MIAKT Component Services

19th November 2004

Contents

Introduction

This document describes the component services that are available in the MIAKT application, in their various forms. These services are called via the enterprise server by the client application, or by other services on the enterprise server.

SOAP services

The following list provides an overview of the services that are provided via a SOAP webservice interface for the MIAKT application.

Patient Database

Provider: University of Southampton
API: 3Store API
Author: David Dupplaw

This service is provided by the University of Southampton and is a generic 3store web-service interface API. The data stored in the knowledge-base is the patient data.

performQuery	This method provides a means for performing a query on the knowledge-base using RDQL. This allows retrieval of patient-related data.

Input 1	*String_1* – ***String (RDQL)*** The RDQL to execute on this knowledge-base.
Input 2	*int_2* – ***Integer (First Record)*** The index of the first record to return in the return set.
Input 3	*Int_3* – ***Integer (Last Record)***

<table>
<tr><td></td><td><table>
<tr><td></td><td>The index of the last record to return in the return set.</td></tr>
<tr><td>Input 4</td><td>String_4 – String (Database)
The name of the database into which this triple should be asserted.</td></tr>
<tr><td>Output</td><td>result – String
The result set is returned as an XML document where the elements in the document will depend on the RDQL query. Each record will be returned as a set of <record> tags, enclosed in a <record-set> envelope. The elements within each record will be named by the return variables stated within the RDQL query, with the body of each element containing the resultant value.</td></tr>
</table></td></tr>
<tr><td>assertTriple</td><td>This method provides a means for asserting a new triple into the knowledge-base. This allows updating of patient-related data.

There is no useful output to this method.

<table>
<tr><td>Input 1</td><td>String_1 – String (Object 1)
The URL of the source object in this new relationship.</td></tr>
<tr><td>Input 2</td><td>String_2 – String (Predicate)
The URL of the predicate in this new relationship.</td></tr>
<tr><td>Input 3</td><td>String_3 – String (Object 2)
The URL of the destination object in this new relationship.</td></tr>
<tr><td>Input 4</td><td>String_4 – String (Database)
The name of the database into which this triple should be asserted.</td></tr>
<tr><td>Input 5</td><td>Int_5 – Integer (Type)
The type of the assertion. If the destination object (in input 3) is a literal, then this input should be equal to 0. Any other value will indicate that the destination object is a URL.</td></tr>
</table></td></tr>
</table>

Natural Language Generation

Provider: University of Sheffield
Author: Kalina Bontcheva

This service is provided by the University of Sheffield and generates a natural language summarisation of an RDF description of a patient's details.

init	The functionality of this method is undefined, and it should not be called remotely. *This method has no inputs or outputs.*
ping	This method provides a means for asserting that the service is accepting connections.

	Output	*pingReturn* – ***String*** Will return the string "Service is Alive!" if the service is accepting connections.
generateText	This method provides a means for generating the natural language summerisation of the input RDF file.	
	Input 1	*rdfFile* – ***String*** The RDF summerisation of the patient's data.
	Output	*generateTextReturn* – ***String*** Returns an plain text string containing the natural language summerisation of the input RDF file.

UMLS Lookup

Provider: University of Sheffield
Author: Kalina Bontcheva

This service is provided by the University of Sheffield for providing the definitions of medical terms by redirection to the Unified Medical Language System (UMLS).

init	The functionality of this method is undefined, and it should not be called remotely. *This method has no inputs or outputs.*	
ping	This method provides a means for asserting that the service is accepting connections.	
	Output	*pingReturn* – ***String*** Will return the string "Service is Alive!" if the service is accepting connections.
lookupTerm	This method provides a means for looking up a term in the UMLS.	
	Input 1	*theTerm* – ***String*** This input provides the term to lookup in the UMLS.
	Output	*lookupTermReturn* – ***String*** Will return an XML document containing the definitions of the given term, including their sources.

Image Analysis (Southampton)

Provider: University of Southampton
API: FeatureService API
Author: David Dupplaw

This service provided by the University of Southampton provides the functionality for analysing images.

The following feature modules are available on this service:

Monochrome Histogram **Author:** David Dupplaw	Generates a monochrome histogram of an IIP image. The features generated consist of the frequency of pixels in each histogram bin. Comparison functions are currently limited.
String Matching **Author:** David Dupplaw	Provides a means for matching strings to determine whether they're the same or not. The features generated are simply the strings passed in.

ping	This function simply returns "FeatureService is Alive!" so that the service can be tested. Output: *result* – ***String*** Will return the string "FeatureService is Alive!" if the service is accepting connections.
getNumberOfAvailableFeatureModules	Returns the number of feature modules available on this service. Output: *result* – ***String*** The number of feature modules available on this service.
getAvailableFeatureModules	Returns an XML document containing the definition and abilities of the feature modules available on this service. Output: *result* – ***String*** An XML document containing the details about the feature modules on this service.
getFeatureModuleAt	Returns an XML document containing the definition and abilities of the feature module at the given zero-based index. If the index is greater or equal to the result of the ***getNumberOfFeatureModules*** method, an error message will be returned. Input: *int_1* – ***Integer*** The index of the feature module to retrieve the details of. Output: *result* – ***String***

<table>
<tr><td></td><td><table>
<tr><td></td><td>An XML document containing the details about the feature module at the given index, or an error message if the index is out of range.</td></tr>
</table></td></tr>
<tr><td>compareFeatureVectors</td><td>Returns a distance measure between the two feature vectors identified by the given inputs.

<table>
<tr><td>Input 1</td><td>String_1 – String
The identifier of the first feature vector to compare.</td></tr>
<tr><td>Input 2</td><td>String_2 – String
The identifier of the second feature vector to compare against the first.</td></tr>
<tr><td>Output</td><td>result – String
A distance measure (0 means the feature vectors are the same, and, ideally, 1 means the feature vectors are entirely different).</td></tr>
</table></td></tr>
<tr><td>generateFeatureVector</td><td>Generates a feature vector for some media specified in the source input. The feature vector will be stored in a database at the service end, and an identifier for that feature vector will be returned.

<table>
<tr><td>Input 1</td><td>String_1 – String
The feature module to use to generate the feature vectors.</td></tr>
<tr><td>Input 2</td><td>String_2 – String
An XML document containing the information about the source of the media from which to generate a feature vector.</td></tr>
<tr><td>Output</td><td>result – String
The ID of the generated feature vector.</td></tr>
</table></td></tr>
<tr><td>getFeatureVector</td><td>Get the feature vector identified by the given identifier.

<table>
<tr><td>Input</td><td>String_1 – String
The identifier of the feature vector to retrieve.</td></tr>
<tr><td>Output</td><td>result – String
A representation of the</td></tr>
</table></td></tr>
</table>

		feature vector.

Image Analysis (Oxford)

Provider: University of Oxford
API: FeatureService API
Author: David Dupplaw and Maud Poissonnier

This service provided by the University of Southampton provides the functionality for analysing images.

The following feature modules are available on this service:

Dickinson Shape Analysis Feature **Author:** James Dickinson and Maud Poissonnier	*Unknown – Currently Unavailable*

ping	This function simply returns "FeatureService is Alive!" so that the service can be tested. Output: *result* – ***String*** Will return the string "FeatureService is Alive!" if the service is accepting connections.
getNumberOfAvailableFeatureModules	Returns the number of feature modules available on this service. Output: *result* – ***String*** The number of feature modules available on this service.
getAvailableFeatureModules	Returns an XML document containing the definition and abilities of the feature modules available on this service. Output: *result* – ***String*** An XML document containing the details about the feature modules on this service.
getFeatureModuleAt	Returns an XML document containing the definition and abilities of the feature module at the given zero-based index. If the index is greater or equal to the result of the ***getNumberOfFeatureModules*** method, an error message will be returned.

<table>
<tr><td></td><td><table>
<tr><td>Input</td><td>int_1 – Integer
The index of the feature module to retrieve the details of.</td></tr>
<tr><td>Output</td><td>result – String
An XML document containing the details about the feature module at the given index, or an error message if the index is out of range.</td></tr>
</table></td></tr>
<tr><td>compareFeatureVectors</td><td>Returns a distance measure between the two feature vectors identified by the given inputs.

<table>
<tr><td>Input 1</td><td>String_1 – String
The identifier of the first feature vector to compare.</td></tr>
<tr><td>Input 2</td><td>String_2 – String
The identifier of the second feature vector to compare against the first.</td></tr>
<tr><td>Output</td><td>result – String
A distance measure (0 means the feature vectors are the same, and, ideally, 1 means the feature vectors are entirely different).</td></tr>
</table></td></tr>
<tr><td>generateFeatureVector</td><td>Generates a feature vector for some media specified in the source input. The feature vector will be stored in a database at the service end, and an identifier for that feature vector will be returned.

<table>
<tr><td>Input 1</td><td>String_1 – String
The feature module to use to generate the feature vectors.</td></tr>
<tr><td>Input 2</td><td>String_2 – String
An XML document containing the information about the source of the media from which to generate a feature vector.</td></tr>
<tr><td>Output</td><td>result – String
The ID of the generated feature vector.</td></tr>
</table></td></tr>
<tr><td>getFeatureVector</td><td>Get the feature vector identified by the given identifier.</td></tr>
</table>

	Input	*String_1* – ***String*** The identifier of the feature vector to retrieve.
	Output	*result* – ***String*** A representation of the feature vector.

Image Registration

Provider: Kings College London
Author: Yalin Zheng

This service is provided by Kings College London, and provides the functionality for registration of MRI images. This service calls a GRID service which operates on a cluster of computers in a demilitarised zone at KCL.

This service can ONLY be called from the Southampton enterprise server due to firewall regulations.

Currently the MRI images have to be local to the registration cluster, so therefore only the images that already exist on those machines can be registered. However, these images can be retrieved using the image passthrough servlet on the Southampton enterprise server.

rigid	This method provides a means for rigid registration of MRI images.	
	Input	*remoteHost* ***- String***
	Input	*user* – ***String***
	Input	*remotePort* – ***String***
	Input	sourceImage – ***String***
	Input	*region* – ***String***
	Input	*interpolation* – ***String***
	Input	*remoteDir* – ***String***
	Input	*parameterFile* – ***String***
	Input	*targetImage* – ***String***
	Output	*rigidReturn* – ***String*** Returns an XML document containing the identifier of the submitted job.
nonRigid	This method provides a means for non-rigid registration of MRI images.	
	Input	*remoteHost* ***- String***
	Input	*user* – ***String***
	Input	*remotePort* – ***String***
	Input	sourceImage – ***String***
	Input	*region* – ***String***
	Input	*interpolation* – ***String***

<table>
<tr><td></td><td><table>
<tr><td>Input</td><td>remoteDir – String</td></tr>
<tr><td>Input</td><td>parameterFile – String</td></tr>
<tr><td>Input</td><td>targetImage – String</td></tr>
<tr><td>Input</td><td>controlPointSpacing – String</td></tr>
<tr><td>Output</td><td>nonRigidReturn – String
Returns an XML document containing the identifier of the submitted job.</td></tr>
</table></td></tr>
<tr><td>queryJobStatus</td><td>This method provides a means for retrieving the job status of a GRID job that has been previously submitted.

<table>
<tr><td>Input</td><td>query – String
An XML document containing the identifier of the query to retrieve the status of.</td></tr>
<tr><td>Output</td><td>queryJobStatusReturn – String
Returns an XML document containing the status of the given job.</td></tr>
</table></td></tr>
</table>

MRI Lesion Classification

Provider: Kings College London
Author: Christine Tanner

This service is provided by Kings College London, and provides the functionality for classification of an identified 2.5D region of interest within an MRI image.

This service can ONLY be called from the Southampton enterprise server due to firewall regulations.

Currently the MRI images have to be local to the service, so therefore only the images that already exist on those machines can be used as input for this service. However, these images can be retrieved using the image passthrough servlet on the Southampton enterprise server.

Naïve Bayes Classification

Provider: University of Southampton
Author: Bo Hu

This service provided by the University of Southampton, provides a means for classifying a region of interest from the semantic concepts representing its type and shape, etc. to a malignant or benign finding. The classification is based on the data provided in the University of South Florida's Mammogram Database.

Note, this service is not thread safe and should only ever be called serially by one client.

loadTranningData	This method primes the service by loading the pre-stored training data.
inferencingWithExistingNetwork	Performs classification.
inferencingWithExistingNetwork1	Performs classification
inferencingWithExistingNetwork2	Performs classification
inferencingWithExistingNetwork22	Performs classification
inferencingWithExistingNetwork3	Performs classification
isCalcification	Return whether the lesion is a calcification.
isMass	Return whether the lesion is a mass.
setFeature	Set a particular feature on the lesion.
setLesionFlag	Set the flag of which type of lesion this is – calcification or mass.

Settings Database

Provider: University of Southampton
API: 3Store API
Author: David Dupplaw

This service is provided by Southampton University and contains the settings for the various applications. This is used mainly by the enterprise server for retrieving the various settings for which services to populate the task registry with.

<table>
<tr><td>performQuery</td><td>This method provides a means for performing a query on the knowledge-base using RDQL. This allows retrieval of patient-related data.

<table>
<tr><td>Input 1</td><td>String_1 – String (RDQL)
The RDQL to execute on this knowledge-base.</td></tr>
<tr><td>Input 2</td><td>int_2 – Integer (First Record)
The index of the first record to return in the return set.</td></tr>
<tr><td>Input 3</td><td>Int_3 – Integer (Last Record)
The index of the last record to return in the return set.</td></tr>
<tr><td>Input 4</td><td>String_4 – String (Database)
The name of the database into which this triple should be asserted.</td></tr>
<tr><td>Output</td><td>result – String
The result set is returned as an XML document where the elements in the document will depend on the RDQL query. Each record will be returned as a set of <record> tags, enclosed in a <record-set> envelope. The elements</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td></td><td>within each record will be named by the return variables stated within the RDQL query, with the body of each element containing the resultant value.</td></tr></table></td></tr>
<tr><td>assertTriple</td><td>This method provides a means for asserting a new triple into the knowledge-base. This allows updating of patient-related data.

There is no useful output to this method.

<table><tr><td>Input 1</td><td>String_1 – String (Object 1)
The URL of the source object in this new relationship.</td></tr><tr><td>Input 2</td><td>String_2 – String (Predicate)
The URL of the predicate in this new relationship.</td></tr><tr><td>Input 3</td><td>String_3 – String (Object 2)
The URL of the destination object in this new relationship.</td></tr><tr><td>Input 4</td><td>String_4 – String (Database)
The name of the database into which this triple should be asserted.</td></tr><tr><td>Input 5</td><td>Int_5 – Integer (Type)
The type of the assertion. If the destination object (in input 3) is a literal, then this input should be equal to 0. Any other value will indicate that the destination object is a URL.</td></tr></table></td></tr>
</table>

IRS services

The following list provides an overview of the services that are provided via the internet reasoning service for the MIAKT application. These services are semantically marked up, and are accessed through an IRS API. The enterprise server uses the Java API.

Villapark

Server: http://villapark.open.ac.uk:3000/
Ontology: miakt-matching-task
Username: guest
Password: guest
Author: John Domingue, Liliana Cabral, Dyanesh

This service is provided by the Open University and provides a set of services that perform retrieval on a duplicate set of knowledge that is stored at the Open University.

find-images-matching-abno-by-finding-task	
find-abnos-matching-abno-by-overall-impression-task	
find-images-matching-image-by-finding-task	
find-abnos-matching-abno-by-same-morpho-task	
find-images-matching-image-by-morpho-feature-task	
find-abnos-matching-abno-by-finding-task	
find-image-files-matching-patient-task	

find-patients-matching-patient-by-age-task	
find-image-files-matching-abno-by-finding-task	
find-images-matching-image-by-view-task	
find-taps-matching-tap-by-patient-age-task	
find-images-matching-abno-by-overall-impression-task	
find-metarois-matching-metaroi-by-distribution-task	
find-image-files-matching-abno-by-overall-impression-task	