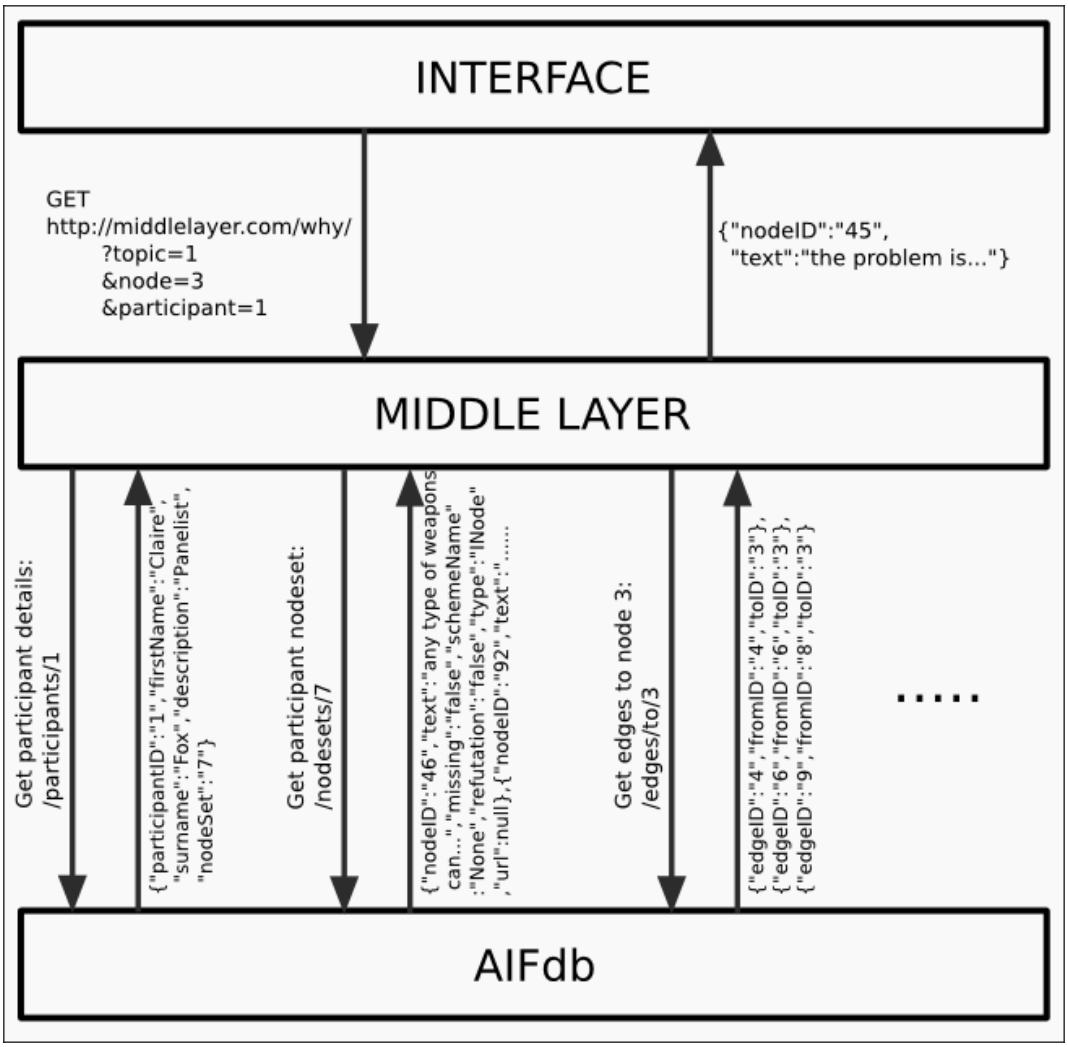

AIFdb

AIFdb is a database reification of AIF, allowing for the storage and retrieval of AIF compliant argument structures. At the lowest level, AIFdb consists of a database schema representing the elements available in AIF.

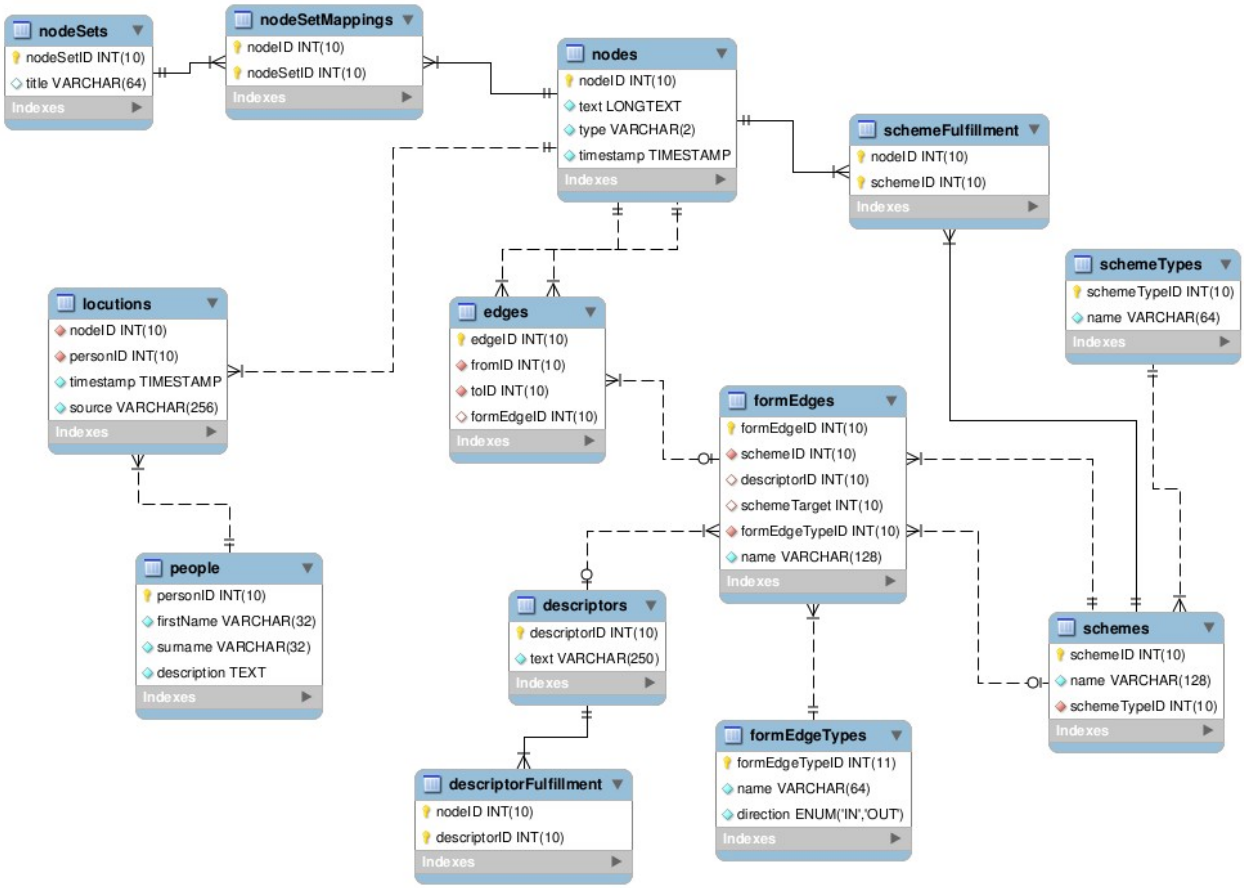
This schema can then be implemented in a range of databases, for example MySQL. On top of the database there are a range of web services written in PHP which allow for the addition and retrieval of AIF components from the database.

At a slightly higher level, further web services offer the ability to import and export argument data in a range of widely used formats including Carneades(LKIF), Rationale(RTNL), SVG, DOT, and RDF files.

Finally a “middle layer” exists which groups simple web service queries to allow for the execution of more complex interactions. For example, a query to determine what reasons a dialogue participant has for a particular point.



DB Schema


nodeSets
nodeSetID INT(10)
title VARCHAR(64)
Indexes
nodeSetMappings
nodeID INT(10)
nodeSetID INT(10)
Indexes
nodes
nodeID INT(10)
text LONGTEXT
type VARCHAR(2)
timestamp TIMESTAMP
Indexes
schemeFulfillment
nodeID INT(10)
schemeID INT(10)
Indexes
schemeTypes
schemeTypeID INT(10)
name VARCHAR(64)
Indexes
locutions
nodeID INT(10)
personID INT(10)
timestamp TIMESTAMP
source VARCHAR(256)
Indexes
edges
edgeID INT(10)
fromID INT(10)
toID INT(10)
formEdgeID INT(10)
Indexes
formEdges
formEdgeID INT(10)
schemeID INT(10)
descriptorID INT(10)
schemeTarget INT(10)
formEdgeTypeID INT(10)
name VARCHAR(128)
Indexes
people
personID INT(10)
firstName VARCHAR(32)
surname VARCHAR(32)
description TEXT
Indexes
descriptors
descriptorID INT(10)
text VARCHAR(250)
Indexes
schemes
schemeID INT(10)
name VARCHAR(128)
schemeTypeID INT(10)
Indexes
formEdgeTypes
formEdgeTypeID INT(11)
name VARCHAR(64)
direction ENUM('IN','OUT')
Indexes
descriptorFulfillment
nodeID INT(10)
descriptorID INT(10)
Indexes

Web Service Reference

Nodes

nodeID	int	numerical identifier
text	longtext	node text for an INode
type	INode/SNode	type of node (INode or SNode)
timestamp	timestamp	time node is added to the database

GET nodes/[NodeID]


```
{"nodeID":"1","text":"Britain should disarm","type":"I","timestamp":"2012-01-10
14:35:50"}
```


POST nodes/

Edges

edgeID	int	numerical identifier
fromID	int	nodeID of the node the edge comes from
toID	int	nodeID of the node the edge goes to
formEdgeID	int	ID of the formEdge that the edge fulfills

GET edges/[EdgeID]


```
{"edgeID":"1","fromID":"2","toID":"1","formEdgeID":null}
```


GET edges/to/[NodeID]


```
{"edges":[{"edgeID":"1","fromID":"2","toID":"1","formEdgeID":null},
{"edgeID":"3","fromID":"5","toID":"1","formEdgeID":null},
{"edgeID":"23","fromID":"25","toID":"1","formEdgeID":null}]}
```


GET edges/from/[NodeID]


```
{"edges":[{"edgeID":"1","fromID":"2","toID":"1","formEdgeID":null}]}
```


POST edges/

NodeSets

nodeSetID	int	numerical identifier

title	varchar(128)	title for the nodeset

GET nodesets/[NodeSetID]

```
{"nodeID":"1","text":"Britain should disarm","type":"I","timestamp":"2012-01-10
14:35:50"},{"nodeID":"2","text":"YA","type":"YA","timestamp":"2012-01-10
14:47:19"},{"nodeID":"3","text":"Rebecca says","type":"L","timestamp":"2012-01-
10 14:47:52"},...
```

GET nodesets/contains/[NodeID]

```
{"nodeSets":[{"nodeSetID":"1","title":"BBC Moral Maze Trident Debate"}]}
```

GET nodesets/new/

Creates a new nodeset and returns it's JSON represetation including the new nodeset ID.

POST nodesets/

NodeSetMappings

nodeSetID	int	numerical identifier for the nodeSet
nodeID	int	numerical identifier for the node

POST nodesetmappings/

Schemes

schemeID	int	numerical identifier
name	varchar(128)	title for the nodeset
schemeTypeID	int	ID for the type of scheme

GET schemes/[schemeID]

```
{"schemeID":"1","name":"Analogy","schemeTypeID":"1"}
```

GET schemes/all

```
{"schemes":[{"schemeID":"1","name":"Analogy","schemeTypeID":"1"},
{"schemeID":"2","name":"Bias","schemeTypeID":"1"},
{"schemeID":"3","name":"CausalSlipperySlope","schemeTypeID":"1"},...]}
```

POST schemes/search

```
{"schemeID":"1","name":"Analogy","schemeTypeID":"1"}
```

POST schemes/

SchemeFulfillment

nodeID	int	numerical identifier for the node
schemeID	int	numerical identifier for the scheme

POST schemefulfillment/

Descriptors

descriptorID	int	numerical identifier
text	varchar(250)	text for the descriptor

GET descriptors/[descriptorID]


```
{"descriptorID":"1","text":"Generally, case C1 is similar to case C2"}
```


POST descriptors/

DescriptorFulfillment

nodeID	int	numerical identifier for the node
descriptorID	int	numerical identifier for the descriptor

POST descriptorfulfillment/

FormEdges

formEdgeID	int	numerical identifier
schemeID	int	numerical identifier for the scheme
descriptorID	int	numerical identifier for the descriptor
schemeTarget	int	numerical identifier for the scheme target
formEdgeTypeID	int	numerical identifier for the formEdgeType
name	varchar(128)	name of the formEdge

GET formedges/[formEdgeID]

{"formEdgeID":"1","schemeID":"1","descriptorID":"1","schemeTarget":null,"formEdgeTypeID":"1","name":"SimilarityOfCases"}

GET formedges/schemes/[schemeID]

{"formedges":
[{"formEdgeID":"1","schemeID":"1","descriptorID":"1","schemeTarget":null,"formEdgeTypeID":"1","name":"SimilarityOfCases"},
{"formEdgeID":"2","schemeID":"1","descriptorID":"2","schemeTarget":null,"formEdgeTypeID":"2","name":"CaseOutcome"},...]}

POST formedges/

People

personID	int	numerical identifier
firstName	varchar(64)	first name for person
surname	varchar(64)	surname for person
description	text	short biographical description

GET people/[personID]

{"personID":"1","firstName":"Melanie","surname":"Philips","description":null}

POST people/

Locutions

nodeID	int	numerical identifier for the node
personID	varchar(64)	numerical identifier for the person
timestamp	varchar(64)	time of the locution
source	text	source (URL) of the locution

GET locutions/[personID]

{"locutions":[{"nodeID":"20","text":"Britain and America would only use nuclear weapons for defence","timestamp":"2012-01-10 16:30:38","source":""},
{"nodeID":"16","text":"There is a difference between Britain and other countries (e.g. Iran)","timestamp":"2012-01-10 16:27:41","source":""},...]}

POST locutions/

Import/Export

LKIF (Carneades)

GET:
http://AIFdb_URL/lkif/[nodeSet ID]
e.g.
http://www.arg.dundee.ac.uk/AIFdb/lkif/861

POST:
http://AIFdb_URL/lkif/
e.g.
curl -i -X POST -u user:password -F file=@analysis.lkif http://www.arg.dundee.ac.uk/AIFdb/lkif/

RTNL (Rationale)

GET:
http://AIFdb_URL/rtnl/[nodeSet ID]
e.g.
http://www.arg.dundee.ac.uk/AIFdb/rtnl/861

POST:
http://AIFdb_URL/rtnl/
e.g.
curl -i -X POST -u user:password -F file=@analysis.rtnl http://www.arg.dundee.ac.uk/AIFdb/rtnl/

RDF

GET:
http://AIFdb_URL/rdf/[nodeSet ID]
e.g.
http://www.arg.dundee.ac.uk/AIFdb/rdf/861

POST:
http://AIFdb_URL/rdf/
e.g.
curl -i -X POST -u user:password -F file=@analysis.rdf http://www.arg.dundee.ac.uk/AIFdb/rdf/

DOT

GET:
http://AIFdb_URL/dot/[nodeSet ID]
e.g.
http://www.arg.dundee.ac.uk/AIFdb/dot/861

JSON

GET:
http://AIFdb_URL/json/[nodeSet ID]
e.g.
http://www.arg.dundee.ac.uk/AIFdb/json/861

POST:
http://AIFdb_URL/json/
e.g.
curl -i -X POST -u user:password -F file=@analysis.json http://www.arg.dundee.ac.uk/AIFdb/json/

Diagram

PNG:
http://AIFdb_URL/diagram/[nodeSet ID]
e.g.
http://www.arg.dundee.ac.uk/AIFdb/diagram/861

SVG:
http://AIFdb_URL/diagram/svg/[nodeSet ID]
e.g.
http://www.arg.dundee.ac.uk/AIFdb/diagram/svg/861