appears and you can name the instance:



### 3.6 Sub-Concepts and Sub-Properties

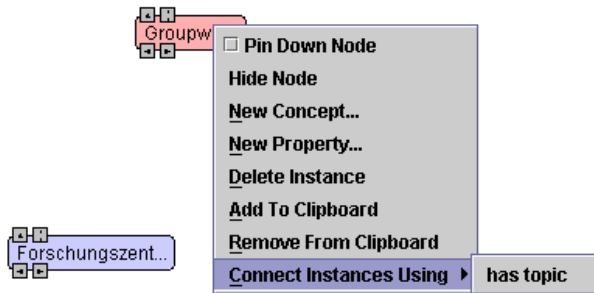
Sub-Concepts and Sub-Properties are thematic refinements of concepts or properties.



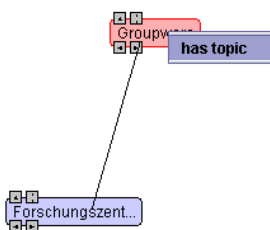
In the picture you can see two sub-concepts, which are combined to one super-concept. Thereby it's possible to create IS-A relations between concepts.

### 3.7 Creating properties between instances.

The properties between the instances are the relations between these instances. There are two different ways to connect instances through a property:



1.) As shown in the picture, you can click on the instance you want to connect, and then do a right click on the instance that ought to be connected.



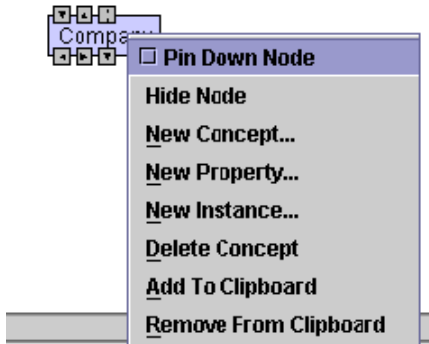
2.) It's also possible to press and hold the left mouse-button and then to drag the cursor to the instance you want to connect. A line appears, as shown in the picture and if you disengage the mouse-button, a menu appears and you can choose the property.

---

## 3.8 The Context menus of concepts, instances and properties

The OI-Modeler provides context sensitive context menus. That means the appearing context menu depends on the entity you have chosen and the entities on which you click the right mouse button.

### 3.8.1 The Context menu of concepts:



This window appears if you click the right mouse button over a concept. It offers you several options:

Hide Node: Hides the selected node.

New Concept, New Property; New Instance:

To create new entities related to the selected concept.

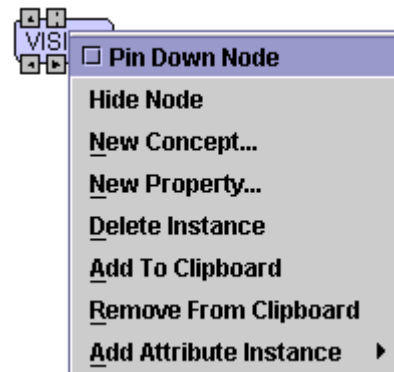
(In case of "new Concept" the new Concept would be a Sub Concept.)

Delete Concept: To delete the selected concept.

Add to Clipboard: To add the selected concept to the Clipboard (Details about the Clipboard see at Point 2.4).

Remove From Clipboard: Removes the selection from the Clipboard.

### 3.8.2 The Context menu of Instances



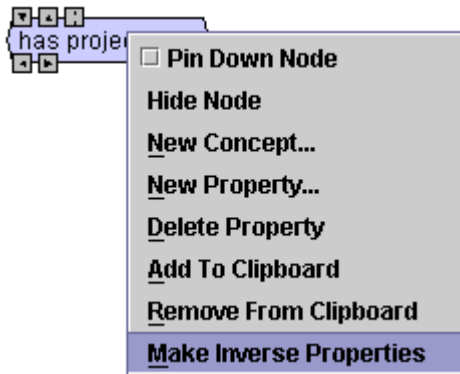
This window appears if you click the right mouse button over an instance. The options are almost the same as the concept

menu of concepts except the option **Add Attribute Instance:**

If you have defined attributes of instances (see 3.4.1) choose this function to specify the attributes.

---

### 3.8.3 The Context menu of Properties

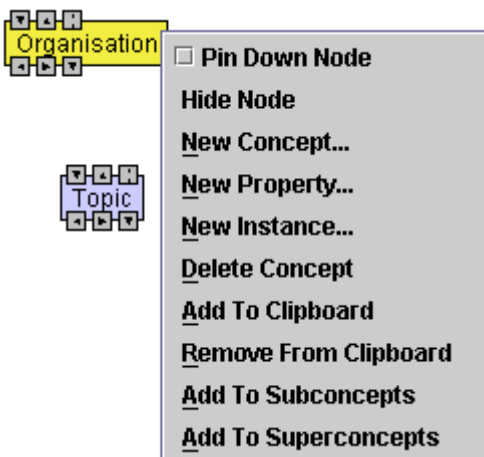


This window appears if you click the right mouse button over a property.

**New Property:** This option creates a sub-property of the chosen property.

**Make Inverse Properties:** This option creates inverse relations between properties.

### 3.8.4 The context menu between two concepts



To get this menu one first has to click on a concept (marked blue then), move the mouse cursor over another concept and click the right mouse button. There are two different options than in point 3.8.1:

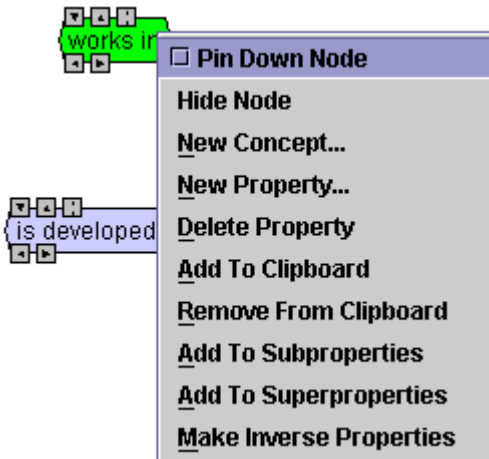
**Add to Subconcepts:** To make the concept the context menu appeared a subconcept of the marked (In this case make concept "Organisation" a subconcept of "Topic").

**Add to Superconcepts:** To make the marked concept a subconcept of the concept the context menu appeared (in this case make concept "Topic" a subconcept of "Organisation")

### 3.8.5 The context menu between two instances

Details and a figure see Point 3.7

### 3.8.6 The context menu between two properties



**Add to Subproperties:** Make property **works in** in subproperty of **is developed**.

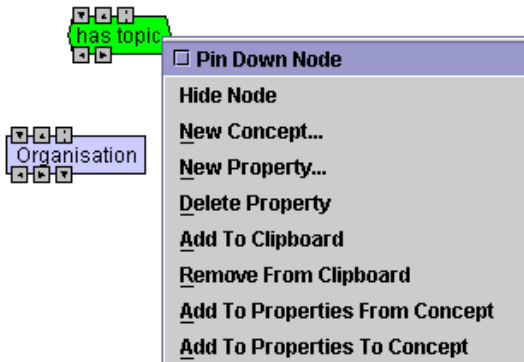
**Add to Superproperties:** Make property **is developed** subproperty of **works in**.

**Make Inverse Properties:** Make properties **is developed by** and **works in** inverse.

### 3.8.7 The context menu between Concepts and Properties

The following figure shows the first case:

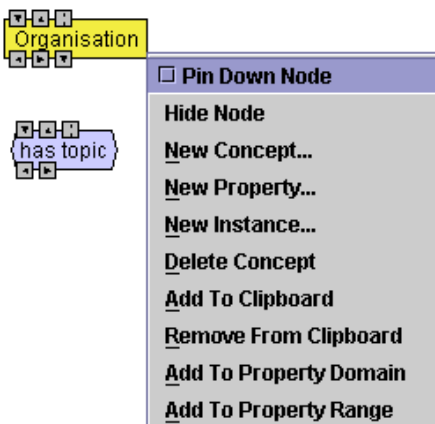
The concept is marked and the context menu appears over the property:



**Add to Properties from Concept:** Allows to add a Property from the Concept (see 3.4.1).

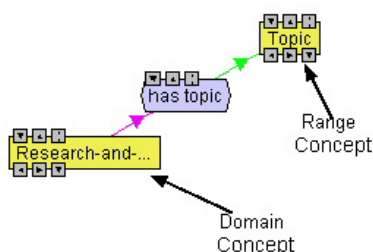
**Add to Properties to Concept:** Allows to add a Property to the Concept.

second case: the property is marked:



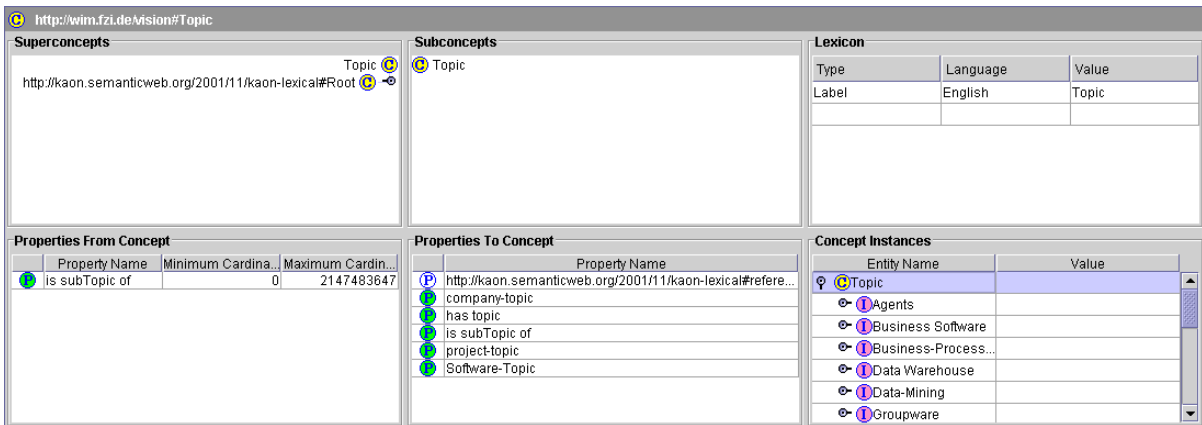
**Add to Property Domain:** This option allows to add a concept that is related **to** the property.

**Add to Property Range:** This option allows to add a concept that is related **from** the property.



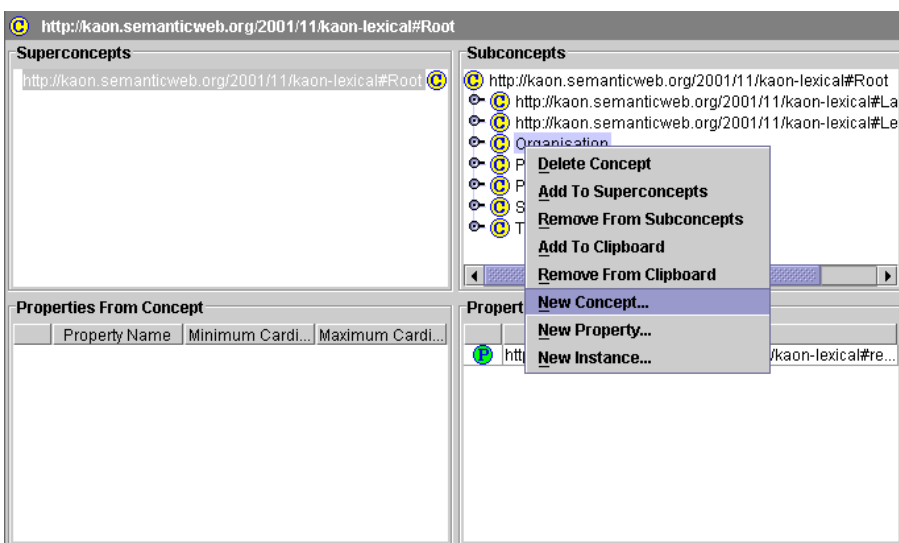
## 3.9 The Inspector of concepts, properties and instances

### The Inspector of concepts:



Here you can find all information about the concept, for instance the related sub - or super-concepts, the properties to and from the concept and the concept instances. It's possible to edit the different labels in the layer or to use the "drag and drop function" to paste the labels from the layer into the graph and it's possible to directly create new (sub-)concepts in the inspector. Because the inspector offers a view on the super - and subconcepts, you are able to choose the existing concept you want to extend with a new subconcept, push the right mouse button and select "new concept". The same way is possible if you want to add an instance or a property directly to a chosen concept.

The following figure shows you the inspector and the context menu with its specific options (details about the context menu see point 3.8.1)



## The Inspector of instances:

The screenshot shows the 'Inspector of instances' window for the URI <http://wim.fzi.de/vision/#Forschungszentrum-Informatik>. It is divided into three main sections:

- Parent Concepts:** A tree view showing the hierarchy of concepts: Forschungszentrum Informatik, Research-and-Education-Organisation, and http://kaon.semanticweb.org/2001/11/kaon-lexical/#Root.
- Property Instances:** A table listing various properties and their values for the instance.
- Lexicon:** A table showing the label, language, and value for the instance.

Entity Name	Value
Forschungszentrum Infor...	
Adress	Haid-und-Neu-Straße 10-14...
Country	Germany
develops software	KAON
E-Mail	http://wwwneu.fzi.de/kontaktf...
Fax	+49 (0) 721-9654-909
has related organisation	AIFB
has topic	Data Warehouse
has topic	Groupware
has topic	Knowledge management
has topic	Knowledge-Discovery
has topic	Natural-Language-Processi...
has topic	Ontology
has topic	Organisational learning
has tonic	Peer-2-Peer

Type	Language	Value
Label	English	Forschungszentru...

This layer provides you a view on the parent concepts of a single instance (left), all the related properties and attributes (middle), and the lexicon (right; see point 3.10)

## The Inspector of properties:

The screenshot shows the 'Inspector of properties' window for the URI <http://wim.fzi.de/vision/#isMemberOf>. It includes several sections:

- Options:** Checkboxes for  Attribute,  Symmetric, and  Transitive. An **Inverse property:** field is also present.
- Superproperties:** A list showing 'is Member of' with a green dot.
- Subproperties:** A list showing 'is Member of' with a green dot.
- Domain Concepts:** A table listing concepts in the domain.
- Range Concepts:** A table listing concepts in the range.
- Lexicon:** A table showing the label, language, and value for the property.

Concept Name	Minimum Cardin...	Maximum Cardin...
Person	0	2147483647

Concept Name
Organisation

Type	Language	Value
Label	English	is Member of

This layer shows you all information about a property:

Super- and sub-property relations.

The domain- and the range- concept (the domain concept shows the property from and the range the property to the concept).

You can also set some “properties of the property” like symmetric, transitive or inverse option.

The attribute function allows you to set properties without a range and allows you to set special information about an instance directly to the instance.

It's also possible to directly edit properties in the inspector using the context menu (right mouse button). (Details see 3.8.3 and following)

## 3.10 The lexical layer

### Modelling the Lexical Layer of Concepts, Attributes, Relations, and Instances

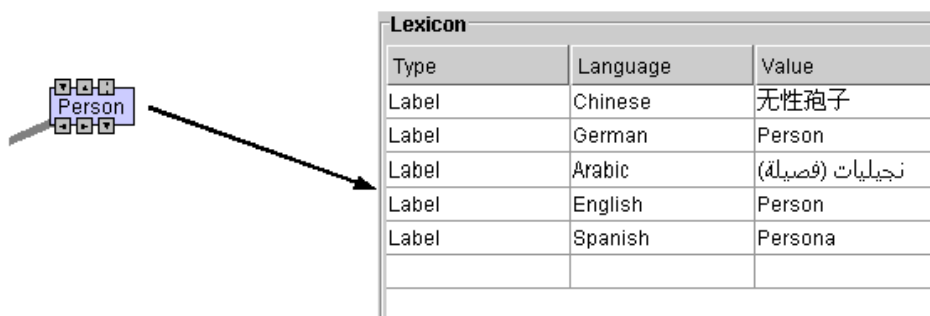
All ontological entities are considered as language neutral. On the lexical layer, lexical descriptions referring to different entities in the KAON representation vocabulary may be defined. A lexical entry is a lexicalisation of a concept, attribute, relation, and instance. Several types of lexical entries are defined. The standard lexical description are multilingual labels that may be used for the user's interface. **A label** is a specific kind of a lexical entry, describing a primary descriptor of an ontological or knowledge base entity. Another kind of lexical entries are morphologically reduced word **stems** that may be used by a natural language processing system. Figure 23 shows lexical elements available in OI-Modeler:



**A synonym** is a specific kind of a lexical entry, describing synonymous words for an ontological or knowledge base entity.

**The documentation** allows you to enter a text description of the ontological entity.

The lexical layer also allows you to create multilingual ontologies. As shown in the picture it is possible to label a concept (in this case the concept "Person") in different languages (OI-Modeler provides different language parameters, see 2.2 "Language parameters").



The diagram shows a concept node labeled "Person" with a small icon. An arrow points from this node to a table titled "Lexicon" which lists multilingual labels for the concept "Person".

Type	Language	Value
Label	Chinese	无性孢子
Label	German	Person
Label	Arabic	نجليات (فصيلة)
Label	English	Person
Label	Spanish	Persona

---

Lexicon		
Type	Language	Value
Label	English	Submodul
Label	<b>English</b> ▼	
	English	
	Portuguese	
	German	
	Arabic	
	Spanish	
	French	
Concept Instances	Chinese	

To define a concept in different languages, click on the concept, choose the language in the lexicon and then define it in the “value”-column in your chosen language.

---

## 4. References

B. Motik, A. Maedche and R. Volz : **A Conceptual Modeling Approach for building semantics-driven enterprise applications** . Proceedings of the First International Conference on Ontologies, Databases and Application of Semantics (ODBASE-2002), Springer, LNAI, California, USA.

(<http://kaon.semanticweb.org/Members/rvo/papers/odbase.pdf>)

L. Stojanovic, A. Maedche, B. Motik, N. Stojanovic : **User-driven Ontology Evolution Management** . Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management EKAW, Madrid, Spain, 2002

(<http://kaon.semanticweb.org/Members/maedche/1023194568.pdf>)

A. Maedche, B. Motik, L. Stojanovic, R. Studer and R. Volz : **Ontologies for Enterprise Knowledge Management** . IEEE Intelligent Systems, January/February 2003.

(<http://kaon.semanticweb.org/Members/rvo/papers/ieee-is-maedcheetal.pdf>)