ModuloRecommender Reference Manual

1.0.0

Generated by Doxygen 1.5.2

Sat Nov 13 18:41:21 2010

Contents

Chapter 1

ModuloRecommender Namespace Documentation

1.1 br Namespace Reference

Namespaces

- namespace **pucrio**
- namespace **ufscar**

1.2 br::pucrio Namespace Reference

Namespaces

- namespace **telemidia**

1.3 br::pucrio::telemidia Namespace Reference

Namespaces

- namespace **ginga**

1.4 br::pucrio::telemidia::ginga Namespace Reference

Namespaces

- namespace **core**
- namespace **ncl**

1.5 br::pucrio::telemidia::ginga::core Namespace Reference

Namespaces

- namespace **contextmanager**

1.6 br::pucrio::telemidia::ginga::core::contextmanager Namespace Reference

Classes

- class **ContextManager**
- class **TVData**
- class **DataInterface**
- class **IContextManager**
- class **ImplicitContextManager**
- class **ImplicitProfiles**
- class **ISystemInfo**
- class **SystemInfo**
- class **GingaUser**
- class **IGingaUser**

Typedefs

- typedef pair< time_t, vector< int > * > **tIProfileObject**
- typedef map< int, **tIProfileObject** > **tIProfileMap**
- typedef map< int, **tIProfileMap** * > **tIProfiles**

Functions

- void * **run_Thread** (void *p)

1.6.1 Typedef Documentation

1.6.1.1 typedef pair<time_t, vector<int>*> tIProfileObject

! Profile type: timestamp and feature vector.

Definition at line 105 of file ImplicitProfiles.h.

1.6.1.2 typedef map<int, tIProfileObject> tIProfileMap

! Profile map type: map for profiles of a user.

Definition at line 108 of file ImplicitProfiles.h.

1.6.1.3 typedef map<int, tIProfileMap*> tIProfiles

! Profiles type: all the users and their profiles..

Definition at line 111 of file ImplicitProfiles.h.

1.7 br::pucrio::telemidia::ginga::ncl Namespace Reference

Classes

- class **FormatterMediator**
- class **FormatterScheduler**
- class **GingaNclGfx**
- class **IFormatterSchedulerListener**
- class **IPrivateBaseManager**
- class **PrivateBaseContext**
- class **PrivateBaseManager**

Namespaces

- namespace **adaptation**
- namespace **adapters**
- namespace **animation**
- namespace **emconverter**
- namespace **focus**
- namespace **model**
- namespace **multidevice**
- namespace **prefetch**

1.8 br::pucrio::telemidia::ginga::ncl::adaptation Namespace Reference

Namespaces

- namespace **context**

1.9 br::pucrio::telemidia::ginga::ncl::adaptation::context Namespace Reference

Classes

- class **AuthenticationServer**
- class **ContextBase**
- class **PresentationContext**
- class **RuleAdapter**

Functions

- void ∗ **runAuthenticationServer** (void ∗p)

1.10 br::pucrio::telemidia::ginga::ncl::adapters Namespace Reference

Classes

- class **FormatterPlayerAdapter**
- class **INclEditListener**
- class **IPlayerAdapter**
- class **IPlayerAdapterManager**
- class **NominalEventMonitor**
- class **PlayerAdapterManager**

Namespaces

- namespace **application**
- namespace **av**
- namespace **image**
- namespace **remote**
- namespace **text**

Variables

- static IComponentManager * **cm** = IComponentManager::getCMInstance()

1.11 br::pucrio::telemidia::ginga::ncl::adapters::application Namespace Reference

Classes

- class **ApplicationPlayerAdapter**

Namespaces

- namespace **ncl**
- namespace **xhtml**
- namespace **lua**

1.12 br::pucrio::telemidia::ginga::ncl::adapters::application::lua Namespace Reference

Classes

- class **LuaPlayerAdapter**

1.13 br::pucrio::telemidia::ginga::ncl::adapters::application::ncl Namespace Reference

Classes

- class **NCLPlayerAdapter**

1.14 br::pucrio::telemidia::ginga::ncl::adapters::application::xhtml Namespace Reference

Classes

- class **LinksPlayerAdapter**

1.15 br::pucrio::telemidia::ginga::ncl::adapters::av Namespace Reference

Classes

- class **AVPlayerAdapter**
- class **ChannelPlayerAdapter**

Namespaces

- namespace **tv**

1.16 br::pucrio::telemidia::ginga::ncl::adapters::av::tv Namespace Reference

Classes

- class **ProgramAVPlayerAdapter**

1.17 br::pucrio::telemidia::ginga::ncl::adapters::image Namespace Reference

Classes

- class **ImagePlayerAdapter**

1.18 br::pucrio::telemidia::ginga::ncl::adapters::remote Namespace Reference

Classes

- class **RemotePlayerAdapter**

1.19 br::pucrio::telemidia::ginga::ncl::adapters::text Namespace Reference

Classes

- class **PlainTxtPlayerAdapter**
- class **SubtitlePlayerAdapter**

1.20 br::pucrio::telemidia::ginga::ncl::animation Namespace Reference

Classes

- class **AnimationController**

1.21 br::pucrio::telemidia::ginga::ncl::emconverter Namespace Reference

Classes

- class **FormatterConverter**
- class **FormatterLinkConverter**
- class **ObjectCreationForbiddenException**

1.22 br::pucrio::telemidia::ginga::ncl::focus Namespace Reference

Classes

- class **FormatterFocusManager**

1.23 br::pucrio::telemidia::ginga::ncl::model Namespace Reference

Namespaces

- namespace **event**
- namespace **components**
- namespace **presentation**
- namespace **time**
- namespace **switches**
- namespace **link**

1.24 br::pucrio::telemidia::ginga::ncl::model::components Namespace Reference

Classes

- class **ApplicationExecutionObject**
- class **CompositeExecutionObject**
- class **ExecutionObject**
- class **NodeNesting**

1.25 br::pucrio::telemidia::ginga::ncl::model::event Namespace Reference

Classes

- class **AnchorEvent**
- class **AttributionEvent**
- class **FormatterEvent**
- class **IAttributeValueMaintainer**
- class **IEventListener**
- class **IFormatterEvent**
- class **PresentationEvent**
- class **SelectionEvent**

Namespaces

- namespace **transition**

1.26 br::pucrio::telemidia::ginga::ncl::model::event::transition Namespace Reference

Classes

- class **BeginEventTransition**
- class **EndEventTransition**
- class **EventTransition**
- class **EventTransitionManager**
- class **TransitionDispatcher**

1.27 br::pucrio::telemidia::ginga::ncl::model::link Namespace Reference

Classes

- class **FormatterCausalLink**
- class **FormatterLink**
- class **LinkAction**
- class **LinkActionListener**
- class **LinkActionProgressionListener**
- class **LinkAndCompoundTriggerCondition**
- class **LinkAssessment**
- class **LinkAssessmentStatement**
- class **LinkAssignmentAction**
- class **LinkAttributeAssessment**
- class **LinkCompoundAction**
- class **LinkCompoundStatement**
- class **LinkCompoundTriggerCondition**
- class **LinkCondition**
- class **LinkListener**
- class **LinkRepeatAction**
- class **LinkSimpleAction**
- class **LinkStatement**
- class **LinkTransitionTriggerCondition**
- class **LinkTriggerCondition**
- class **LinkTriggerListener**
- class **LinkValueAssessment**

1.28 br::pucrio::telemidia::ginga::ncl::model::presentation Namespace Reference

Classes

- class **CascadingDescriptor**
- class **FormatterDeviceRegion**
- class **FormatterLayout**
- class **FormatterRegion**

Namespaces

- namespace **focus**

1.29 br::pucrio::telemidia::ginga::ncl::model::presentation::focus Namespace Reference

Classes

- class **FocusSourceManager**

1.30 br::pucrio::telemidia::ginga::ncl::model::switches Namespace Reference

Classes

- class **ExecutionObjectSwitch**
- class **SwitchEvent**

1.31 br::pucrio::telemidia::ginga::ncl::model::time Namespace Reference

Classes

- class **CostFunctionDuration**
- class **DurationFactory**
- class **FlexibleTimeMeasurement**
- class **LinearCostFunctionDuration**
- class **TimeMeasurement**
- class **UnflexibleDuration**

1.32 br::pucrio::telemidia::ginga::ncl::multidevice Namespace Reference

Classes

- class **FormatterActiveDevice**
- class **FormatterBaseDevice**
- class **FormatterMultiDevice**
- class **FormatterPassiveDevice**
- class **IFormatterMultiDevice**

Variables

- static IComponentManager * **cm** = IComponentManager::getCMInstance()

1.33 br::pucrio::telemidia::ginga::ncl::prefetch Namespace Reference

Classes

- class **IPrefetchManager**
- class **PrefetchManager**

1.34 br::ufscar Namespace Reference

Namespaces

- namespace **lince**

1.35 br::ufscar::lince Namespace Reference

Namespaces

- namespace **ginga**

1.36 br::ufscar::lince::ginga Namespace Reference

Namespaces

- namespace **recommender**

1.37 br::ufscar::lince::ginga::recommender Namespace Reference

Classes

- class **SystemResource**
- class **IMiningAlgorithmApriori**
- class **MiningAlgorithm**
- class **si**
- class **User**
- class **sdt**
- class **eit**
- class **Channel**
- class **Context**
- class **Device**
- class **Document**
- class **Iteraction**
- class **IteractionBase**
- class **Key**
- class **LocalAgent**
- class **Media**
- class **Meta**
- class **NclStateMachine**
- class **Program**
- class **Thread**
- class **Volume**
- class **Scheduler**
- class **DataMining**
- class **Database**
- class **SchedulerItem**
- class **AgentListener**
- class **IAgentListener**

1.38 std Namespace Reference

STL namespace.

Classes

- class **allocator**
  STL class.
- class **auto_ptr**
  STL class.
- class **ios_base**
  STL class.
- class **basic_ios**
  STL class.
- class **basic_istream**
  STL class.
- class **basic_ostream**
  STL class.
- class **basic_iostream**
  STL class.
- class **basic_ifstream**
  STL class.
- class **basic_ofstream**
  STL class.
- class **basic_fstream**
  STL class.
- class **basic_istringstream**
  STL class.
- class **basic_ostringstream**
  STL class.
- class **basic_stringstream**
  STL class.
- class **ios**
  STL class.
- class **wios**

STL class.

- class **istream**
  STL class.
- class **wistream**
  STL class.
- class **ostream**
  STL class.
- class **wostream**
  STL class.
- class **ifstream**
  STL class.
- class **wifstream**
  STL class.
- class **ofstream**
  STL class.
- class **wofstream**
  STL class.
- class **fstream**
  STL class.
- class **wfstream**
  STL class.
- class **istringstream**
  STL class.
- class **wistringstream**
  STL class.
- class **ostringstream**
  STL class.
- class **wostringstream**
  STL class.
- class **stringstream**
  STL class.
- class **wstringstream**
  STL class.

- class **basic_string**
  STL class.
- class **string**
  STL class.
- class **wstring**
  STL class.
- class **complex**
  STL class.
- class **bitset**
  STL class.
- class **deque**
  STL class.
- class **list**
  STL class.
- class **map**
  STL class.
- class **multimap**
  STL class.
- class **set**
  STL class.
- class **multiset**
  STL class.
- class **vector**
  STL class.
- class **queue**
  STL class.
- class **priority_queue**
  STL class.
- class **stack**
  STL class.
- class **valarray**
  STL class.
- class **exception**
  STL class.

- class **bad_alloc**
  STL class.
- class **bad_cast**
  STL class.
- class **bad_typeid**
  STL class.
- class **logic_error**
  STL class.
- class **runtime_error**
  STL class.
- class **bad_exception**
  STL class.
- class **domain_error**
  STL class.
- class **invalid_argument**
  STL class.
- class **length_error**
  STL class.
- class **out_of_range**
  STL class.
- class **range_error**
  STL class.
- class **overflow_error**
  STL class.
- class **underflow_error**
  STL class.

1.38.1 Detailed Description

STL namespace.

Chapter 2

ModuloRecommender Class Documentation

2.1 Apriori Class Reference

`#include <Apriori.hpp>`

Collaboration diagram for Apriori:

Public Member Functions

- **Apriori** (ifstream &basket_file, const char *output_file_name, const bool **store_input**)
- void **APRIORI_alg** (const double min_supp, const double min_conf, const bool quiet, const unsigned long size_threshold)

  *Esse método implementa o algoritmo **Apriori*** (p. **??**).

Private Member Functions

- void **support** (const itemtype &candidate_size)

Private Attributes

- **Apriori_Trie** * **apriori_trie**

  Abstração de uma árvore recursiva que armazena os item mais frequentes e portanto candidatos.

- **Input_Output_Manager input_output_manager**

  Input_output_manager é responsável por gerenciar as operações de leitura e escrita.

- map< vector< itemtype >, unsigned long > **reduced_baskets**

  Armazenar as informações do banco de dados em memória, se o parâmetro store_input=true;.

- bool **store_input**

  se store_input = true, então as informações provinientes do arquivo de entrada serão armazenadas na memória

2.1.1 Detailed Description

Essa classe implementa o algoritmo APRIORI.

O objetivo do algoritmo apriori é encontrar os conjuntos de dados frequentes L no banco de dados. Itens com pouca frequencia devem ser deletados permanecendo apenas aqueles que ocorrem mais vezes.

Definition at line 81 of file Apriori.hpp.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 Apriori (ifstream & *basket_file*, const char * *output_file_name*, const bool *store_input*)

Construtor.

Parameters:

basket_file arquivo com o conjunto de dados para mineração

output_file_name arquivo de saída com as regras de associação

store_input **true** indica para ler todo arquivo e armazenar as informações em memória

The documentation for this class was generated from the following file:

- Recommender/MiningAlgorithm/include/apriori23/Apriori.hpp

2.2 Apriori_Trie Class Reference

`#include <Apriori_Trie.hpp>`

Collaboration diagram for Apriori_Trie:

Public Member Functions

- **Apriori_Trie** (const unsigned long counter_of_emptyset)
- void **insert_frequent_items** (const vector< unsigned long > &counters)

  Insere os itens mais frequentes.

- void **candidate_generation** (const itemtype &frequent_size)

  Gera os canditados a regras.

- void **find_candidate** (const vector< itemtype > &basket, const itemtype candidate_size, const unsigned long counter=1)

  Incrementa o contador dos canditados.

- void **delete_infrequent** (const double min_occurrence, const itemtype candidate_size)

  Deleta os itemsets menos frequentes.

- void **association** (const double min_conf, **Input_Output_Manager** &input_output_manager) const

  Gera as regras de associação.

- itemtype **longest_path** () const

  Retorna o tamanho do maior caminho até o nó folha da arvore (trie.hpp).

- void **write_content_to_file** (**Input_Output_Manager** &input_output_manager) const

  Escreve os itens mais frequentes no arquivo.

- void **show_content_preorder** () const

  apresenta o conteúdo da árvore gerada em pré-ordem

Protected Member Functions

- bool **is_all_subset_frequent** (const set< itemtype > &maybe_candidate) const

  Decide se todos os nós na arvore são frequentes.

- void **candidate_generation_two** ()

  Gera os candidatos com tamanho 2. Ex: Quem assiste A assiste B.

- void **candidate_generation_assist** (**Trie** *__Trie__, const itemtype distance_from_generator, set< itemtype > &maybe_candidate)

  Gera os candidatos com 2 ou mais. Ex: Quem assiste A assiste B,C,D.

- void **find_candidate_two** (const vector< itemtype > &basket, const unsigned long counter=1)

  Incrementa o contador dos candidatos encontrados.

- void **delete_infrequent_two** (const double min_occurrence)

  Deleta os itens menos frequentes de tamanho 2.

- void **assoc_rule_find** (const double min_conf, set< itemtype > &condition_part, set< itemtype > &consequence_part, const unsigned long union_support, **Input_Output_Manager** &input_output_manager) const

  Encontra regras de associação.

- void **assoc_rule_assist** (const double min_conf, const **Trie** *__Trie__, set< itemtype > &consequence_part, **Input_Output_Manager** &input_output_manager) const

  Geração parcial das regras de associação.

- void **write_content_to_file_assist** (**Input_Output_Manager** &input_output_manager, const **Trie** *actual_state, const itemtype distance_from_frequent, set< itemtype > &frequent_itemset) const

  Escreve as regras de associação encontradas no disco.

Protected Attributes

- **Trie main_trie**

  Trie (p. **??**) *armazena os candidatos e os itens mais frequentes.*

- vector< vector< unsigned long > > **temp_counter_array**

2.2.1 Detailed Description

Apriori_Trie (p. **??**)

Apriori_Trie (p. **??**) cria a estrutura de uma árvore concebida para fornecer métodos eficientes para o algoritmo apriori.

Definition at line 85 of file Apriori_Trie.hpp.

2.2.2 Member Data Documentation

2.2.2.1 vector< vector<unsigned long> > temp_counter_array [protected]

temp_counter_array stores armazena as ocorrência de pares

Definition at line 172 of file Apriori_Trie.hpp.

The documentation for this class was generated from the following file:

- Recommender/MiningAlgorithm/include/apriori23/Apriori_Trie.hpp

2.3 BITSTREAM Struct Reference

Collaboration diagram for BITSTREAM:

Public Attributes

- FILE ∗ **File**
- int **BitsInQueue**
- int **BitQueue**

2.3.1 Detailed Description

Definition at line 18 of file file.h.

The documentation for this struct was generated from the following file:

- Recommender/MiningAlgorithm/include/ikmeans/file.h

2.4 BOOKNODE Struct Reference

Collaboration diagram for BOOKNODE:

Public Attributes

- VECTORTYPE **vector**
- VECTORELEMENT **vmean**
- int **freq**
- char * **name**

2.4.1 Detailed Description

Definition at line 28 of file cb.h.

The documentation for this struct was generated from the following file:

- Recommender/MiningAlgorithm/include/ikmeans/cb.h

2.5 ContextManager Class Reference

Inheritance diagram for ContextManager:Collaboration diagram for ContextManager:

Public Member Functions

- void **addContextVar** (int userId, string varName, string varValue)
- void **addUser** (**IGingaUser** ∗newUser)
- void **saveUsersAccounts** ()
- void **saveUsersProfiles** ()
- void **setCurrentUserId** (int userId)
- int **getCurrentUserId** ()
- **IGingaUser** ∗ **getUser** (int userId)
- map< string, string > ∗ **getUserProfile** (int userId)
- map< string, string > ∗ **getUsersNames** ()
- **ISystemInfo** ∗ **getSystemInfo** ()
- vector< int > ∗ **getUserIds** ()
- int **addUser** (string name, string passwd, int age, string location, char gender)
- bool **removeUser** (int userid)

Static Public Member Functions

- static **IContextManager** ∗ **getInstance** ()

Private Member Functions

- void **initializeUsers** ()
- void **initializeContexts** ()
- void **saveProfile** (int fd, int userId, map< string, string > ∗profile)
- void **listUsersNicks** ()

Private Attributes

- map< int, **IGingaUser** ∗ > ∗ **users**
- map< int, map< string, string > ∗ > ∗ **contexts**
- string **usersUri**
- string **contextsUri**
- int **curUserId**
- **ISystemInfo** ∗ **systemInfo**

Static Private Attributes

- static **IContextManager** ∗ **_instance**

2.5.1 Detailed Description

Definition at line 62 of file ContextManager.h.

2.5.2 Member Function Documentation

2.5.2.1 vector<int>* getUserIds () [virtual]

Give a list of users ids

Returns:

vector<int>* pointer for a feature vector which corresponds to the user ids

Implements **IContextManager** (p. **??**).

2.5.2.2 int addUser (string *name*, string *passwd*, int *age*, string *location*, char *gender*) [virtual]

Add user to context.

Parameters:

name string user name

passwd string user passwd

age int user age

location string user location

gender char user gender

Returns:

int the next ID of user

Implements **IContextManager** (p. **??**).

2.5.2.3 bool removeUser (int *userid*) [virtual]

Remove user from context

Parameters:

userid int user id

Returns:

bool the user is successfully removed

Implements **IContextManager** (p. **??**).

2.5.2.4 void listUsersNicks () [private]

List users names

The documentation for this class was generated from the following file:

- gingacc-cpp/gingacc-contextmanager/include/ContextManager.h

2.6 DataInterface Class Reference

`#include <DataInterface.h>`

Collaboration diagram for DataInterface:

Public Member Functions

- **DataInterface** ()
- ∼**DataInterface** ()
- **TVData** ∗ **getData** ()

Private Attributes

- vector< **TVData** > ∗ **DataItens**
- time_t **starttime**
- int **startpoint**

2.6.1 Detailed Description

Class **DataInterface** (p. **??**) This class provides simulates the broadcast of TV meta-data to be used to infer implicit profiles. In particular, this class simulates the broadcast of some data from SI tables. The actual class should provide access to the SI data through the TS decoding services provided by Ginga or by the Recommender API Note that it also could provided data related to user interactions with the set-top-box, gathered by the recommender

Author:

Marco Cristo & Angelo Bittar

Definition at line 173 of file DataInterface.h.

2.6.2 Constructor & Destructor Documentation

2.6.2.1 DataInterface ()

Load data schedule from a file where each line has the format: data1,data2,data3,... Each line represents a service. data1 is the offset time (in seconds) for the start of the service. Eg: 7,globo,corujao,filme,filme. Globo service "corujao" (genre: filme, subgenre: filme) will start 7 seconds after the beginning of the simulation.

2.6.2.2 ∼DataInterface ()

Destroys data schedule from memory

2.6.3 Member Function Documentation

2.6.3.1 TVData* getData ()

Returns object with data of interest, according with request time (current time) and data schedule (loaded previously)

Returns:

TVData* pointer to object with data of interest

The documentation for this class was generated from the following file:

- gingacc-cpp/gingacc-contextmanager/include/DataInterface.h

2.7 GingaUser Class Reference

Inheritance diagram for GingaUser:Collaboration diagram for GingaUser:

Public Member Functions

- **GingaUser** (int **userId**, string **userName**, string passwd)
- int **getUserId** ()
- string **getUserName** ()
- int **getUserAge** ()
- string **getUserLocation** ()
- char **getUserGenre** ()
- bool **isValidPassword** (string passwd)
- bool **setPassword** (string oldPasswd, string newPasswd)
- void **setUserName** (string passwd, string **userName**)
- void **setUserAge** (string passwd, int **userAge**)
- void **setUserLocation** (string passwd, string **userLocation**)
- void **setUserGenre** (string passwd, char **userGenre**)
- void **saveTo** (int fd)

Static Public Member Functions

- static void **saveString** (int fd, string bytesToSave)

Private Attributes

- int **userId**
- string **userName**
- string **userPasswd**
- int **userAge**
- string **userLocation**
- char **userGenre**

2.7.1 Detailed Description

Definition at line 64 of file GingaUser.h.

The documentation for this class was generated from the following file:

- gingacc-cpp/gingacc-contextmanager/include/user/GingaUser.h

2.8 IContextManager Class Reference

Inheritance diagram for IContextManager:

Public Member Functions

- virtual ~**IContextManager** ()
- virtual void **addContextVar** (int userId, string varName, string varValue)=0
- virtual void **addUser** (**IGingaUser** *newUser)=0
- virtual void **saveUsersAccounts** ()=0
- virtual void **saveUsersProfiles** ()=0
- virtual void **setCurrentUserId** (int userId)=0
- virtual int **getCurrentUserId** ()=0
- virtual **IGingaUser** * **getUser** (int userId)=0
- virtual map< string, string > * **getUserProfile** (int userId)=0
- virtual map< string, string > * **getUsersNames** ()=0
- virtual **ISystemInfo** * **getSystemInfo** ()=0
- virtual vector< int > * **getUserIds** ()=0
- virtual int **addUser** (string name, string passwd, int age, string location, char gender)=0
- virtual bool **removeUser** (int userid)=0

2.8.1 Detailed Description

Definition at line 66 of file IContextManager.h.

The documentation for this class was generated from the following file:

- gingacc-cpp/gingacc-contextmanager/include/IContextManager.h

2.9 IGingaUser Class Reference

Inheritance diagram for IGingaUser:

Public Member Functions

- virtual ~**IGingaUser** ()
- virtual int **getUserId** ()=0
- virtual string **getUserName** ()=0
- virtual int **getUserAge** ()=0
- virtual char **getUserGenre** ()=0
- virtual string **getUserLocation** ()=0
- virtual bool **isValidPassword** (string passwd)=0
- virtual bool **setPassword** (string oldPasswd, string newPasswd)=0
- virtual void **setUserName** (string passwd, string userName)=0
- virtual void **setUserAge** (string passwd, int userAge)=0
- virtual void **setUserLocation** (string passwd, string userLocation)=0
- virtual void **setUserGenre** (string passwd, char userGenre)=0
- virtual void **saveTo** (int fd)=0

2.9.1 Detailed Description

Definition at line 62 of file IGingaUser.h.

The documentation for this class was generated from the following file:

- gingacc-cpp/gingacc-contextmanager/include/user/IGingaUser.h

2.10 ImplicitContextManager Class Reference

`#include <ImplicitContextManager.h>`

Collaboration diagram for ImplicitContextManager:

Public Member Functions

- **ImplicitContextManager** ()
- ∼**ImplicitContextManager** ()
- int **getCurrentUserId** ()
- int **getCurrentImplicitUserId** ()
- int **getCurrentProfileId** (int userid)
- bool **setCurrentUserId** (int userid)
- bool **setCurrentImplicitUserId** (int userid)
- bool **setCurrentProfileId** (int userid, int profid)
- vector< int > ∗ **getProfile** (int userid, int profid)
- void **createProfile** (int uid)
- void **guessImplicitUser** ()
- bool **remove** (int userid)
- void **checkRecommenderModule** ()
- pthread_t **getThread** ()
- bool **should_stop** (void)
- void **stop** (void)

Static Public Member Functions

- static bool **canInstantiate** ()
- static **ImplicitContextManager** ∗ **getInstance** ()

Private Member Functions

- float **cosine** (vector< float > ∗v1, vector< float > ∗v2)
- void **gatherStats** (**TVData** ∗si)
- void **compareProfiles** (int &bestc, pair< int, int > &bestp, float &sim)
- void **updateBestProfile** (int bestc, pair< int, int > bestp)
- int **selectNewProfile** (void)
- string **showv** (vector< float > ∗v)
- string **showv** (vector< int > ∗v)

Private Attributes

- **PresentationContext** ∗ **pctx**
- **DataInterface** ∗ **sii**
- **ImplicitProfiles** ∗ **iprofiles**
- int **curImplicitUser**
- string **prevData**
- vector< pair< vector< int >, int > > **candidates**
- string **prevSubGenre**

- int **prevTimeStamp**
- int **prevUserId**
- bool **newData**
- pthread_t **recomenda**
- volatile bool **stopflag**

Static Private Attributes

- static **ImplicitContextManager** * **_instance**
- static const float **MIN_ACCEPTABLE_SIMILARITY** = 0.3
- static const float **HIGH_LEVEL_OF_CERTAINTY** = 0.95
- static const int **DEFAULT_USER_ID** = 0

2.10.1 Detailed Description

Wrapper for context manager that provides an interface for implicit users

Author:

Marco Cristo & Angelo Bittar

Definition at line 109 of file ImplicitContextManager.h.

2.10.2 Constructor & Destructor Documentation

2.10.2.1 ImplicitContextManager ()

Get instance of (explicit) presentation context and corresponding context manager. Load users and profiles Set current implicit user profile. Clean up profile base Starts class **DataInterface** (p. **??**) which simulates TV metadata broadcast (test purpose only – actual **DataInterface** (p. **??**) should only provide access to TV metadata).

2.10.2.2 ~ImplicitContextManager ()

Destructor for implicit context manager

2.10.3 Member Function Documentation

2.10.3.1 float cosine (vector< float > * *v1*, vector< float > * *v2*) [private]

Returns value of cosine between two vectors

Parameters:

v1 first vector.

v2 second vector.

Returns:

float value of cossine.

2.10.3.2 void gatherStats (TVData * *si*) [private]

Gather stats.

Parameters:

si pointer to TV metadata.

2.10.3.3 void compareProfiles (int & *bestc*, pair< int, int > & *bestp*, float & *sim*) [private]

Compare base profiles with candidates.

Parameters:

bestc reference of best candidate.

bestp reference of best user and implicituser.

sim similarity value for best candidate.

2.10.3.4 void updateBestProfile (int *bestc*, pair< int, int > *bestp*) [private]

Update best profile.

Parameters:

bestc the best candidate.

bestp the best implicitUser.

2.10.3.5 int selectNewProfile (void) [private]

Select new profile.

Returns:

int the selected profile.

2.10.3.6 static bool canInstantiate () [static]

Check if PresentationContext is already instantiated

Returns:

static bool value indicating if context instance has already been created.

2.10.3.7 static ImplicitContextManager* getInstance () [static]

Returns instance of implicit context manager

Returns:

static ImplicitContextManager* pointer to object implicit context manager

2.10.3.8 int getCurrentUserId ()

Returns current user id

Returns:

int current user id

2.10.3.9 int getCurrentImplicitUserId ()

Returns current implicit user id

Returns:

int current implicit user id

2.10.3.10 int getCurrentProfileId (int *userid*)

Returns current profile id for giver user id

Parameters:

userid user id.

Returns:

int current profile id

2.10.3.11 bool setCurrentUserId (int *userid*)

Returns true if current user id was set

Parameters:

userid user id.

Returns:

bool value indicating if opeation was performed

2.10.3.12 bool setCurrentImplicitUserId (int *userid*)

Returns true if current implicit user id was set

Parameters:

userid user id.

Returns:

bool value indicating if opeation was performed

2.10.3.13 bool setCurrentProfileId (int *userid*, int *profid*)

Returns true if current profile id was set

Parameters:

userid user id.

profileid profile id.

Returns:

bool value indicating if opeation was performed

2.10.3.14 vector<int>* getProfile (int *userid*, int *profid*)

Returns a profile vector of a user

Parameters:

userid user id

profid profile id

Returns:

vector<int>* pointer for a feature vector which corresponds to the user profile

2.10.3.15 void createProfile (int *uid*)

Create a new implicit profile

Parameters:

uid user id

2.10.3.16 void guessImplicitUser ()

Guess implicit user based on its profiles and additional evidence

2.10.3.17 bool remove (int *userid*)

Returns true if profile profid of user userid was deleted

Parameters:

userid user id

Returns:

bool value indicating if opeation was performed

2.10.3.18 void checkRecommenderModule ()

Create a thread named recomenda

2.10.3.19 pthread_t getThread ()

Get the thread that was create in chekRecommenderModule pthread_t thread recommenda

2.10.3.20 bool should_stop (void) [inline]

Flag used to stop the thread bool false if you want stop the thread

Definition at line 359 of file ImplicitContextManager.h.

References ImplicitContextManager::stopflag.

2.10.3.21 void stop (void) [inline]

Change the status of stopflag to true

Definition at line 365 of file ImplicitContextManager.h.

References ImplicitContextManager::stopflag.

2.10.4 Member Data Documentation

2.10.4.1 PresentationContext* pctx [private]

! Pointer to presentation context.

Definition at line 115 of file ImplicitContextManager.h.

2.10.4.2 DataInterface* sii [private]

! Pointer to access interface to TV metadata

Definition at line 118 of file ImplicitContextManager.h.

2.10.4.3 ImplicitProfiles* iprofiles [private]

! Pointer to implicit user profiles

Definition at line 121 of file ImplicitContextManager.h.

2.10.4.4 int curImplicitUser [private]

! Current user id

Definition at line 124 of file ImplicitContextManager.h.

2.10.4.5 string prevData [private]

! Previous data observed by **guessImplicitUser()** (p. **??**). Corresponds to service name information.

Definition at line 127 of file ImplicitContextManager.h.

2.10.4.6 vector<pair<vector<int> , int> > candidates [private]

! List of profile candidates

Definition at line 131 of file ImplicitContextManager.h.

2.10.4.7 string prevSubGenre [private]

! Previous subgenre observed by **guessImplicitUser()** (p. **??**).

Definition at line 134 of file ImplicitContextManager.h.

2.10.4.8 int prevTimeStamp [private]

! Previous timestamp observed by **guessImplicitUser()** (p. **??**).

Definition at line 137 of file ImplicitContextManager.h.

2.10.4.9 int prevUserId [private]

! Previous user id observed by **guessImplicitUser()** (p. **??**).

Definition at line 140 of file ImplicitContextManager.h.

2.10.4.10 ImplicitContextManager* _instance [static, private]

! Address of **ImplicitContextManager** (p. **??**) instance.

Definition at line 143 of file ImplicitContextManager.h.

2.10.4.11 const float MIN_ACCEPTABLE_SIMILARITY = 0.3 [static, private]

! Minimum probability to profile being considered similar to another profile.

Definition at line 146 of file ImplicitContextManager.h.

2.10.4.12 const float HIGH_LEVEL_OF_CERTAINTY = 0.95 [static, private]

! Minimum level of certainty to profile being considered new.

Definition at line 149 of file ImplicitContextManager.h.

2.10.4.13 const int DEFAULT_USER_ID = 0 [static, private]

! User id of default Ginga user

Definition at line 152 of file ImplicitContextManager.h.

2.10.4.14 bool newData [private]

! Control when information changes - Debug purpose.

Definition at line 207 of file ImplicitContextManager.h.

2.10.4.15 pthread_t recomenda [private]

! Thread responsible for the calculation of User

Definition at line 210 of file ImplicitContextManager.h.

2.10.4.16 volatile bool stopflag [private]

! Flag responsible to stop de thread function

Definition at line 213 of file ImplicitContextManager.h.

Referenced by ImplicitContextManager::should_stop(), and ImplicitContextManager::stop().

The documentation for this class was generated from the following file:

- gingacc-cpp/gingacc-contextmanager/include/ImplicitContextManager.h

2.11 ImplicitProfiles Class Reference

`#include <ImplicitProfiles.h>`

Collaboration diagram for ImplicitProfiles:

Public Member Functions

- **ImplicitProfiles** (**PresentationContext** ∗**pctx**)
- ∼**ImplicitProfiles** ()
- void **createProfile** (int uid)
- int **getCurrentId** (int userid)
- bool **setCurrentImplicitUserId** (int userid)
- bool **setCurrentId** (int userid, int profid)
- **tIProfiles** ∗ **get** ()
- vector< int > ∗ **get** (int userid, int profid)
- int **findLastUpdateInDays** (int userid, int profid)
- int **getLastUpdateInDays** (int t)
- bool **insert** (int userid, vector< int > ∗v, int &profid)
- bool **remove** (int userid, int profid)
- bool **modify** (int userid, int profid)
- void **show** ()
- int **getAxis** (string s)
- int **getSize** (void)

Private Member Functions

- void **deleteOld** ()
- void **cluster** ()
- void **setValue** (int userid, string property, string value)
- void **delVar** (int userid, string property)

Private Attributes

- **PresentationContext** ∗ **pctx**
- **tIProfiles** ∗ **iprofiles**
- map< int, int > ∗ **nextpid**
- map< int, int > ∗ **curProfile**
- int **profileSize**
- map< string, int > **s2i**

Static Private Attributes

- static const int **MAX_PROFILE_UNUSED_TIME** = 365

2.11.1 Detailed Description

Class **ImplicitProfiles** (p. **??**) Implicit user profiles

Author:

Marco Cristo & Angelo Bittar

Definition at line 117 of file ImplicitProfiles.h.

2.11.2 Constructor & Destructor Documentation

2.11.2.1 ImplicitProfiles (PresentationContext * *pctx*)

Get instance of (explicit) presentation context and corresponding context manager. Load users and profiles and create new implicit profiles for users that do not have one. Set current profile for each user and current implicit user profile. Clean up profile base by eliminating old profiles and clustering similar profiles.

Parameters:

pcxt pointer to Presentation Context

2.11.2.2 ~ImplicitProfiles ()

Destructor for implicit profiles

2.11.3 Member Function Documentation

2.11.3.1 void deleteOld () [private]

Delete profiles that have not been updated for a long time

2.11.3.2 void cluster () [private]

Merge very similar pofiles, if number of profiles is greater than a certain threshold

2.11.3.3 void setValue (int *userid*, string *property*, string *value*) [private]

Set the value of an implicit profile property for a given user id, that is, property[userid] = value.

Parameters:

userid integer user id

property the string property to be set

value the string value to be instantiated

2.11.3.4 void delVar (int *userid*, string *property*) [private]

Delete an implicit profile property for a given user id.

Parameters:

userid the user id
property the string property to be deleted

2.11.3.5 void createProfile (int *uid*)

Create a new implicit profile

Parameters:

uid user id

2.11.3.6 int getCurrentId (int *userid*)

Returns current profile id for giver user id

Parameters:

userid user id

Returns:

int current profile id

2.11.3.7 bool setCurrentImplicitUserId (int *userid*)

Returns true if current implicit user id was set

Parameters:

userid user id

Returns:

bool value indicating if opeation was performed

2.11.3.8 bool setCurrentId (int *userid*, int *profid*)

Returns true if current profile id was set

Parameters:

userid user id
profid profile id

Returns:

bool value indicating if opeation was performed

2.11.3.9 tIProfiles* get ()

Returns map profiles which stores the profiles of all users

Returns:

map<int, map<int, pair<time_t, vector<int>*> >*>* pointer for all profiles

2.11.3.10 vector<int>* get (int *userid*, int *profid*)

Returns a profile vector of a user

Parameters:

userid user id

profid profile id

Returns:

vector<int>* pointer for a feature vector which corresponds to the user profile

2.11.3.11 int findLastUpdateInDays (int *userid*, int *profid*)

Find profile age in days

Parameters:

userid user id

profid profile id

Returns:

int profile age (note: if profile has 4.7 days, function will return 4 days)

2.11.3.12 int getLastUpdateInDays (int *t*)

Calculate profile age in days given a certain unix timestamp t

Parameters:

t timestamp

Returns:

int profile age (note: if profile has 4.7 days, function will return 4 days)

2.11.3.13 bool insert (int *userid*, vector< int > * *v*, int & *profid*)

Returns true if profile profid of user userid given by vector v was inserted

Parameters:

userid user id

∗***v*** pointer to profile vector

profid profile id given for inserted profile

Returns:

bool value indicating if opeation was performed

2.11.3.14 bool remove (int *userid*, int *profid*)

Returns true if profile profid of user userid was deleted

Parameters:

userid user id

profid profile id of deleted profile

Returns:

bool value indicating if opeation was performed

2.11.3.15 bool modify (int *userid*, int *profid*)

Returns true if profile profid of user userid was modified

Parameters:

userid user id

profid profile id of modified profile

Returns:

bool value indicating if opeation was performed

2.11.3.16 void show ()

Show implicit profiles – debug assistance.

2.11.3.17 int getAxis (string *s*)

Indicating the position of subgenre in the vector Get axis value associated with input data.

Parameters:

s name of the subGenre.

Returns:

int value indicating the position of subgenre in the vector.

2.11.3.18 int getSize (void) [inline]

Return the size that profiles have. Get profile size.

Returns:

int the size of profiles.

Definition at line 293 of file ImplicitProfiles.h.

References ImplicitProfiles::profileSize.

2.11.4 Member Data Documentation

2.11.4.1 PresentationContext* pctx [private]

! Pointer to Ginga presentation context. Allows for access to live context variables.

Definition at line 122 of file ImplicitProfiles.h.

2.11.4.2 tIProfiles* iprofiles [private]

! Implicit profiles.

Definition at line 125 of file ImplicitProfiles.h.

2.11.4.3 map<int, int>* nextpid [private]

! Next pid to be used for each user.

Definition at line 128 of file ImplicitProfiles.h.

2.11.4.4 map<int, int>* curProfile [private]

! Current profile for each user

Definition at line 131 of file ImplicitProfiles.h.

2.11.4.5 int profileSize [private]

! Size (number of features) of the profile vector.

Definition at line 134 of file ImplicitProfiles.h.

Referenced by ImplicitProfiles::getSize().

2.11.4.6 const int MAX_PROFILE_UNUSED_TIME = 365 [static, private]

! Maximum time the profile stays in database without being used.

Definition at line 138 of file ImplicitProfiles.h.

2.11.4.7 map<string, int> s2i [private]

! Map data to axis in internal vector representation

Definition at line 141 of file ImplicitProfiles.h.

The documentation for this class was generated from the following file:

- gingacc-cpp/gingacc-contextmanager/include/ImplicitProfiles.h

2.12 ISystemInfo Class Reference

Inheritance diagram for ISystemInfo:

Public Member Functions

- virtual ~**ISystemInfo** ()
- virtual void **setSystemTable** (map< string, string > *sysTable)=0
- virtual string **getSystemLanguage** ()=0
- virtual string **getCaptionLanguage** ()=0
- virtual string **getSubtitleLanguage** ()=0
- virtual float **getReturnBitRate** ()=0
- virtual void **getScreenSize** (int *width, int *height)=0
- virtual void **getScreenGraphicSize** (int *width, int *height)=0
- virtual string **getAudioType** ()=0
- virtual float **getCPUClock** ()=0
- virtual float **getMemorySize** ()=0
- virtual string **getOperatingSystem** ()=0
- virtual string **getJavaConfiguration** ()=0
- virtual string **getJavaProfile** ()=0
- virtual string **getLuaVersion** ()=0

2.12.1 Detailed Description

Definition at line 63 of file ISystemInfo.h.

The documentation for this class was generated from the following file:

- gingacc-cpp/gingacc-contextmanager/include/system/ISystemInfo.h

2.13 SystemInfo Class Reference

Inheritance diagram for SystemInfo:Collaboration diagram for SystemInfo:

Public Member Functions

- void **setSystemTable** (map< string, string > ∗**sysTable**)
- string **getSystemLanguage** ()
- string **getCaptionLanguage** ()
- string **getSubtitleLanguage** ()
- float **getReturnBitRate** ()
- void **getScreenSize** (int ∗width, int ∗height)
- void **getScreenGraphicSize** (int ∗width, int ∗height)
- string **getAudioType** ()
- float **getCPUClock** ()
- float **getMemorySize** ()
- string **getOperatingSystem** ()
- string **getJavaConfiguration** ()
- string **getJavaProfile** ()
- string **getLuaVersion** ()

Private Member Functions

- void **initializeClockSpeed** ()
- string **getValue** (string attribute)
- void **printSysTable** ()

Private Attributes

- utsname **sn**
- sysinfo **info**
- float **clockSpeed**
- map< string, string > ∗ **sysTable**

2.13.1 Detailed Description

Definition at line 67 of file SystemInfo.h.

The documentation for this class was generated from the following file:

- gingacc-cpp/gingacc-contextmanager/include/system/SystemInfo.h

2.14 TVData Class Reference

`#include <DataInterface.h>`

Collaboration diagram for TVData:

Public Member Functions

- **TVData** (int sec, string pn, string sn, string g, string sg)
- int **getTimeStamp** ()
- string **getSProviderName** ()
- string **getServiceName** ()
- string **getGenre** ()
- string **getSubGenre** ()

Private Attributes

- int **seconds**
- string **providerName**
- string **serviceName**
- string **genre**
- string **subgenre**

2.14.1 Detailed Description

This class provides the data to be used in simulation of SI data broadcast. The data to be "broadcasted" comprise the service provider name, service name (from SDT table), genre, and subgenre (from EIT table)

Author:

Marco Cristo & Angelo Bittar

Definition at line 87 of file DataInterface.h.

2.14.2 Constructor & Destructor Documentation

2.14.2.1 TVData (int *sec*, string *pn*, string *sn*, string *g*, string *sg*) [inline]

Starts **TVData** (p. **??**) object state

Definition at line 109 of file DataInterface.h.

References TVData::genre, TVData::providerName, TVData::seconds, TVData::serviceName, and TVData::subgenre.

2.14.3 Member Function Documentation

2.14.3.1 int getTimeStamp () [inline]

Returns program start time (in seconds, counting from the start of the simulation)

Returns:

int program start time (in seconds, counting from the start of the simulation)

Definition at line 122 of file DataInterface.h.

References TVData::seconds.

2.14.3.2 string getSProviderName () [inline]

Returns provider name

Returns:

string provider name

Definition at line 131 of file DataInterface.h.

References TVData::providerName.

2.14.3.3 string getServiceName () [inline]

Returns service name

Returns:

string service name

Definition at line 140 of file DataInterface.h.

References TVData::serviceName.

2.14.3.4 string getGenre () [inline]

Returns service genre

Returns:

string service genre

Definition at line 149 of file DataInterface.h.

References TVData::genre.

2.14.3.5 string getSubGenre () [inline]

Returns service subgenre

Returns:

string service subgenre

Definition at line 158 of file DataInterface.h.

References TVData::subgenre.

2.14.4 Member Data Documentation

2.14.4.1 int seconds [private]

! Time in seconds since the begining of the simulation.

Definition at line 91 of file DataInterface.h.

Referenced by TVData::getTimeStamp(), and TVData::TVData().

2.14.4.2 string providerName [private]

! Name of the service proveder.

Definition at line 94 of file DataInterface.h.

Referenced by TVData::getSProviderName(), and TVData::TVData().

2.14.4.3 string serviceName [private]

! Name of the service.

Definition at line 97 of file DataInterface.h.

Referenced by TVData::getServiceName(), and TVData::TVData().

2.14.4.4 string genre [private]

! Service genre.

Definition at line 100 of file DataInterface.h.

Referenced by TVData::getGenre(), and TVData::TVData().

2.14.4.5 string subgenre [private]

! Service sub-genre.

Definition at line 103 of file DataInterface.h.

Referenced by TVData::getSubGenre(), and TVData::TVData().

The documentation for this class was generated from the following file:

- gingacc-cpp/gingacc-contextmanager/include/DataInterface.h

2.15 AuthenticationServer Class Reference

Collaboration diagram for AuthenticationServer:

Public Member Functions

- **AuthenticationServer** (**PresentationContext** ∗**pctx**)
- void **start** ()

Private Member Functions

- void **tokenize** (const std::string &str, std::vector< std::string > &tokens, const std::string &delimiters=" ")
- IGingaUser ∗ **getGingaUser** (string login, string passw)
- bool **setUserProperty** (IGingaUser ∗u, string passwd, string p, string v)
- bool **getUserProperty** (IGingaUser ∗u, string passwd, string p, string &v)
- void **processClient** (int new_socket)
- const char ∗ **addUser** (int csocket, string passwd, string name, string age, string location, string gender)
- const char ∗ **removeUser** (int csocket, string userid, string passwd)
- const char ∗ **updateUser** (int csocket, string userid, string passwd, string property, string value)
- const char ∗ **queryUser** (int csocket, string userid, string passwd, string property)
- const char ∗ **login** (int csocket, string userid, string passwd)
- const char ∗ **logout** (int csocket, string userid, string passwd)
- const char ∗ **getCurrentUser** (int csocket)
- const char ∗ **listUsers** (int csocket)
- const char ∗ **showVar** (int csocket, string userid, string passwd, string property)
- const char ∗ **updateVar** (int csocket, string userid, string passwd, string property, string value)
- void **saveUser** (int uid)

Private Attributes

- set< string > **userfields**
- **PresentationContext** ∗ **pctx**
- IContextManager ∗ **ctxm**
- ImplicitContextManager ∗ **ictxm**
- LocalAgent ∗ **lagt**

Static Private Attributes

- static const int **BufferSize** = 1024
- static const int **ResultSize** = 6
- static const int **ASPort** = 8183

2.15.1 Detailed Description

Definition at line 88 of file AuthenticationServer.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adaptation/context/AuthenticationServer.h

2.16 ContextBase Class Reference

Inheritance diagram for ContextBase:Collaboration diagram for ContextBase:

Public Member Functions

- virtual ~**ContextBase** ()

Static Public Attributes

- static const string **SYSTEM_LANGUAGE**
- static const string **SYSTEM_CAPTION**
- static const string **SYSTEM_SUBTITLE**
- static const string **SYSTEM_RETURN_BIT_RATE**
- static const string **SYSTEM_SCREEN_SIZE**
- static const string **SYSTEM_SCREEN_GRAPHIC_SIZE**
- static const string **SYSTEM_AUDIO_TYPE**
- static const string **SYSTEM_CPU**
- static const string **SYSTEM_MEMORY**
- static const string **SYSTEM_OPERATING_SYSTEM**
- static const string **SYSTEM_JAVA_CONFIG**
- static const string **SYSTEM_JAVA_PROFILE**
- static const string **SYSTEM_LUA_VERSION**
- static const string **SYSTEM_DEVNUMBER**
- static const string **SYSTEM_CLASSTYPE**
- static const string **SYSTEM_INFO**
- static const string **SYSTEM_CLASS_NUMBER**
- static const string **USER_AGE**
- static const string **USER_LOCATION**
- static const string **USER_GENRE**
- static const string **DEFAULT_FOCUS_BORDER_COLOR**
- static const string **DEFAULT_FOCUS_BORDER_WIDTH**
- static const string **DEFAULT_FOCUS_BORDER_TRANSPARENCY**
- static const string **DEFAULT_SEL_BORDER_COLOR**

2.16.1 Detailed Description

Definition at line 63 of file ContextBase.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adaptation/context/ContextBase.h

2.17 PresentationContext Class Reference

Inheritance diagram for PresentationContext:Collaboration diagram for PresentationContext:

Public Member Functions

- void **setPropertyValue** (string propertyName, string value)
- void **incPropertyValue** (string propertyName)
- void **decPropertyValue** (string propertyName)
- vector< string > * **getPropertyNames** ()
- string **getPropertyValue** (string attributeId)
- IContextManager * **getContextManager** ()
- void **lock** ()
- void **unlock** ()
- void **initializeContext** ()
- bool **stopServer** (void)
- void **save** ()

Static Public Member Functions

- static bool **instantiated** ()
- static **PresentationContext** * **getInstance** ()

Private Member Functions

- void **initializeUserContext** ()
- void **initializeUserInfo** (int currentUserId)
- void **initializeSystemValues** ()

Private Attributes

- map< string, string > * **contextTable**
- IContextManager * **contextManager**
- pthread_mutex_t **attrMutex**
- pthread_t **aserver**
- volatile bool **stop**

Static Private Attributes

- static **PresentationContext** * **_instance**

2.17.1 Detailed Description

Definition at line 105 of file PresentationContext.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adaptation/context/PresentationContext.h

2.18 RuleAdapter Class Reference

Inheritance diagram for RuleAdapter:Collaboration diagram for RuleAdapter:

Public Member Functions

- void **reset** ()
- void **adapt** (CompositeExecutionObject ∗compositeObject, bool force)
- void **initializeAttributeRuleRelation** (Rule ∗topRule, Rule ∗rule)
- void **initializeRuleObjectRelation** (ExecutionObjectSwitch ∗object)
- void **adapt** (ExecutionObjectSwitch ∗objectAlternatives, bool force)
- bool **adaptDescriptor** (ExecutionObject ∗executionObject)
- Node ∗ **adaptSwitch** (SwitchNode ∗switchNode)
- bool **evaluateRule** (Rule ∗rule)
- virtual void **update** (void ∗arg0, void ∗arg1)

Private Member Functions

- bool **evaluateCompositeRule** (CompositeRule ∗rule)
- bool **evaluateSimpleRule** (SimpleRule ∗rule)

Private Attributes

- map< string, vector< Rule ∗ > ∗ > ∗ **ruleListenMap**
- map< Rule ∗, vector< ExecutionObjectSwitch ∗ > ∗ > ∗ **entityListenMap**
- map< Rule ∗, vector< DescriptorSwitch ∗ > ∗ > ∗ **descListenMap**

2.18.1 Detailed Description

Definition at line 96 of file RuleAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adaptation/context/RuleAdapter.h

2.19 ApplicationPlayerAdapter Class Reference

Inheritance diagram for ApplicationPlayerAdapter:Collaboration diagram for ApplicationPlayerAdapter:

Public Member Functions

- **ApplicationPlayerAdapter** (**IPlayerAdapterManager** ∗**manager**)
- void **setNclEditListener** (**INclEditListener** ∗listener)
- virtual bool **hasPrepared** ()
- virtual bool **prepare** (ExecutionObject ∗**object**, FormatterEvent ∗mainEvent)
- virtual bool **start** ()
- virtual bool **stop** ()
- virtual bool **pause** ()
- virtual bool **resume** ()
- virtual bool **abort** ()
- virtual bool **unprepare** ()
- virtual void **naturalEnd** ()
- virtual void **updateStatus** (short code, string parameter="", short type=10)
- virtual void **setCurrentEvent** (FormatterEvent ∗event)=0
- virtual void **flip** ()

Protected Member Functions

- void **prepare** (FormatterEvent ∗event)
- bool **startEvent** (string anchorId, short type)
- bool **stopEvent** (string anchorId, short type)
- bool **abortEvent** (string anchorId, short type)
- bool **pauseEvent** (string anchorId, short type)
- bool **resumeEvent** (string anchorId, short type)
- void **lockEvent** ()
- void **unlockEvent** ()

Protected Attributes

- map< string, FormatterEvent ∗ > ∗ **preparedEvents**
- FormatterEvent ∗ **currentEvent**

Private Attributes

- **INclEditListener** ∗ **editingCommandListener**
- pthread_mutex_t **eventMutex**

2.19.1 Detailed Description

Definition at line 68 of file ApplicationPlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/application/ApplicationPlayerAdapter.h

2.20 LuaPlayerAdapter Class Reference

Inheritance diagram for LuaPlayerAdapter:Collaboration diagram for LuaPlayerAdapter:

Public Member Functions

- **LuaPlayerAdapter** (**IPlayerAdapterManager** ∗**manager**)
- virtual ∼**LuaPlayerAdapter** ()
- void **setCurrentEvent** (FormatterEvent ∗event)
- void **pushEPGEvent** (map< string, string > t)

Protected Member Functions

- void **createPlayer** ()

2.20.1 Detailed Description

Definition at line 67 of file LuaPlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/application/imperative/lua/LuaPlayerAdapter.h

2.21 NCLPlayerAdapter Class Reference

Inheritance diagram for NCLPlayerAdapter:Collaboration diagram for NCLPlayerAdapter:

Public Member Functions

- **NCLPlayerAdapter** (**IPlayerAdapterManager** ∗**manager**)
- virtual ∼**NCLPlayerAdapter** ()
- void **setCurrentEvent** (FormatterEvent ∗event)
- void **flip** ()

Protected Member Functions

- void **createPlayer** ()

2.21.1 Detailed Description

Definition at line 64 of file NCLPlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/application/declarative/ncl/NCLPlayerAdapter.h

2.22 LinksPlayerAdapter Class Reference

Inheritance diagram for LinksPlayerAdapter:Collaboration diagram for LinksPlayerAdapter:

Public Member Functions

- **LinksPlayerAdapter** (**IPlayerAdapterManager** ∗**manager**)
- virtual ∼**LinksPlayerAdapter** ()
- void **createPlayer** ()
- bool **setPropertyValue** (AttributionEvent ∗event, string value, Animation ∗anim)

2.22.1 Detailed Description

Definition at line 77 of file LinksPlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/application/declarative/xhtml/LinksPlayerAdapter.h

2.23 AVPlayerAdapter Class Reference

Inheritance diagram for AVPlayerAdapter:Collaboration diagram for AVPlayerAdapter:

Public Member Functions

- **AVPlayerAdapter** (**IPlayerAdapterManager** ∗**manager**, bool **hasVisual**)
- virtual ∼**AVPlayerAdapter** ()
- bool **instanceOf** (string s)
- bool **getHasVisual** ()

Protected Member Functions

- void **createPlayer** ()
- bool **setPropertyValue** (AttributionEvent ∗event, string value, Animation ∗anim)

Private Attributes

- bool **hasVisual**

2.23.1 Detailed Description

Definition at line 79 of file AVPlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/av/AVPlayerAdapter.h

2.24 ChannelPlayerAdapter Class Reference

Inheritance diagram for ChannelPlayerAdapter:Collaboration diagram for ChannelPlayerAdapter:

Public Member Functions

- **ChannelPlayerAdapter** (**IPlayerAdapterManager** ∗**manager**, bool visual)
- virtual ∼**ChannelPlayerAdapter** ()
- bool **getHasVisual** ()

Protected Member Functions

- void **createPlayer** ()
- bool **setPropertyValue** (AttributionEvent ∗event, string value, Animation ∗anim)

Private Attributes

- bool **hasVisual**

2.24.1 Detailed Description

Definition at line 86 of file ChannelPlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/av/ChannelPlayerAdapter.h

2.25 ProgramAVPlayerAdapter Class Reference

Inheritance diagram for ProgramAVPlayerAdapter:Collaboration diagram for ProgramAVPlayerAdapter:

Public Member Functions

- **ProgramAVPlayerAdapter** (**IPlayerAdapterManager** ∗**manager**)
- virtual ∼**ProgramAVPlayerAdapter** ()
- virtual bool **hasPrepared** ()
- virtual bool **stop** ()
- virtual bool **resume** ()

Protected Member Functions

- void **createPlayer** ()
- bool **setPropertyValue** (AttributionEvent ∗event, string value, Animation ∗anim)

Private Member Functions

- void **updateAVBounds** ()

Private Attributes

- IProgramAV ∗ **pav**

2.25.1 Detailed Description

Definition at line 83 of file ProgramAVPlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/av/tv/ProgramAVPlayerAdapter.h

2.26 FormatterPlayerAdapter Class Reference

Inheritance diagram for FormatterPlayerAdapter:Collaboration diagram for FormatterPlayerAdapter:

Public Member Functions

- **FormatterPlayerAdapter** (**IPlayerAdapterManager** ∗**manager**)
- bool **instanceOf** (string s)
- virtual void **setNclEditListener** (**INclEditListener** ∗listener)
- ISurface ∗ **getObjectDisplay** ()
- virtual bool **hasPrepared** ()
- bool **setKeyHandler** (bool isHandler)
- virtual bool **prepare** (ExecutionObject ∗**object**, FormatterEvent ∗mainEvent)
- virtual bool **start** ()
- virtual bool **stop** ()
- virtual bool **pause** ()
- virtual bool **resume** ()
- virtual bool **abort** ()
- virtual void **naturalEnd** ()
- virtual bool **unprepare** ()
- virtual bool **setPropertyValue** (AttributionEvent ∗event, string value, Animation ∗anim)
- bool **startAttribution** (string propName, string propValue)
- string **getPropertyValue** (void ∗event)
- double **getObjectExpectedDuration** ()
- void **updateObjectExpectedDuration** ()
- double **getMediaTime** ()
- IPlayer ∗ **getPlayer** ()
- void **setTimeBasePlayer** (**FormatterPlayerAdapter** ∗timeBasePlayer)
- virtual void **updateStatus** (short code, string parameter="", short type=10)
- bool **userEventReceived** (IInputEvent ∗ev)
- virtual void **flip** ()
- virtual void **timeShift** (string direction)
- virtual PlayerState ∗ **getPlayerState** ()

Protected Member Functions

- virtual void **createPlayer** ()
- void **prepare** ()
- bool **lockObject** ()
- bool **unlockObject** ()

Protected Attributes

- **IPlayerAdapterManager** ∗ **manager**
- **NominalEventMonitor** ∗ **anchorMonitor**
- set< string > **typeSet**
- ExecutionObject ∗ **object**
- IPlayer ∗ **player**
- string **playerCompName**
- string **mrl**

Static Protected Attributes

- static IInputManager * **im**

Private Member Functions

- bool **checkRepeat** (PresentationEvent *mainEvent)
- void **setVisible** (bool visible)

Private Attributes

- bool **circularSituation**
- bool **isLocked**
- pthread_mutex_t **objectMutex**

2.26.1 Detailed Description

Definition at line 102 of file FormatterPlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/FormatterPlayerAdapter.h

2.27 ImagePlayerAdapter Class Reference

Inheritance diagram for ImagePlayerAdapter:Collaboration diagram for ImagePlayerAdapter:

Public Member Functions

- **ImagePlayerAdapter** (**IPlayerAdapterManager** ∗**manager**)
- virtual ∼**ImagePlayerAdapter** ()

Protected Member Functions

- void **createPlayer** ()

2.27.1 Detailed Description

Definition at line 74 of file ImagePlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/image/ImagePlayerAdapter.h

2.28 INclEditListener Class Reference

Public Member Functions

- virtual ~**INclEditListener** ()
- virtual bool **nclEdit** (string editingCommand)=0

2.28.1 Detailed Description

Definition at line 63 of file INclEditListener.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/INclEditListener.h

2.29 IPlayerAdapter Class Reference

Inheritance diagram for IPlayerAdapter:Collaboration diagram for IPlayerAdapter:

Public Member Functions

- virtual ~**IPlayerAdapter** ()
- virtual IPlayer * **getPlayer** ()=0
- virtual bool **setPropertyValue** (AttributionEvent *event, string value, Animation *anim)=0

Protected Member Functions

- virtual void **createPlayer** ()=0

2.29.1 Detailed Description

Definition at line 75 of file IPlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/IPlayerAdapter.h

2.30 IPlayerAdapterManager Class Reference

Inheritance diagram for IPlayerAdapterManager:

Public Member Functions

- virtual ~**IPlayerAdapterManager** ()
- virtual void **setStandByState** (bool standBy)=0
- virtual NclPlayerData * **getNclPlayerData** ()=0
- virtual ITimeBaseProvider * **getTimeBaseProvider** ()=0
- virtual bool **removePlayer** (void *object)=0

2.30.1 Detailed Description

Definition at line 65 of file IPlayerAdapterManager.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/IPlayerAdapterManager.h

2.31 NominalEventMonitor Class Reference

Inheritance diagram for NominalEventMonitor:Collaboration diagram for NominalEventMonitor:

Public Member Functions

- **NominalEventMonitor** (ExecutionObject ∗obj, **IPlayerAdapter** ∗player)
- void **setTimeBaseProvider** (ITimeBaseProvider ∗**timeBaseProvider**)
- void **updateTimeBaseId** (unsigned char oldContentId, unsigned char newContentId)
- void **valueReached** (unsigned char **timeBaseId**, int64_t timeValue)
- void **loopDetected** ()
- void **startMonitor** ()
- void **pauseMonitor** ()
- void **resumeMonitor** ()
- void **stopMonitor** ()
- virtual void **run** ()

Private Member Functions

- void **prepareNptTransitionsEvents** ()

Private Attributes

- double **sleepTime**
- double **expectedSleepTime**
- ExecutionObject ∗ **executionObject**
- **IPlayerAdapter** ∗ **adapter**
- int **timeBaseId**
- ITimeBaseProvider ∗ **timeBaseProvider**
- bool **running**
- bool **deleting**
- bool **paused**
- bool **stopped**

Static Private Attributes

- static const double **DEFAULT_SLEEP_TIME** = 5000
- static const double **DEFAULT_ERROR** = 50.0

2.31.1 Detailed Description

Definition at line 89 of file NominalEventMonitor.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/NominalEventMonitor.h

2.32 PlayerAdapterManager Class Reference

Inheritance diagram for PlayerAdapterManager:Collaboration diagram for PlayerAdapterManager:

Public Member Functions

- **PlayerAdapterManager** (NclPlayerData ∗**nclPlayerData**)
- NclPlayerData ∗ **getNclPlayerData** ()
- void **setTimeBaseProvider** (ITimeBaseProvider ∗**timeBaseProvider**)
- ITimeBaseProvider ∗ **getTimeBaseProvider** ()
- void **setVisible** (string objectId, string visible, AttributionEvent ∗event)
- bool **removePlayer** (void ∗object)
- void **setNclEditListener** (**INclEditListener** ∗listener)
- **FormatterPlayerAdapter** ∗ **getPlayer** (ExecutionObject ∗execObj)
- void **setStandByState** (bool standBy)
- void **pushEPGEventToEPGFactory** (map< string, string > t)
- void **timeShift** (string direction)

Static Public Member Functions

- static bool **isEmbeddedApp** (NodeEntity ∗dataObject)

Protected Member Functions

- void **clearDeletePlayers** ()
- void **run** ()

Private Member Functions

- string **getPlayerClass** (CascadingDescriptor ∗descriptor, NodeEntity ∗dataObject)
- bool **removePlayer** (string objectId)
- void **clear** ()
- void **readConfigFiles** ()
- **FormatterPlayerAdapter** ∗ **initializePlayer** (ExecutionObject ∗object)

Static Private Member Functions

- static bool **isEmbeddedAppMediaType** (string mediaType)

Private Attributes

- map< string, **IPlayerAdapter** ∗ > ∗ **objectPlayers**
- map< string, **IPlayerAdapter** ∗ > ∗ **deletePlayers**
- map< **IPlayerAdapter** ∗, string > ∗ **playerNames**
- map< string, string > ∗ **mimeDefaultTable**
- map< string, string > ∗ **playerTable**
- **INclEditListener** ∗ **editingCommandListener**

- **IPlayerAdapter** ∗ **epgFactoryAdapter**
- bool **running**
- ITimeBaseProvider ∗ **timeBaseProvider**
- NclPlayerData ∗ **nclPlayerData**
- pthread_mutex_t **mutexPlayer**

2.32.1 Detailed Description

Definition at line 85 of file PlayerAdapterManager.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/PlayerAdapterManager.h

2.33 RemotePlayerAdapter Class Reference

Inheritance diagram for RemotePlayerAdapter:Collaboration diagram for RemotePlayerAdapter:

Public Member Functions

- void **setNclEditListener** (**INclEditListener** ∗listener)
- virtual bool **hasPrepared** ()
- virtual bool **prepare** (ExecutionObject ∗**object**, FormatterEvent ∗mainEvent)
- virtual bool **start** ()
- virtual bool **stop** ()
- virtual bool **pause** ()
- virtual bool **resume** ()
- virtual bool **abort** ()
- virtual bool **unprepare** ()
- virtual void **naturalEnd** ()
- virtual void **updateStatus** (short code, string parameter="", short type=10)
- virtual void **setCurrentEvent** (FormatterEvent ∗event)=0

Protected Member Functions

- void **prepare** (FormatterEvent ∗event)
- bool **startEvent** (string anchorId, short type)
- bool **stopEvent** (string anchorId, short type)
- bool **abortEvent** (string anchorId, short type)
- bool **pauseEvent** (string anchorId, short type)
- bool **resumeEvent** (string anchorId, short type)
- void **lockEvent** ()
- void **unlockEvent** ()
- FormatterEvent ∗ **getEvent** (string interfaceId)

Protected Attributes

- map< string, FormatterEvent ∗ > ∗ **preparedEvents**
- FormatterEvent ∗ **currentEvent**

Private Attributes

- **INclEditListener** ∗ **editingCommandListener**
- pthread_mutex_t **eventMutex**

2.33.1 Detailed Description

Definition at line 71 of file RemotePlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/remote/RemotePlayerAdapter.h

2.34 PlainTxtPlayerAdapter Class Reference

Inheritance diagram for PlainTxtPlayerAdapter:Collaboration diagram for PlainTxtPlayerAdapter:

Public Member Functions

- **PlainTxtPlayerAdapter** (**IPlayerAdapterManager** ∗**manager**)
- virtual ∼**PlainTxtPlayerAdapter** ()

Protected Member Functions

- void **createPlayer** ()

2.34.1 Detailed Description

Definition at line 66 of file PlainTxtPlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/text/PlainTxtPlayerAdapter.h

2.35 SubtitlePlayerAdapter Class Reference

Inheritance diagram for SubtitlePlayerAdapter:Collaboration diagram for SubtitlePlayerAdapter:

Public Member Functions

- **SubtitlePlayerAdapter** (**IPlayerAdapterManager** ∗**manager**)
- virtual ∼**SubtitlePlayerAdapter** ()
- bool **instanceOf** (string s)

Protected Member Functions

- void **createPlayer** ()

2.35.1 Detailed Description

Definition at line 80 of file SubtitlePlayerAdapter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/adapters/text/SubtitlePlayerAdapter.h

2.36 AnimationController Class Reference

Inheritance diagram for AnimationController:Collaboration diagram for AnimationController:

Static Public Member Functions

- static void **startAnimation** (ExecutionObject ∗**execObj**, FormatterPlayerAdapter ∗**player**, AttributionEvent ∗**event**, string value, Animation ∗anim)

Properties

- vector< string > **SingleNames**
- vector< double > **initValues**
- vector< string > ∗ **strTargetValues**
- vector< double > ∗ **targetValues**
- double **initTime**
- double **duration**
- int **stepSize**

Private Member Functions

- **AnimationController** (ExecutionObject ∗**execObj**, FormatterPlayerAdapter ∗**player**, AttributionEvent ∗**event**, string value, Animation ∗anim)
- void **run** ()
- bool **loadInitValues** ()
- bool **loadTargetValues** ()
- bool **animeStep** ()
- double **getSinglePropertyTarget** (int i)
- bool **updateTargetRegion** ()
- bool **ungroupProperty** ()
- bool **isExecutionObjectProperty** (string)

Private Attributes

- Animation ∗ **animation**
- FormatterPlayerAdapter ∗ **player**
- ExecutionObject ∗ **execObj**
- AttributionEvent ∗ **event**
- LayoutRegion ∗ **initRegion**
- LayoutRegion ∗ **targetRegion**
- string **propName**
- bool **isExecObjAnim**
- vector< string > ∗ **params**

2.36.1 Detailed Description

Definition at line 76 of file AnimationController.h.

2.36.2 Member Function Documentation

2.36.2.1 void run () [private]

This method will execute the **animation** (p. **??**).

WARNING: this method calls "delete this" when the **animation** (p. **??**) is done. So, the user of this class should never call delete to the pointer that creates the **animation** (p. **??**). Additionally, this object always has to be instantiate as a pointer, never as a variable (when the destructor is called automatically when the scope finishes).

2.36.2.2 bool loadInitValues () [private]

Load the current values from the properties will be animated in the object to the initValues vector.

2.36.2.3 bool loadTargetValues () [private]

Based on initValues and the parameters passed by the author this function calculate and load the target values with the final value for each property that must be changed

2.36.2.4 double getSinglePropertyTarget (int *i*) [private]

Return the value of the the target from a single property

2.36.2.5 bool updateTargetRegion () [private]

The targetRegion variable is used to calculate the value of dimension and position the region will be when the **animation** (p. **??**) ends. First of all, this targetRegion will be a copy of the initial region, so this function is responsible to, based on initial region and the values in targetValues vector, update the targetRegion to have the final parameters.

2.36.2.6 bool ungroupProperty () [private]

The property can be a single name that groups a lot of single properties like location, size, bound, etc. So, this function is responsible to separate this property name and the parameters passed by the user in single property names loading the vectors: propertySingleNames and strTargetValues

The documentation for this class was generated from the following file:

- gingancl-cpp/include/animation/AnimationController.h

2.37 FormatterConverter Class Reference

Inheritance diagram for FormatterConverter:Collaboration diagram for FormatterConverter:

Public Member Functions

- **FormatterConverter** (RuleAdapter ∗**ruleAdapter**)
- void **stopListening** (void ∗event)
- set< ExecutionObject ∗ > ∗ **getRunningObjects** ()
- void **setHandlingStatus** (bool hanling)
- ExecutionObject ∗ **getObjectFromNodeId** (string id)
- void **setScheduler** (**FormatterScheduler** ∗**scheduler**)
- void **setDepthLevel** (int level)
- int **getDepthLevel** ()
- void **checkGradSameInstance** (set< ReferNode ∗ > ∗gradSame, ExecutionObject ∗object)
- CompositeExecutionObject ∗ **addSameInstance** (ExecutionObject ∗executionObject, ReferNode ∗referNode)
- void **compileExecutionObjectLinks** (ExecutionObject ∗executionObject, int **depthLevel**)
- ExecutionObject ∗ **getExecutionObject** (NodeNesting ∗perspective, GenericDescriptor ∗descriptor, int **depthLevel**) throw (ObjectCreationForbiddenException∗)
- set< ExecutionObject ∗ > ∗ **getSettingNodeObjects** ()
- FormatterEvent ∗ **getEvent** (ExecutionObject ∗executionObject, InterfacePoint ∗interfacePoint, int ncmEventType, string key)
- void **compileExecutionObjectLinks** (ExecutionObject ∗executionObject, Node ∗dataObject, CompositeExecutionObject ∗parentObject, int **depthLevel**)
- ExecutionObject ∗ **processExecutionObjectSwitch** (ExecutionObjectSwitch ∗switchObject)
- FormatterEvent ∗ **insertContext** (NodeNesting ∗contextPerspective, Port ∗port)
- bool **removeExecutionObject** (ExecutionObject ∗executionObject, ReferNode ∗referNode)
- bool **removeExecutionObject** (ExecutionObject ∗executionObject)
- ExecutionObject ∗ **hasExecutionObject** (Node ∗node, GenericDescriptor ∗descriptor)
- FormatterCausalLink ∗ **addCausalLink** (ContextNode ∗context, CausalLink ∗link)
- void **eventStateChanged** (void ∗someEvent, short transition, short previousState)
- short **getPriorityType** ()
- void **reset** ()

Static Public Member Functions

- static CascadingDescriptor ∗ **getCascadingDescriptor** (NodeNesting ∗nodePerspective, GenericDescriptor ∗descriptor)

Private Member Functions

- void **addExecutionObject** (ExecutionObject ∗executionObject, CompositeExecutionObject ∗parentObject, int **depthLevel**)
- CompositeExecutionObject ∗ **getParentExecutionObject** (NodeNesting ∗perspective, int **depthLevel**) throw (ObjectCreationForbiddenException∗)
- void **createMultichannelObject** (CompositeExecutionObject ∗compositeObject, int **depthLevel**)
- ExecutionObject ∗ **createExecutionObject** (string id, NodeNesting ∗perspective, CascadingDescriptor ∗descriptor, int **depthLevel**)

- void **setActionListener** (LinkAction ∗action)
- void **resolveSwitchEvents** (ExecutionObjectSwitch ∗switchObject, int **depthLevel**)
- FormatterEvent ∗ **insertNode** (NodeNesting ∗perspective, InterfacePoint ∗interfacePoint, GenericDescriptor ∗descriptor)

Private Attributes

- int **depthLevel**
- map< string, ExecutionObject ∗ > ∗ **executionObjects**
- set< ExecutionObject ∗ > ∗ **settingObjects**
- void ∗ **linkCompiler**
- **FormatterScheduler** ∗ **scheduler**
- RuleAdapter ∗ **ruleAdapter**
- pthread_mutex_t **objectsMutex**
- bool **handling**

2.37.1 Detailed Description

Definition at line 131 of file FormatterConverter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/emconverter/FormatterConverter.h

2.38 FormatterLinkConverter Class Reference

Collaboration diagram for FormatterLinkConverter:

Public Member Functions

- **FormatterLinkConverter** (**FormatterConverter** ∗**compiler**)
- FormatterCausalLink ∗ **createCausalLink** (CausalLink ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)

Private Member Functions

- LinkAction ∗ **createAction** (Action ∗actionExpression, CausalLink ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)
- LinkCondition ∗ **createCondition** (ConditionExpression ∗ncmExpression, CausalLink ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)
- LinkCompoundTriggerCondition ∗ **createCompoundTriggerCondition** (short op, double delay, vector< ConditionExpression ∗ > ∗ncmChildConditions, CausalLink ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)
- LinkCondition ∗ **createCondition** (TriggerExpression ∗triggerExpression, CausalLink ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)
- LinkAssessmentStatement ∗ **createAssessmentStatement** (AssessmentStatement ∗assessmentStatement, Bind ∗bind, Link ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)
- LinkStatement ∗ **createStatement** (Statement ∗statementExpression, Link ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)
- LinkAttributeAssessment ∗ **createAttributeAssessment** (AttributeAssessment ∗attributeAssessment, Bind ∗bind, Link ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)
- LinkSimpleAction ∗ **createSimpleAction** (SimpleAction ∗sae, Bind ∗bind, Link ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)
- LinkCompoundAction ∗ **createCompoundAction** (short op, double delay, vector< Action ∗ > ∗ncmChildActions, CausalLink ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)
- LinkTriggerCondition ∗ **createSimpleCondition** (SimpleCondition ∗condition, Bind ∗bind, Link ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)
- FormatterEvent ∗ **createEvent** (Bind ∗bind, Link ∗ncmLink, CompositeExecutionObject ∗parentObject, int depthLevel)
- double **getDelayParameter** (Link ∗ncmLink, Parameter ∗connParam, Bind ∗ncmBind)
- string **getBindKey** (Link ∗ncmLink, Bind ∗ncmBind)
- double **compileDelay** (Link ∗ncmLink, string delayObject, Bind ∗bind)

Private Attributes

- **FormatterConverter** ∗ **compiler**

2.38.1 Detailed Description

Definition at line 123 of file FormatterLinkConverter.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/emconverter/FormatterLinkConverter.h

2.39 ObjectCreationForbiddenException Class Reference

Inheritance diagram for ObjectCreationForbiddenException:Collaboration diagram for ObjectCreationForbiddenException:

Private Member Functions

- virtual const char ∗ **what** () const throw ()

2.39.1 Detailed Description

Definition at line 61 of file ObjectCreationForbiddenException.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/emconverter/ObjectCreationForbiddenException.h

2.40 FormatterFocusManager Class Reference

Inheritance diagram for FormatterFocusManager:Collaboration diagram for FormatterFocusManager:

Public Member Functions

- **FormatterFocusManager** (PlayerAdapterManager ∗**playerManager**, IFormatterMultiDevice ∗**multiDevice**, void ∗**converter**)
- void **setParent** (**FormatterFocusManager** ∗parent)
- bool **setKeyHandler** (bool **isHandler**)
- void **setKeyMaster** (string focusIndex)
- void **setStandByState** (bool standBy)
- void **setFocus** (string focusIndex)
- void **showObject** (ExecutionObject ∗object)
- void **hideObject** (ExecutionObject ∗object)
- void **setDefaultFocusBorderColor** (Color ∗color)
- void **setDefaultFocusBorderWidth** (int width)
- void **setDefaultSelBorderColor** (Color ∗color)
- bool **userEventReceived** (IInputEvent ∗ev)

Private Member Functions

- void **setHandlingObjects** (bool isHandling)
- void **setFocus** (CascadingDescriptor ∗descriptor)
- void **recoveryDefaultState** (ExecutionObject ∗object)
- ExecutionObject ∗ **getObjectFromFocusIndex** (string focusIndex)
- void **insertObject** (ExecutionObject ∗object, string focusIndex)
- void **removeObject** (ExecutionObject ∗object, string focusIndex)
- bool **keyCodeOk** (ExecutionObject ∗currentObject)
- bool **keyCodeBack** ()
- bool **enterSelection** (FormatterPlayerAdapter ∗player)
- void **exitSelection** (FormatterPlayerAdapter ∗player)
- void **registerNavigationKeys** ()
- void **registerBackKeys** ()
- void **unregister** ()
- void **changeSettingState** (string name, string act)

Private Attributes

- map< string, set< ExecutionObject ∗ > ∗ > ∗ **focusTable**
- vector< string > ∗ **focusSequence**
- bool **isHandler**
- string **currentFocus**
- string **objectToSelect**
- ExecutionObject ∗ **selectedObject**
- Color ∗ **defaultFocusBorderColor**
- int **defaultFocusBorderWidth**
- Color ∗ **defaultSelBorderColor**
- PlayerAdapterManager ∗ **playerManager**

- **FormatterFocusManager** ∗ **parentManager**
- IFormatterMultiDevice ∗ **multiDevice**
- void ∗ **converter**
- pthread_mutex_t **mutexFocus**
- pthread_mutex_t **mutexTable**

Static Private Attributes

- static IInputManager ∗ **im**

2.40.1 Detailed Description

Definition at line 93 of file FormatterFocusManager.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/focus/FormatterFocusManager.h

2.41 FormatterMediator Class Reference

Inheritance diagram for FormatterMediator:Collaboration diagram for FormatterMediator:

Public Member Functions

- **FormatterMediator** (NclPlayerData ∗**data**)
- void **printData** (NclPlayerData ∗**data**)
- void **setStandByState** (bool standBy)
- void **setTimeBaseProvider** (ITimeBaseProvider ∗**timeBaseProvider**)
- void **setBackgroundImage** (string uri)
- void **setParentLayout** (void ∗parentLayout)
- string **getScreenShot** ()
- vector< string > ∗ **getCompileErrors** ()
- void ∗ **setCurrentDocument** (string fName)
- void **setDepthLevel** (int level)
- int **getDepthLevel** ()
- bool **nclEdit** (string nclEditApi)
- bool **editingCommand** (string commandTag, string privateDataPayload)
- void **setNotifyContentUpdate** (bool notify)
- void **addListener** (IPlayerListener ∗listener)
- void **removeListener** (IPlayerListener ∗listener)
- void **notifyListeners** (short code, string paremeter, short type)
- void **setSurface** (io::ISurface ∗surface)
- io::ISurface ∗ **getSurface** ()
- void **flip** ()
- double **getMediaTime** ()
- void **setMediaTime** (double newTime)
- bool **setKeyHandler** (bool isHandler)
- void **setScope** (string scope, short type, double begin=-1, double end=-1)
- void **play** ()
- void **stop** ()
- void **abort** ()
- void **pause** ()
- void **resume** ()
- string **getPropertyValue** (string name)
- void **setPropertyValue** (string name, string value, double duration=-1, double by=-1)
- void **setReferenceTimePlayer** (IPlayer ∗player)
- void **addTimeReferPlayer** (IPlayer ∗referPlayer)
- void **removeTimeReferPlayer** (IPlayer ∗referPlayer)
- void **notifyReferPlayers** (int transition)
- void **timebaseObjectTransitionCallback** (int transition)
- void **setTimeBasePlayer** (IPlayer ∗timeBasePlayer)
- bool **hasPresented** ()
- void **setPresented** (bool presented)
- bool **isVisible** ()
- void **setVisible** (bool visible)
- bool **immediatelyStart** ()
- void **setImmediatelyStart** (bool immediatelyStartVal)

- void **forceNaturalEnd** ()
- bool **isForcedNaturalEnd** ()
- bool **setOutWindow** (io::IWindow ∗w)
- IPlayer ∗ **getSelectedPlayer** ()
- void **setPlayerMap** (map< string, IPlayer ∗ > ∗objs)
- map< string, IPlayer ∗ > ∗ **getPlayerMap** ()
- IPlayer ∗ **getPlayer** (string objectId)
- void **select** (IPlayer ∗selObject)
- void **setCurrentScope** (string scopeId)
- string **getActiveUris** (vector< string > ∗uris)
- string **getDepUris** (vector< string > ∗uris, int targetDev=0)
- void **timeShift** (string direction)
- PlayerState ∗ **getPlayerState** ()

Static Public Member Functions

- static void **release** ()

Private Member Functions

- virtual void ∗ **addDocument** (string fName)
- bool **removeDocument** (string documentId)
- ContextNode ∗ **getDocumentContext** (string documentId)
- vector< FormatterEvent ∗ > ∗ **processDocument** (string documentId, string interfaceId)
- vector< FormatterEvent ∗ > ∗ **getDocumentEntryEvent** (string documentId)
- bool **compileDocument** (string documentId, string interfaceId)
- bool **prepareDocument** (string documentId)
- void **solveRemoteDescriptorsUris** (string docLocation, vector< GenericDescriptor ∗ > ∗descs, bool isRemoteDoc)
- void **solveRemoteNodesUris** (string docLocation, vector< Node ∗ > ∗nodes, bool isRemoteDoc)
- void **solveRemoteNclDeps** (string docLocation, bool isRemoteDoc)
- void **solveRemoteLuaDeps** (string docLocation, string src, bool isRemoteDoc)
- string **solveRemoteSourceUri** (string docLocation, string src)
- bool **startDocument** (string documentId, string interfaceId)
- bool **stopDocument** (string documentId)
- bool **pauseDocument** (string documentId)
- bool **resumeDocument** (string documentId)
- void **presentationCompleted** (IFormatterEvent ∗documentEvent)
- LayoutRegion ∗ **addRegion** (string documentId, string regionBaseId, string regionId, string xmlRegion)
- LayoutRegion ∗ **removeRegion** (string documentId, string regionBaseId, string regionId)
- RegionBase ∗ **addRegionBase** (string documentId, string xmlRegionBase)
- RegionBase ∗ **removeRegionBase** (string documentId, string regionBaseId)
- Rule ∗ **addRule** (string documentId, string xmlRule)
- Rule ∗ **removeRule** (string documentId, string ruleId)
- RuleBase ∗ **addRuleBase** (string documentId, string xmlRuleBase)
- RuleBase ∗ **removeRuleBase** (string documentId, string ruleBaseId)
- Transition ∗ **addTransition** (string documentId, string xmlTransition)
- Transition ∗ **removeTransition** (string documentId, string transitionId)

- TransitionBase * **addTransitionBase** (string documentId, string xmlTransitionBase)
- TransitionBase * **removeTransitionBase** (string documentId, string transitionBaseId)
- Connector * **addConnector** (string documentId, string xmlConnector)
- Connector * **removeConnector** (string documentId, string connectorId)
- ConnectorBase * **addConnectorBase** (string documentId, string xmlConnectorBase)
- ConnectorBase * **removeConnectorBase** (string documentId, string connectorBaseId)
- GenericDescriptor * **addDescriptor** (string documentId, string xmlDescriptor)
- GenericDescriptor * **removeDescriptor** (string documentId, string descriptorId)
- DescriptorBase * **addDescriptorBase** (string documentId, string xmlDescriptorBase)
- DescriptorBase * **removeDescriptorBase** (string documentId, string descriptorBaseId)
- Base * **addImportBase** (string documentId, string docBaseId, string xmlImportBase)
- Base * **removeImportBase** (string documentId, string docBaseId, string documentURI)
- NclDocument * **addImportedDocumentBase** (string documentId, string xmlImportedDocumentBase)
- NclDocument * **removeImportedDocumentBase** (string documentId, string importedDocumentBaseId)
- NclDocument * **addImportNCL** (string documentId, string xmlImportNCL)
- NclDocument * **removeImportNCL** (string documentId, string documentURI)
- void **processInsertedReferNode** (ReferNode *referNode)
- void **processInsertedComposition** (CompositeNode *composition)
- Node * **addNode** (string documentId, string compositeId, string xmlNode)
- Node * **removeNode** (string documentId, string compositeId, string nodeId)
- InterfacePoint * **addInterface** (string documentId, string nodeId, string xmlInterface)
- void **removeInterfaceMappings** (Node *node, InterfacePoint *interfacePoint, CompositeNode *composition)
- void **removeInterfaceLinks** (Node *node, InterfacePoint *interfacePoint, LinkComposition *composition)
- void **removeInterface** (Node *node, InterfacePoint *interfacePoint)
- InterfacePoint * **removeInterface** (string documentId, string nodeId, string interfaceId)
- Link * **addLink** (string documentId, string compositeId, string xmlLink)
- void **removeLink** (LinkComposition *composition, Link *link)
- Link * **removeLink** (string documentId, string compositeId, string linkId)
- bool **setPropertyValue** (string documentId, string nodeId, string propertyId, string value)
- void **pushEPGEventToEPGFactory** (map< string, string > t)
- string **getDepUrisFromNodes** (vector< string > *uris, vector< Node * > *nodes, int targetDev=0)
- string **getDepUriFromNode** (vector< string > *uris, Node *node, int targetDev=0)
- string **getBaseUri** (string baseA, string baseB)

Private Attributes

- NclPlayerData * **data**
- string **currentFile**
- NclDocument * **currentDocument**
- map< string, FormatterEvent * > * **documentEvents**
- map< string, vector< FormatterEvent * > * > * **documentEntryEvents**
- **FormatterScheduler** * **scheduler**
- RuleAdapter * **ruleAdapter**
- FormatterConverter * **compiler**
- PlayerAdapterManager * **playerManager**
- ITimeBaseProvider * **timeBaseProvider**

- vector< string > * **compileErrors**
- IDeviceLayout * **deviceLayout**
- IFormatterMultiDevice * **multiDevice**
- bool **isEmbedded**

Static Private Attributes

- static **PrivateBaseManager** * **privateBaseManager**
- static IPrefetchManager * **pm**

2.41.1 Detailed Description

Definition at line 157 of file FormatterMediator.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/FormatterMediator.h

2.42 FormatterScheduler Class Reference

Inheritance diagram for FormatterScheduler:Collaboration diagram for FormatterScheduler:

Public Member Functions

- **FormatterScheduler** (PlayerAdapterManager ∗**playerManager**, RuleAdapter ∗**ruleAdapter**, IFormatterMultiDevice ∗multiDevice, void ∗**compiler**)
- void **stopListening** (void ∗event)
- bool **setKeyHandler** (bool isHandler)
- void **setStandByState** (bool standBy)
- FormatterFocusManager ∗ **getFocusManager** ()
- FormatterLayout ∗ **getFormatterLayout** (CascadingDescriptor ∗descriptor)
- void **runAction** (void ∗action)
- void **startEvent** (FormatterEvent ∗event)
- void **stopEvent** (FormatterEvent ∗event)
- void **pauseEvent** (FormatterEvent ∗event)
- void **resumeEvent** (FormatterEvent ∗event)
- void **startDocument** (FormatterEvent ∗documentEvent, vector< FormatterEvent ∗ > ∗entryEvents)
- void **stopDocument** (FormatterEvent ∗documentEvent)
- void **pauseDocument** (FormatterEvent ∗documentEvent)
- void **resumeDocument** (FormatterEvent ∗documentEvent)
- void **stopAllDocuments** ()
- void **pauseAllDocuments** ()
- void **resumeAllDocuments** ()
- void **eventStateChanged** (void ∗someEvent, short transition, short previousState)
- short **getPriorityType** ()
- void **addSchedulerListener** (**IFormatterSchedulerListener** ∗listener)
- void **removeSchedulerListener** (**IFormatterSchedulerListener** ∗listener)

Private Member Functions

- bool **isDocumentRunning** (FormatterEvent ∗event)
- void **setTimeBaseObject** (ExecutionObject ∗object, FormatterPlayerAdapter ∗objectPlayer, string nodeId)
- void **runAction** (FormatterEvent ∗event, void ∗action)
- void **runActionOverProperty** (FormatterEvent ∗event, LinkSimpleAction ∗action)
- void **runActionOverApplicationObject** (ApplicationExecutionObject ∗executionObject, FormatterEvent ∗event, FormatterPlayerAdapter ∗player, LinkSimpleAction ∗action)
- void **runActionOverComposition** (CompositeExecutionObject ∗compositeObject, LinkSimpleAction ∗action)
- void **runActionOverSwitch** (ExecutionObjectSwitch ∗switchObject, SwitchEvent ∗event, LinkSimpleAction ∗action)
- void **runSwitchEvent** (ExecutionObjectSwitch ∗switchObject, SwitchEvent ∗switchEvent, ExecutionObject ∗selectedObject, LinkSimpleAction ∗action)
- string **solveImplicitRefAssessment** (string propValue, LinkAssignmentAction ∗action, AttributionEvent ∗event)
- void **initializeDefaultSettings** ()
- void **initializeDocumentSettings** (Node ∗node)
- void **removeDocument** (FormatterEvent ∗documentEvent)

Private Attributes

- RuleAdapter * **ruleAdapter**
- PlayerAdapterManager * **playerManager**
- IFormatterMultiDevice * **layoutManager**
- FormatterFocusManager * **focusManager**
- void * **compiler**
- vector< **IFormatterSchedulerListener** * > * **schedulerListeners**
- vector< FormatterEvent * > * **documentEvents**
- map< FormatterEvent *, bool > * **documentStatus**
- set< string > **typeSet**

2.42.1 Detailed Description

Definition at line 130 of file FormatterScheduler.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/FormatterScheduler.h

2.43 GingaNclGfx Class Reference

Collaboration diagram for GingaNclGfx:

Static Public Member Functions

- static void **show** (int l, int t, int **w**, int **h**)
- static void **release** ()
- static void **showTop** (string text)
- static void **showBottom** (string text)

Static Private Member Functions

- static void **roll** (vector< string > *mrls)
- static AnimePlayer * **createAnimePlayer** (vector< string > *mrls)
- static void **showText** (IWindow *window, string text)

Static Private Attributes

- static int **w**
- static int **h**
- static vector< AnimePlayer * > * **animes**
- static IWindow * **bg**
- static IWindow * **top**
- static IWindow * **bottom**

2.43.1 Detailed Description

Definition at line 71 of file GingaNclGfx.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/gfx/GingaNclGfx.h

2.44 IFormatterSchedulerListener Class Reference

Inheritance diagram for IFormatterSchedulerListener:

Public Member Functions

- virtual ~**IFormatterSchedulerListener** ()
- virtual void **presentationCompleted** (IFormatterEvent *documentEvent)=0

2.44.1 Detailed Description

Definition at line 61 of file IFormatterSchedulerListener.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/IFormatterSchedulerListener.h

2.45 IPrivateBaseManager Class Reference

Inheritance diagram for IPrivateBaseManager:

Public Member Functions

- virtual string **getDocumentLocation** (string id, string docId)=0

2.45.1 Detailed Description

Definition at line 61 of file IPrivateBaseManager.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/privatebase/IPrivateBaseManager.h

2.46 ApplicationExecutionObject Class Reference

Inheritance diagram for ApplicationExecutionObject:Collaboration diagram for ApplicationExecutionObject:

Public Member Functions

- **ApplicationExecutionObject** (string **id**, Node ∗node, bool handling)
- **ApplicationExecutionObject** (string **id**, Node ∗node, GenericDescriptor ∗**descriptor**, bool handling)
- **ApplicationExecutionObject** (string **id**, Node ∗node, CascadingDescriptor ∗**descriptor**, bool handling)
- void **setCurrentEvent** (FormatterEvent ∗event)
- bool **prepare** (FormatterEvent ∗event, double **offsetTime**)
- bool **start** ()
- bool **stop** ()
- bool **abort** ()
- bool **pause** ()
- bool **resume** ()
- bool **unprepare** ()

Private Member Functions

- void **initializeApplicationObject** ()

Private Attributes

- map< string, FormatterEvent ∗ > ∗ **preparedEvents**
- FormatterEvent ∗ **currentEvent**

2.46.1 Detailed Description

Definition at line 63 of file ApplicationExecutionObject.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/ApplicationExecutionObject.h

2.47 CompositeExecutionObject Class Reference

Inheritance diagram for CompositeExecutionObject:Collaboration diagram for CompositeExecutionObject:

Public Member Functions

- **CompositeExecutionObject** (string **id**, Node ∗**dataObject**, bool handling)
- **CompositeExecutionObject** (string **id**, Node ∗**dataObject**, CascadingDescriptor ∗**descriptor**, bool handling)
- void **stopListening** (void ∗event)
- bool **addExecutionObject** (**ExecutionObject** ∗execObj)
- bool **containsExecutionObject** (string execObjId)
- **ExecutionObject** ∗ **getExecutionObject** (string execObjId)
- map< string, **ExecutionObject** ∗ > ∗ **getExecutionObjects** ()
- map< string, **ExecutionObject** ∗ > ∗ **recursivellyGetExecutionObjects** ()
- int **getNumExecutionObjects** ()
- bool **removeExecutionObject** (**ExecutionObject** ∗execObj)
- set< Link ∗ > ∗ **getUncompiledLinks** ()
- bool **containsUncompiledLink** (Link ∗dataLink)
- void **removeLinkUncompiled** (Link ∗ncmLink)
- void **setLinkCompiled** (FormatterLink ∗formatterLink)
- void **setLinkUncompiled** (FormatterLink ∗formatterLink)
- void **addNcmLink** (Link ∗ncmLink)
- void **removeNcmLink** (Link ∗ncmLink)
- void **setAllLinksAsUncompiled** (bool isRecursive)
- set< FormatterLink ∗ > ∗ **getLinks** ()
- void **setParentsAsListeners** ()
- void **unsetParentsAsListeners** ()
- void **eventStateChanged** (void ∗event, short transition, short previousState)
- short **getPriorityType** ()
- void **linkEvaluationStarted** (FormatterCausalLink ∗link)
- void **linkEvaluationFinished** (FormatterCausalLink ∗link, bool start)
- bool **setPropertyValue** (AttributionEvent ∗event, string value, Animation ∗anim)

Protected Member Functions

- void **initializeCompositeExecutionObject** (string **id**, Node ∗**dataObject**, CascadingDescriptor ∗**descriptor**)

Protected Attributes

- map< string, **ExecutionObject** ∗ > ∗ **execObjList**
- pthread_mutex_t **mutexComposite**

Private Member Functions

- void **listRunningObjects** ()
- void **lockComposite** ()
- void **unlockComposite** ()

Private Attributes

- set< FormatterLink * > * **links**
- set< Link * > * **uncompiledLinks**
- set< FormatterEvent * > * **runningEvents**
- set< FormatterEvent * > * **pausedEvents**
- short **lastTransition**
- map< FormatterLink *, int > * **pendingLinks**

2.47.1 Detailed Description

Definition at line 90 of file CompositeExecutionObject.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/CompositeExecutionObject.h

2.48 ExecutionObject Class Reference

Inheritance diagram for ExecutionObject:Collaboration diagram for ExecutionObject:

Public Member Functions

- **ExecutionObject** (string **id**, Node ∗node, bool handling)
- **ExecutionObject** (string **id**, Node ∗node, GenericDescriptor ∗**descriptor**, bool handling)
- **ExecutionObject** (string **id**, Node ∗node, CascadingDescriptor ∗**descriptor**, bool handling)
- bool **instanceOf** (string s)
- int **compareToUsingId** (**ExecutionObject** ∗object)
- Node ∗ **getDataObject** ()
- CascadingDescriptor ∗ **getDescriptor** ()
- string **getId** ()
- void ∗ **getParentObject** ()
- void ∗ **getParentObject** (Node ∗node)
- void **addParentObject** (void ∗parentObject, Node ∗parentNode)
- void **addParentObject** (Node ∗node, void ∗parentObject, Node ∗parentNode)
- void **setDescriptor** (CascadingDescriptor ∗cascadingDescriptor)
- void **setDescriptor** (GenericDescriptor ∗**descriptor**)
- string **toString** ()
- virtual bool **addEvent** (FormatterEvent ∗event)
- void **addPresentationEvent** (PresentationEvent ∗event)
- int **compareTo** (**ExecutionObject** ∗object)
- int **compareToUsingStartTime** (**ExecutionObject** ∗object)
- bool **containsEvent** (FormatterEvent ∗event)
- FormatterEvent ∗ **getEventFromAnchorId** (string anchorId)
- FormatterEvent ∗ **getEvent** (string **id**)
- vector< FormatterEvent ∗ > ∗ **getEvents** ()
- set< AnchorEvent ∗ > ∗ **getSampleEvents** ()
- double **getExpectedStartTime** ()
- PresentationEvent ∗ **getWholeContentPresentationEvent** ()
- void **setStartTime** (double t)
- void **updateEventDurations** ()
- void **updateEventDuration** (PresentationEvent ∗event)
- bool **removeEvent** (FormatterEvent ∗event)
- bool **isCompiled** ()
- void **setCompiled** (bool status)
- void **removeNode** (Node ∗node)
- vector< Node ∗ > ∗ **getNodes** ()
- **NodeNesting** ∗ **getNodePerspective** ()
- **NodeNesting** ∗ **getNodePerspective** (Node ∗node)
- vector< **ExecutionObject** ∗ > ∗ **getObjectPerspective** ()
- vector< **ExecutionObject** ∗ > ∗ **getObjectPerspective** (Node ∗node)
- vector< Node ∗ > ∗ **getParentNodes** ()
- FormatterEvent ∗ **getMainEvent** ()
- virtual bool **prepare** (FormatterEvent ∗event, double **offsetTime**)
- virtual bool **start** ()
- void **updateTransitionTable** (double value, IPlayer ∗player, short int transType)

- void **resetTransitionEvents** (short int transType)
- void **prepareTransitionEvents** (short int transType, double **startTime**)
- set< double > * **getTransitionsValues** (short int transType)
- EventTransition * **getNextTransition** ()
- virtual bool **stop** ()
- virtual bool **abort** ()
- virtual bool **pause** ()
- virtual bool **resume** ()
- virtual bool **setPropertyValue** (AttributionEvent *event, string value, Animation *anim)
- virtual bool **setPropertyValue** (string propName, vector< string > *params)
- virtual bool **setPropertyValue** (string propName, vector< double > *params)
- double **getPropertyValueAsDouble** (string param)
- string **getPropertyValueAsString** (string param)
- virtual bool **unprepare** ()
- void **setHandling** (bool **isHandling**)
- void **setHandler** (bool **isHandler**)
- void **selectionEvent** (int accessCode, double currentTime)
- set< int > * **getInputEvents** ()

Protected Member Functions

- bool **lock** ()
- bool **unlock** ()

Protected Attributes

- string **id**
- Node * **dataObject**
- CascadingDescriptor * **descriptor**
- double **offsetTime**
- double **startTime**
- PresentationEvent * **wholeContent**
- set< string > **typeSet**
- pthread_mutex_t **mutex**
- pthread_mutex_t **mutexEvent**
- pthread_mutex_t **mutexParentTable**
- bool **isLocked**
- bool **deleting**
- bool **isHandler**
- bool **isHandling**
- map< Node *, void * > * **parentTable**
- bool **visible**
- map< string, FormatterEvent * > * **events**
- vector< PresentationEvent * > * **presEvents**
- set< SelectionEvent * > * **selectionEvents**
- vector< FormatterEvent * > * **otherEvents**
- int **pauseCount**
- FormatterEvent * **mainEvent**
- EventTransitionManager * **transMan**

Private Member Functions

- void **initializeExecutionObject** (string **id**, Node ∗node, CascadingDescriptor ∗**descriptor**, bool handling)
- void **lockEvents** ()
- void **unlockEvents** ()
- void **lockParentTable** ()
- void **unlockParentTable** ()

Private Attributes

- map< Node ∗, Node ∗ > ∗ **nodeParentTable**
- bool **isItCompiled**

2.48.1 Detailed Description

Definition at line 117 of file ExecutionObject.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/ExecutionObject.h

2.49 NodeNesting Class Reference

Collaboration diagram for NodeNesting:

Public Member Functions

- **NodeNesting** (Node ∗node)
- **NodeNesting** (**NodeNesting** ∗seq)
- **NodeNesting** (vector< Node ∗ > ∗seq)
- bool **instanceOf** (string s)
- void **append** (**NodeNesting** ∗otherSeq)
- void **append** (vector< Node ∗ > ∗otherSeq)
- Node ∗ **getAnchorNode** ()
- Node ∗ **getHeadNode** ()
- Node ∗ **getNode** (int index)
- int **getNumNodes** ()
- void **insertAnchorNode** (Node ∗node)
- void **insertHeadNode** (Node ∗node)
- bool **removeAnchorNode** ()
- bool **removeHeadNode** ()
- bool **removeNode** (Node ∗node)
- string **getId** ()
- **NodeNesting** ∗ **copy** ()
- string **toString** ()

Private Member Functions

- void **initialize** ()

Private Attributes

- string **id**
- vector< Node ∗ > ∗ **nodes**
- set< string > **typeSet**
- pthread_mutex_t **mutexNodes**

2.49.1 Detailed Description

Definition at line 67 of file NodeNesting.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/NodeNesting.h

2.50 AnchorEvent Class Reference

Inheritance diagram for AnchorEvent:Collaboration diagram for AnchorEvent:

Public Member Functions

- **AnchorEvent** (string **id**, void ∗**executionObject**, ContentAnchor ∗**anchor**)
- ContentAnchor ∗ **getAnchor** ()
- virtual bool **start** ()
- virtual bool **stop** ()

Protected Attributes

- ContentAnchor ∗ **anchor**

2.50.1 Detailed Description

Definition at line 68 of file AnchorEvent.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/AnchorEvent.h

2.51 AttributionEvent Class Reference

Inheritance diagram for AttributionEvent:Collaboration diagram for AttributionEvent:

Public Member Functions

- **AttributionEvent** (string **id**, void ∗**executionObject**, PropertyAnchor ∗**anchor**)
- PropertyAnchor ∗ **getAnchor** ()
- string **getCurrentValue** ()
- bool **setValue** (string newValue)
- void **setValueMaintainer** (**IAttributeValueMaintainer** ∗**valueMaintainer**)
- void **setImplicitRefAssessmentEvent** (string roleId, **FormatterEvent** ∗event)
- **FormatterEvent** ∗ **getImplicitRefAssessmentEvent** (string roleId)

Protected Attributes

- PropertyAnchor ∗ **anchor**
- **IAttributeValueMaintainer** ∗ **valueMaintainer**
- map< string, **FormatterEvent** ∗ > ∗ **assessments**

Private Attributes

- bool **settingNode**

2.51.1 Detailed Description

Definition at line 70 of file AttributionEvent.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/AttributionEvent.h

2.52 FormatterEvent Class Reference

Inheritance diagram for FormatterEvent:Collaboration diagram for FormatterEvent:

Public Member Functions

- **FormatterEvent** (string **id**, void ∗**executionObject**)
- bool **instanceOf** (string s)
- void **setId** (string **id**)
- void **addEventListener** (**IEventListener** ∗listener)
- bool **containsEventListener** (**IEventListener** ∗listener)
- bool **abort** ()
- virtual bool **start** ()
- virtual bool **stop** ()
- bool **pause** ()
- bool **resume** ()
- void **setCurrentState** (short newState)
- void **clearEventListeners** ()
- short **getCurrentState** ()
- void ∗ **getExecutionObject** ()
- void **setExecutionObject** (void ∗object)
- string **getId** ()
- long **getOccurrences** ()
- bool **removeEventListener** (**IEventListener** ∗listener)

Static Public Member Functions

- static string **getStateName** (short state)

Protected Member Functions

- short **getNewState** (short transition)
- short **getTransition** (short newState)
- bool **changeState** (short newState, short transition)

Protected Attributes

- string **id**
- short **currentState**
- long **occurrences**
- void ∗ **executionObject**
- set< **IEventListener** ∗ > ∗ **coreListeners**
- set< **IEventListener** ∗ > ∗ **linksListeners**
- set< **IEventListener** ∗ > ∗ **objectsListeners**
- set< string > **typeSet**
- bool **deleting**
- pthread_mutex_t **mutex**

Static Private Attributes

- static const short **ST_ABORTED** = 50

2.52.1 Detailed Description

Definition at line 70 of file FormatterEvent.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/FormatterEvent.h

2.53 IAttributeValueMaintainer Class Reference

Public Member Functions

- virtual ~**IAttributeValueMaintainer** ()
- virtual string **getPropertyValue** (void *attributeEvent)=0

2.53.1 Detailed Description

Definition at line 63 of file IAttributeValueMaintainer.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/IAttributeValueMaintainer.h

2.54 IEventListener Class Reference

Collaboration diagram for IEventListener:

Public Member Functions

- virtual ~**IEventListener** ()
- virtual void **eventStateChanged** (void *event, short transition, short previousState)=0
- virtual short **getPriorityType** ()=0
- virtual void **stopListening** (void *event)=0

Static Public Attributes

- static const short **PT_CORE** = 0
- static const short **PT_LINK** = 1
- static const short **PT_OBJECT** = 2

2.54.1 Detailed Description

Definition at line 63 of file IEventListener.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/IEventListener.h

2.55 IFormatterEvent Class Reference

Inheritance diagram for IFormatterEvent:

Public Member Functions

- virtual ~**IFormatterEvent** ()
- virtual bool **instanceOf** (string s)=0
- virtual void **setId** (string id)=0
- virtual void **addEventListener** (**IEventListener** *listener)=0
- virtual bool **containsEventListener** (**IEventListener** *listener)=0
- virtual bool **abort** ()=0
- virtual bool **start** ()=0
- virtual bool **stop** ()=0
- virtual bool **pause** ()=0
- virtual bool **resume** ()=0
- virtual void **setCurrentState** (short newState)=0
- virtual void **clearEventListeners** ()=0
- virtual short **getCurrentState** ()=0
- virtual void * **getExecutionObject** ()=0
- virtual void **setExecutionObject** (void *object)=0
- virtual string **getId** ()=0
- virtual long **getOccurrences** ()=0
- virtual bool **removeEventListener** (**IEventListener** *listener)=0

2.55.1 Detailed Description

Definition at line 65 of file IFormatterEvent.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/IFormatterEvent.h

2.56 PresentationEvent Class Reference

Inheritance diagram for PresentationEvent:Collaboration diagram for PresentationEvent:

Public Member Functions

- **PresentationEvent** (string **id**, void ∗**executionObject**, ContentAnchor ∗**anchor**)
- virtual ∼**PresentationEvent** ()
- bool **stop** ()
- double **getDuration** ()
- double **getRepetitionInterval** ()
- long **getRepetitions** ()
- void **setDuration** (double dur)
- void **setEnd** (double e)
- void **setRepetitionSettings** (long repetitions, double **repetitionInterval**)
- double **getBegin** ()
- double **getEnd** ()
- void **incrementOccurrences** ()

Static Public Member Functions

- static bool **isUndefinedInstant** (double value)

Static Public Attributes

- static const double **UNDEFINED_INSTANT**

Private Attributes

- double **begin**
- double **end**
- double **duration**
- long **numPresentations**
- double **repetitionInterval**

2.56.1 Detailed Description

Definition at line 72 of file PresentationEvent.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/PresentationEvent.h

2.57 SelectionEvent Class Reference

Inheritance diagram for SelectionEvent:Collaboration diagram for SelectionEvent:

Public Member Functions

- **SelectionEvent** (string **id**, void ∗**executionObject**, ContentAnchor ∗**anchor**)
- bool **start** ()
- int **getSelectionCode** ()
- void **setSelectionCode** (string codeStr)

Private Attributes

- int **selectionCode**

2.57.1 Detailed Description

Definition at line 74 of file SelectionEvent.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/SelectionEvent.h

2.58 BeginEventTransition Class Reference

Inheritance diagram for BeginEventTransition:Collaboration diagram for BeginEventTransition:

Public Member Functions

- **BeginEventTransition** (double **time**, **PresentationEvent** ∗**event**)
- **EndEventTransition** ∗ **getEndTransition** ()
- void **setEndTransition** (**EndEventTransition** ∗entry)

Private Attributes

- **EndEventTransition** ∗ **endTransition**

2.58.1 Detailed Description

Definition at line 67 of file BeginEventTransition.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/BeginEventTransition.h

2.59 EndEventTransition Class Reference

Inheritance diagram for EndEventTransition:Collaboration diagram for EndEventTransition:

Public Member Functions

- **EndEventTransition** (double **time**, **PresentationEvent** ∗**event**, void ∗transition)
- void ∗ **getBeginTransition** ()

Private Attributes

- void ∗ **beginTransition**

2.59.1 Detailed Description

Definition at line 66 of file EndEventTransition.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/EndEventTransition.h

2.60 EventTransition Class Reference

Inheritance diagram for EventTransition:Collaboration diagram for EventTransition:

Public Member Functions

- **EventTransition** (double **time**, **PresentationEvent** ∗**event**)
- int **compareTo** (**EventTransition** ∗object)
- bool **equals** (**EventTransition** ∗object)
- **PresentationEvent** ∗ **getEvent** ()
- double **getTime** ()
- bool **instanceOf** (string s)

Protected Attributes

- set< string > **typeSet**

Private Member Functions

- int **compareType** (**EventTransition** ∗otherEntry)

Private Attributes

- **PresentationEvent** ∗ **event**
- double **time**

2.60.1 Detailed Description

Definition at line 68 of file EventTransition.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/EventTransition.h

2.61 EventTransitionManager Class Reference

Collaboration diagram for EventTransitionManager:

Public Member Functions

- void **addPresentationEvent** (**PresentationEvent** ∗event)
- void **removeEventTransition** (**PresentationEvent** ∗event)
- void **resetTimeIndex** ()
- void **resetTimeIndexByType** (short int type)
- void **prepare** (bool wholeContent, double startTime, short int type)
- void **start** (double offsetTime)
- void **stop** (double endTime)
- void **abort** (double endTime)
- void **updateTransitionTable** (double timeValue, IPlayer ∗player, **FormatterEvent** ∗mainEvent, short int transType)
- set< double > ∗ **getTransitionsValues** (short int transType)
- **EventTransition** ∗ **getNextTransition** (**FormatterEvent** ∗mainEvent)

Private Member Functions

- short int **getType** (**PresentationEvent** ∗event)
- vector< **EventTransition** ∗ > ∗ **getTransitionEvents** (short int type)
- void **addEventTransition** (**EventTransition** ∗transition, short int type)

Private Attributes

- map< short int, int > ∗ **currentTransitionIndex**
- map< short int, int > ∗ **startTransitionIndex**
- map< short int, vector< **EventTransition** ∗ > ∗ > ∗ **transTable**
- pthread_mutex_t **transMutex**

2.61.1 Detailed Description

Definition at line 81 of file EventTransitionManager.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/EventTransitionManager.h

2.62 TransitionDispatcher Class Reference

Inheritance diagram for TransitionDispatcher:Collaboration diagram for TransitionDispatcher:

Public Member Functions

- **TransitionDispatcher** (**EventTransition** ∗**transition**)
- virtual void **run** ()

Private Attributes

- **EventTransition** ∗ **transition**

2.62.1 Detailed Description

Definition at line 67 of file TransitionDispatcher.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/TransitionDispatcher.h

2.63 FormatterCausalLink Class Reference

Inheritance diagram for FormatterCausalLink:Collaboration diagram for FormatterCausalLink:

Public Member Functions

- **FormatterCausalLink** (**LinkTriggerCondition** ∗**condition**, **LinkAction** ∗**action**, Link ∗**ncmLink**, void ∗**parentObject**)
- **LinkAction** ∗ **getAction** ()
- **LinkTriggerCondition** ∗ **getTriggerCondition** ()
- void **conditionSatisfied** (void ∗**condition**)
- virtual vector< FormatterEvent ∗ > ∗ **getEvents** ()
- void **evaluationStarted** ()
- void **evaluationEnded** ()
- void **actionProcessed** (bool start)

Private Attributes

- **LinkTriggerCondition** ∗ **condition**
- **LinkAction** ∗ **action**

2.63.1 Detailed Description

Definition at line 76 of file FormatterCausalLink.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/FormatterCausalLink.h

2.64 FormatterLink Class Reference

Inheritance diagram for FormatterLink:Collaboration diagram for FormatterLink:

Public Member Functions

- **FormatterLink** (Link ∗**ncmLink**, void ∗**parentObject**)
- Link ∗ **getNcmLink** ()
- bool **instanceOf** (string s)

Protected Attributes

- Link ∗ **ncmLink**
- void ∗ **parentObject**
- set< string > **typeSet**

2.64.1 Detailed Description

Definition at line 68 of file FormatterLink.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/FormatterLink.h

2.65 LinkAction Class Reference

Inheritance diagram for LinkAction:Collaboration diagram for LinkAction:

Public Member Functions

- **LinkAction** (double **delay**)
- bool **instanceOf** (string s)
- double **getWaitDelay** ()
- void **setWaitDelay** (double **delay**)
- bool **hasDelay** ()
- void **addActionProgressionListener** (**LinkActionProgressionListener** ∗listener)
- void **removeActionProgressionListener** (**LinkActionProgressionListener** ∗listener)
- void **notifyProgressionListeners** (bool start)
- virtual vector< FormatterEvent ∗ > ∗ **getEvents** ()=0
- virtual vector< **LinkAction** ∗ > ∗ **getImplicitRefRoleActions** ()=0
- virtual void **run** ()

Protected Attributes

- set< string > **typeSet**

Private Attributes

- double **delay**
- vector< **LinkActionProgressionListener** ∗ > ∗ **progressionListeners**

2.65.1 Detailed Description

Definition at line 76 of file LinkAction.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkAction.h

2.66 LinkActionListener Class Reference

Public Member Functions

- virtual ~**LinkActionListener** ()
- virtual void **runAction** (void *action)=0

2.66.1 Detailed Description

Definition at line 60 of file LinkActionListener.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkActionListener.h

2.67 LinkActionProgressionListener Class Reference

Inheritance diagram for LinkActionProgressionListener:

Public Member Functions

- virtual ~**LinkActionProgressionListener** ()
- virtual void **actionProcessed** (bool start)=0

2.67.1 Detailed Description

Definition at line 60 of file LinkActionProgressionListener.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkActionProgressionListener.h

2.68 LinkAndCompoundTriggerCondition Class Reference

Inheritance diagram for LinkAndCompoundTriggerCondition:Collaboration diagram for LinkAndCompoundTriggerCondition:

Public Member Functions

- void **addCondition** (**LinkCondition** ∗condition)
- void **conditionSatisfied** (void ∗condition)
- vector< FormatterEvent ∗ > ∗ **getEvents** ()

Private Attributes

- vector< **LinkCondition** ∗ > ∗ **unsatisfiedConditions**
- vector< **LinkCondition** ∗ > ∗ **statements**

2.68.1 Detailed Description

Definition at line 68 of file LinkAndCompoundTriggerCondition.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkAndCompoundTriggerCondition.h

2.69 LinkAssessment Class Reference

Inheritance diagram for LinkAssessment:Collaboration diagram for LinkAssessment:

Public Member Functions

- **LinkAssessment** ()
- virtual ~**LinkAssessment** ()
- virtual string **getValue** ()=0
- bool **instanceOf** (string type)

Protected Attributes

- set< string > **typeSet**

2.69.1 Detailed Description

Definition at line 64 of file LinkAssessment.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkAssessment.h

2.70 LinkAssessmentStatement Class Reference

Inheritance diagram for LinkAssessmentStatement:Collaboration diagram for LinkAssessmentStatement:

Public Member Functions

- **LinkAssessmentStatement** (short **comparator**, **LinkAttributeAssessment** ∗**mainAssessment**, **LinkAssessment** ∗**otherAssessment**)
- vector< FormatterEvent ∗ > ∗ **getEvents** ()
- short **getComparator** ()
- void **setComparator** (short comp)
- **LinkAssessment** ∗ **getMainAssessment** ()
- void **setMainAssessment** (**LinkAssessment** ∗assessment)
- **LinkAssessment** ∗ **getOtherAssessment** ()
- void **setOtherAssessment** (**LinkAssessment** ∗assessment)
- virtual bool **evaluate** ()

Protected Attributes

- short **comparator**
- **LinkAssessment** ∗ **otherAssessment**

Private Attributes

- **LinkAssessment** ∗ **mainAssessment**

2.70.1 Detailed Description

Definition at line 74 of file LinkAssessmentStatement.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkAssessmentStatement.h

2.71 LinkAssignmentAction Class Reference

Inheritance diagram for LinkAssignmentAction:Collaboration diagram for LinkAssignmentAction:

Public Member Functions

- **LinkAssignmentAction** (FormatterEvent ∗**event**, short **actionType**, string **value**)
- string **getValue** ()
- void **setValue** (string **value**)
- Animation ∗ **getAnimation** ()
- void **setAnimation** (Animation ∗**animation**)

Private Attributes

- string **value**
- Animation ∗ **animation**

2.71.1 Detailed Description

Definition at line 69 of file LinkAssignmentAction.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkAssignmentAction.h

2.72 LinkAttributeAssessment Class Reference

Inheritance diagram for LinkAttributeAssessment:Collaboration diagram for LinkAttributeAssessment:

Public Member Functions

- **LinkAttributeAssessment** (FormatterEvent ∗ev, short attrType)
- FormatterEvent ∗ **getEvent** ()
- void **setOffset** (string **offset**)
- string **getOffset** ()
- void **setEvent** (FormatterEvent ∗ev)
- short **getAttributeType** ()
- void **setAttributeType** (short attrType)
- string **getValue** ()

Private Member Functions

- string **getAssessmentWithOffset** (string assessmentValue)

Private Attributes

- FormatterEvent ∗ **event**
- short **attributeType**
- string **offset**

2.72.1 Detailed Description

Definition at line 74 of file LinkAttributeAssessment.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkAttributeAssessment.h

2.73 LinkCompoundAction Class Reference

Inheritance diagram for LinkCompoundAction:Collaboration diagram for LinkCompoundAction:

Public Member Functions

- **LinkCompoundAction** (short **op**)
- short **getOperator** ()
- void **addAction** (**LinkAction** ∗action)
- vector< **LinkAction** ∗ > ∗ **getActions** ()
- virtual vector< FormatterEvent ∗ > ∗ **getEvents** ()
- void **actionProcessed** (bool start)
- virtual vector< **LinkAction** ∗ > ∗ **getImplicitRefRoleActions** ()

Protected Attributes

- vector< **LinkAction** ∗ > ∗ **actions**
- short **op**

Private Member Functions

- virtual void **run** ()

Private Attributes

- int **pendingActions**
- bool **hasStart**
- bool **runing**

2.73.1 Detailed Description

Definition at line 72 of file LinkCompoundAction.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkCompoundAction.h

2.74 LinkCompoundStatement Class Reference

Inheritance diagram for LinkCompoundStatement:Collaboration diagram for LinkCompoundStatement:

Public Member Functions

- **LinkCompoundStatement** (short **op**)
- short **getOperator** ()
- void **addStatement** (**LinkStatement** ∗statement)
- vector< **LinkStatement** ∗ > ∗ **getStatements** ()
- bool **isNegated** ()
- void **setNegated** (bool neg)
- virtual vector< FormatterEvent ∗ > ∗ **getEvents** ()
- virtual bool **evaluate** ()

Protected Member Functions

- bool **returnEvaluationResult** (bool result)

Protected Attributes

- vector< **LinkStatement** ∗ > ∗ **statements**
- bool **negated**
- short **op**

2.74.1 Detailed Description

Definition at line 71 of file LinkCompoundStatement.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkCompoundStatement.h

2.75 LinkCompoundTriggerCondition Class Reference

Inheritance diagram for LinkCompoundTriggerCondition:Collaboration diagram for LinkCompoundTriggerCondition:

Public Member Functions

- virtual void **addCondition** (**LinkCondition** ∗condition)
- virtual void **conditionSatisfied** (void ∗condition)
- vector< **LinkCondition** ∗ > ∗ **getConditions** ()
- virtual vector< FormatterEvent ∗ > ∗ **getEvents** ()
- void **evaluationStarted** ()
- void **evaluationEnded** ()

Protected Attributes

- vector< **LinkCondition** ∗ > ∗ **conditions**

2.75.1 Detailed Description

Definition at line 71 of file LinkCompoundTriggerCondition.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkCompoundTriggerCondition.h

2.76 LinkCondition Class Reference

Inheritance diagram for LinkCondition:Collaboration diagram for LinkCondition:

Public Member Functions

- **LinkCondition** ()
- virtual ∼**LinkCondition** ()
- bool **instanceOf** (string s)
- virtual vector< FormatterEvent ∗ > ∗ **getEvents** ()=0

Protected Attributes

- set< string > **typeSet**

2.76.1 Detailed Description

Definition at line 69 of file LinkCondition.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkCondition.h

2.77 LinkListener Class Reference

Public Member Functions

- virtual ∼**LinkListener** ()
- virtual void **linkEvaluationStarted** (**FormatterCausalLink** ∗link)=0
- virtual void **linkEvaluationFinished** (**FormatterCausalLink** ∗link, bool start)=0

2.77.1 Detailed Description

Definition at line 60 of file LinkListener.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkListener.h

2.78 LinkRepeatAction Class Reference

Inheritance diagram for LinkRepeatAction:Collaboration diagram for LinkRepeatAction:

Public Member Functions

- **LinkRepeatAction** (FormatterEvent ∗**event**, short **actionType**)
- virtual ∼**LinkRepeatAction** ()
- long **getRepetitions** ()
- double **getRepetitionInterval** ()
- void **setRepetitions** (long **repetitions**)
- void **setRepetitionInterval** (double **delay**)

Private Member Functions

- virtual void **run** ()

Private Attributes

- long **repetitions**
- double **repetitionInterval**

2.78.1 Detailed Description

Definition at line 70 of file LinkRepeatAction.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkRepeatAction.h

2.79 LinkSimpleAction Class Reference

Inheritance diagram for LinkSimpleAction:Collaboration diagram for LinkSimpleAction:

Public Member Functions

- **LinkSimpleAction** (FormatterEvent ∗**event**, short type)
- FormatterEvent ∗ **getEvent** ()
- short **getType** ()
- void **setActionListener** (**LinkActionListener** ∗**listener**)
- virtual vector< FormatterEvent ∗ > ∗ **getEvents** ()
- virtual vector< **LinkAction** ∗ > ∗ **getImplicitRefRoleActions** ()

Protected Member Functions

- virtual void **run** ()

Protected Attributes

- FormatterEvent ∗ **event**
- short **actionType**

Private Attributes

- **LinkActionListener** ∗ **listener**

2.79.1 Detailed Description

Definition at line 73 of file LinkSimpleAction.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkSimpleAction.h

2.80 LinkStatement Class Reference

Inheritance diagram for LinkStatement:Collaboration diagram for LinkStatement:

Public Member Functions

- **LinkStatement** ()
- virtual ~**LinkStatement** ()
- virtual bool **evaluate** ()=0

2.80.1 Detailed Description

Definition at line 62 of file LinkStatement.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkStatement.h

2.81 LinkTransitionTriggerCondition Class Reference

Inheritance diagram for LinkTransitionTriggerCondition:Collaboration diagram for LinkTransitionTriggerCondition:

Public Member Functions

- **LinkTransitionTriggerCondition** (FormatterEvent ∗**event**, short **transition**)
- void **stopListening** (void ∗**event**)
- virtual void **eventStateChanged** (void ∗**event**, short **transition**, short previousState)
- short **getPriorityType** ()
- FormatterEvent ∗ **getEvent** ()
- short **getTransition** ()
- virtual vector< FormatterEvent ∗ > ∗ **getEvents** ()

Protected Attributes

- FormatterEvent ∗ **event**
- short **transition**

2.81.1 Detailed Description

Definition at line 70 of file LinkTransitionTriggerCondition.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkTransitionTriggerCondition.h

2.82 LinkTriggerCondition Class Reference

Inheritance diagram for LinkTriggerCondition:Collaboration diagram for LinkTriggerCondition:

Public Member Functions

- void **setTriggerListener** (**LinkTriggerListener** ∗**listener**)
- **LinkTriggerListener** ∗ **getTriggerListener** ()
- double **getDelay** ()
- void **setDelay** (double **delay**)
- virtual void **conditionSatisfied** (void ∗condition)
- void **run** ()
- virtual vector< FormatterEvent ∗ > ∗ **getEvents** ()=0

Protected Member Functions

- virtual void **notifyConditionObservers** (short satus)

Protected Attributes

- **LinkTriggerListener** ∗ **listener**
- double **delay**

2.82.1 Detailed Description

Definition at line 66 of file LinkTriggerCondition.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkTriggerCondition.h

2.83 LinkTriggerListener Class Reference

Inheritance diagram for LinkTriggerListener:Collaboration diagram for LinkTriggerListener:

Public Member Functions

- virtual ~**LinkTriggerListener** ()
- virtual void **conditionSatisfied** (void *condition)=0
- virtual void **evaluationStarted** ()=0
- virtual void **evaluationEnded** ()=0

Static Public Attributes

- static const short **CONDITION_SATISFIED** = 0
- static const short **EVALUATION_STARTED** = 1
- static const short **EVALUATION_ENDED** = 2

2.83.1 Detailed Description

Definition at line 60 of file LinkTriggerListener.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkTriggerListener.h

2.84 LinkValueAssessment Class Reference

Inheritance diagram for LinkValueAssessment:Collaboration diagram for LinkValueAssessment:

Public Member Functions

- **LinkValueAssessment** (string **value**)
- string **getComparableValue** ()
- void **setComparableValue** (string **value**)
- string **getValue** ()

Protected Attributes

- string **value**

2.84.1 Detailed Description

Definition at line 66 of file LinkValueAssessment.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinkValueAssessment.h

2.85 CascadingDescriptor Class Reference

Collaboration diagram for CascadingDescriptor:

Public Member Functions

- **CascadingDescriptor** (GenericDescriptor ∗firstDescriptor)
- **CascadingDescriptor** (**CascadingDescriptor** ∗descriptor)
- bool **instanceOf** (string s)
- string **getId** ()
- bool **isLastDescriptor** (GenericDescriptor ∗descriptor)
- void **cascade** (GenericDescriptor ∗preferredDescriptor)
- GenericDescriptor ∗ **getFirstUnsolvedDescriptor** ()
- vector< GenericDescriptor ∗ > ∗ **getUnsolvedDescriptors** ()
- void **cascadeUnsolvedDescriptor** ()
- double **getExplicitDuration** ()
- bool **getFreeze** ()
- string **getPlayerName** ()
- LayoutRegion ∗ **getRegion** ()
- **FormatterRegion** ∗ **getFormatterRegion** ()
- void **setFormatterRegion** (**FormatterRegion** ∗**region**)
- void **setFormatterLayout** (void ∗formatterLayout)
- long **getRepetitions** ()
- vector< Parameter ∗ > ∗ **getParameters** ()
- string **getParameterValue** (string paramName)
- vector< GenericDescriptor ∗ > ∗ **getNcmDescriptors** ()
- Color ∗ **getFocusBorderColor** ()
- double **getFocusBorderTransparency** ()
- int **getFocusBorderWidth** ()
- string **getFocusIndex** ()
- string **getFocusSrc** ()
- string **getSelectionSrc** ()
- string **getMoveDown** ()
- string **getMoveLeft** ()
- string **getMoveRight** ()
- string **getMoveUp** ()
- Color ∗ **getSelBorderColor** ()
- vector< Transition ∗ > ∗ **getInputTransitions** ()
- vector< Transition ∗ > ∗ **getOutputTransitions** ()

Protected Member Functions

- void **initializeCascadingDescriptor** ()

Protected Attributes

- set< string > **typeSet**

Private Member Functions

- void **cascadeDescriptor** (Descriptor ∗descriptor)

Private Attributes

- string **id**
- vector< GenericDescriptor ∗ > ∗ **descriptors**
- vector< GenericDescriptor ∗ > ∗ **unsolvedDescriptors**
- map< string, string > ∗ **parameters**
- double **explicitDuration**
- string **playerName**
- long **repetitions**
- bool **freeze**
- LayoutRegion ∗ **region**
- **FormatterRegion** ∗ **formatterRegion**
- string **focusIndex**
- string **moveUp**
- string **moveDown**
- string **moveLeft**
- string **moveRight**
- string **focusSrc**
- string **selectionSrc**
- Color ∗ **focusBorderColor**
- Color ∗ **selBorderColor**
- int **focusBorderWidth**
- double **focusBorderTransparency**
- vector< Transition ∗ > ∗ **inputTransitions**
- vector< Transition ∗ > ∗ **outputTransitions**

2.85.1 Detailed Description

Definition at line 92 of file CascadingDescriptor.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/CascadingDescriptor.h

2.86 FocusSourceManager Class Reference

Static Public Member Functions

- static ISurface * **getComponent** (string src)

2.86.1 Detailed Description

Definition at line 67 of file FocusSourceManager.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/FocusSourceManager.h

2.87 FormatterDeviceRegion Class Reference

Inheritance diagram for FormatterDeviceRegion:Collaboration diagram for FormatterDeviceRegion:

Public Member Functions

- **FormatterDeviceRegion** (string **id**)
- virtual ∼**FormatterDeviceRegion** ()
- void **addRegion** (LayoutRegion ∗region)
- LayoutRegion ∗ **cloneRegion** ()
- int **compareWidthSize** (string w)
- int **compareHeightSize** (string h)
- short **getBackgroundColor** ()
- double **getBottom** ()
- double **getHeight** ()
- double **getLeft** ()
- double **getRight** ()
- LayoutRegion ∗ **getRegion** (string **id**)
- LayoutRegion ∗ **getRegionRecursively** (string **id**)
- vector< LayoutRegion ∗ > ∗ **getRegions** ()
- string **getTitle** ()
- double **getTop** ()
- double **getWidth** ()
- int **getZIndex** ()
- int **getZIndexValue** ()
- bool **isBottomPercentual** ()
- bool **isHeightPercentual** ()
- bool **isLeftPercentual** ()
- bool **isRightPercentual** ()
- bool **isTopPercentual** ()
- bool **isWidthPercentual** ()
- string **toString** ()
- bool **removeRegion** (LayoutRegion ∗region)
- void **removeRegions** ()
- void **setBackgroundColor** (Color ∗newBackgroundColor)
- bool **setBottom** (double newBottom, bool isPercentual)
- bool **setHeight** (double newHeight, bool isPercentual)
- bool **setLeft** (double newLeft, bool isPercentual)
- bool **setRight** (double newRight, bool isPercentual)
- void **setTitle** (string newTitle)
- bool **setTop** (double newTop, bool isPercentual)
- bool **setWidth** (double newWidth, bool isPercentual)
- void **setZIndex** (int newZIndex)
- vector< LayoutRegion ∗ > ∗ **getRegionsSortedByZIndex** ()
- vector< LayoutRegion ∗ > ∗ **getRegionsOverRegion** (LayoutRegion ∗region)
- LayoutRegion ∗ **getParent** ()
- void **setParent** (LayoutRegion ∗parent)
- int **getTopInPixels** ()
- int **getBottomInPixels** ()

- int **getRightInPixels** ()
- int **getLeftInPixels** ()
- int **getHeightInPixels** ()
- int **getWidthInPixels** ()
- bool **isMovable** ()
- bool **isResizable** ()
- bool **isDecorated** ()
- void **setMovable** (bool movable)
- void **setResizable** (bool resizable)
- void **setDecorated** (bool decorated)
- void **resetTop** ()
- void **resetBottom** ()
- void **resetLeft** ()
- void **resetHeight** ()
- void **resetWidth** ()
- void **resetZIndex** ()
- void **resetDecorated** ()
- void **resetMovable** ()
- void **resetResizable** ()
- int **getAbsoluteLeft** ()
- int **getAbsoluteTop** ()
- void **dispose** ()
- string **getId** ()
- void **setId** (string **id**)
- Entity ∗ **getDataEntity** ()
- int **compareTo** (void ∗arg0)

Private Attributes

- string **id**
- int **top**
- int **left**
- int **width**
- int **height**
- vector< LayoutRegion ∗ > ∗ **regions**
- set< LayoutRegion ∗ > **regionSet**

2.87.1 Detailed Description

Definition at line 73 of file FormatterDeviceRegion.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/FormatterDeviceRegion.h

2.88 FormatterLayout Class Reference

Collaboration diagram for FormatterLayout:

Public Member Functions

- **FormatterLayout** (int x, int y, int w, int h)
- void **addChild** (string objectId, **FormatterLayout** ∗child)
- **FormatterLayout** ∗ **getChild** (string objectId)
- string **getBitMapRegionId** ()
- LayoutRegion ∗ **getNcmRegion** (string regionId)
- bool **getScreenShot** (IWindow ∗region)
- ExecutionObject ∗ **getObject** (int x, int y)
- void **prepareFormatterRegion** (ExecutionObject ∗object, ISurface ∗renderedSurface)
- void **showObject** (ExecutionObject ∗object)
- void **hideObject** (ExecutionObject ∗object)
- set< **FormatterRegion** ∗ > ∗ **getFormatterRegionsFromNcmRegion** (string regionId)
- void **clear** ()

Protected Attributes

- set< string > **typeSet**

Private Member Functions

- void **printRegionMap** ()
- void **printObjectMap** ()
- void **createDeviceRegion** (int x, int y, int w, int h)
- void **sortRegion** (string regionId, int zIndex)
- void **lock** ()
- void **unlock** ()

Private Attributes

- map< string, **FormatterLayout** ∗ > ∗ **childs**
- LayoutRegion ∗ **deviceRegion**
- vector< string > ∗ **sortedRegions**
- map< string, int > ∗ **regionZIndex**
- map< string, set< **FormatterRegion** ∗ > ∗ > ∗ **regionMap**
- map< **FormatterRegion** ∗, ExecutionObject ∗ > ∗ **objectMap**
- pthread_mutex_t **mutex**

2.88.1 Detailed Description

Definition at line 78 of file FormatterLayout.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/FormatterLayout.h

2.89 FormatterRegion Class Reference

Collaboration diagram for FormatterRegion:

Public Member Functions

- **FormatterRegion** (string **objectId**, void ∗**descriptor**, void ∗**layoutManager**)
- void ∗ **getLayoutManager** ()
- void **updateRegionBounds** ()
- bool **intersects** (int x, int y)
- LayoutRegion ∗ **getLayoutRegion** ()
- LayoutRegion ∗ **getOriginalRegion** ()
- io::IWindow ∗ **getODContentPane** ()
- void **prepareOutputDisplay** (io::ISurface ∗renderedSurface)
- void **renderSurface** (io::ISurface ∗renderedSurface)
- void **showContent** ()
- void **hideContent** ()
- void **setRegionVisibility** (bool visible)
- void **disposeOutputDisplay** ()
- void **toFront** ()
- void **windowGainedFocus** ()
- io::IWindow ∗ **getRegionRectangle** ()
- void **setGhostRegion** (bool ghost)
- bool **isVisible** ()
- short **getFocusState** ()
- bool **setSelection** (bool selOn)
- void **setFocus** (bool focusOn)
- void **unselect** ()
- void **setFocusInfo** (Color ∗**focusBorderColor**, int **focusBorderWidth**, string **focusComponentSrc**, Color ∗**selBorderColor**, int **selBorderWidth**, string **selComponentSrc**)
- string **getFocusIndex** ()
- Color ∗ **getBackgroundColor** ()
- float **getTransparency** ()
- void **setTransparency** (string strTrans)
- void **setTransparency** (float **transparency**)
- void **setBackgroundColor** (string color)
- void **setBackgroundColor** (Color ∗color)
- void **setChromaKey** (string value)
- void **setRgbChromaKey** (string value)
- void **setFit** (string value)
- void **setFit** (short value)
- void **setScroll** (string value)
- void **setScroll** (short value)

Static Public Attributes

- static const short **UNSELECTED** = 0
- static const short **FOCUSED** = 1
- static const short **SELECTED** = 2

Private Member Functions

- void **meetComponent** (int width, int height, int prefWidth, int prefHeight, io::ISurface ∗component)
- void **sliceComponent** (int width, int height, int prefWidth, int prefHeight, io::ISurface ∗component)
- void **updateCurrentComponentSize** ()
- void **updateCurrentComponentLocation** ()
- void **sizeRegion** ()
- void **bringChildrenToFront** (LayoutRegion ∗parentRegion)
- void **traverseFormatterRegions** (LayoutRegion ∗region, LayoutRegion ∗baseRegion)
- void **bringHideWindowToFront** (LayoutRegion ∗baseRegion, LayoutRegion ∗hideRegion)
- void **bringSiblingToFront** (**FormatterRegion** ∗region)
- void **barWipe** (Transition ∗transition, bool isShowEffect)
- void **fade** (Transition ∗transition, bool isShowEffect)
- void **lock** ()
- void **unlock** ()
- void **lockTransition** ()
- void **unlockTransition** ()
- void **lockFocusInfo** ()
- void **unlockFocusInfo** ()

Static Private Member Functions

- static void ∗ **barWipeT** (void ∗ptr)
- static void ∗ **fadeT** (void ∗ptr)

Private Attributes

- void ∗ **layoutManager**
- void ∗ **descriptor**
- string **objectId**
- LayoutRegion ∗ **ncmRegion**
- LayoutRegion ∗ **originalRegion**
- io::IWindow ∗ **outputDisplay**
- short **focusState**
- Color ∗ **focusBorderColor**
- int **focusBorderWidth**
- string **focusComponentSrc**
- Color ∗ **selBorderColor**
- string **selComponentSrc**
- int **selBorderWidth**
- Color ∗ **bgColor**
- float **transparency**
- short **fit**
- short **scroll**
- Color ∗ **chromaKey**
- string **transitionIn**
- string **transitionOut**
- bool **abortTransitionIn**
- bool **abortTransitionOut**
- pthread_mutex_t **mutex**
- pthread_mutex_t **mutexT**
- pthread_mutex_t **mutexFI**

2.89.1 Detailed Description

Definition at line 86 of file FormatterRegion.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/FormatterRegion.h

2.90 ExecutionObjectSwitch Class Reference

Inheritance diagram for ExecutionObjectSwitch:Collaboration diagram for ExecutionObjectSwitch:

Public Member Functions

- **ExecutionObjectSwitch** (string id, Node ∗switchNode, bool handling)
- virtual ∼**ExecutionObjectSwitch** ()
- ExecutionObject ∗ **getSelectedObject** ()
- void **select** (ExecutionObject ∗executionObject)
- bool **addEvent** (FormatterEvent ∗event)

Protected Attributes

- ExecutionObject ∗ **selectedObject**

2.90.1 Detailed Description

Definition at line 79 of file ExecutionObjectSwitch.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/ExecutionObjectSwitch.h

2.91 SwitchEvent Class Reference

Inheritance diagram for SwitchEvent:Collaboration diagram for SwitchEvent:

Public Member Functions

- **SwitchEvent** (string id, void ∗executionObjectSwitch, InterfacePoint ∗**interfacePoint**, int **eventType**, string **key**)
- void **stopListening** (void ∗event)
- InterfacePoint ∗ **getInterfacePoint** ()
- int **getEventType** ()
- string **getKey** ()
- void **setMappedEvent** (FormatterEvent ∗event)
- FormatterEvent ∗ **getMappedEvent** ()
- virtual void **eventStateChanged** (void ∗event, short transition, short previousState)
- short **getPriorityType** ()

Private Attributes

- InterfacePoint ∗ **interfacePoint**
- int **eventType**
- string **key**
- FormatterEvent ∗ **mappedEvent**

2.91.1 Detailed Description

Definition at line 70 of file SwitchEvent.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/SwitchEvent.h

2.92 CostFunctionDuration Class Reference

Inheritance diagram for CostFunctionDuration:Collaboration diagram for CostFunctionDuration:

Public Member Functions

- **CostFunctionDuration** (double **expectedValue**, double minValue, double maxValue, TemporalFlexibilityFunction ∗function)
- **CostFunctionDuration** (double **expectedValue**, TemporalFlexibilityFunction ∗function)
- virtual ∼**CostFunctionDuration** ()
- TemporalFlexibilityFunction ∗ **getCostFunction** ()
- void **setCostFunction** (TemporalFlexibilityFunction ∗function)
- virtual double **getCostValue** (double value)

Protected Member Functions

- void **overwrite** (**CostFunctionDuration** ∗dur)

Protected Attributes

- TemporalFlexibilityFunction ∗ **costFunction**

Private Member Functions

- void **updateDurationInterval** ()

2.92.1 Detailed Description

Definition at line 68 of file CostFunctionDuration.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/CostFunctionDuration.h

2.93 DurationFactory Class Reference

Static Public Member Functions

- static **TimeMeasurement** ∗ **createDuration** (double duration, TemporalFlexibilityFunction ∗flexFunction)

2.93.1 Detailed Description

Definition at line 68 of file DurationFactory.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/DurationFactory.h

2.94 FlexibleTimeMeasurement Class Reference

Inheritance diagram for FlexibleTimeMeasurement:Collaboration diagram for FlexibleTimeMeasurement:

Public Member Functions

- **FlexibleTimeMeasurement** (double **expectedValue**, double minValue, double maxValue)
- virtual ~**FlexibleTimeMeasurement** ()
- double **getComputedValue** ()
- double **getOptimumValue** ()
- void **setComputedValue** (double value)
- void **setOptimumValue** (double optValue)
- double **getMaximumValue** ()
- double **getMaximumFeasibleValue** ()
- double **getMinimumFeasibleValue** ()
- double **getMinimumValue** ()
- bool **isPredictable** ()
- bool **isComputed** ()
- bool **isUpperBounded** ()
- **TimeMeasurement** ∗ **duplicate** ()
- string **toString** ()
- double **getValue** ()
- void **setMinimumValue** (double value)
- void **setMaximumValue** (double value)
- void **setMinimumFeasibleValue** (double value)
- void **setMaximumFeasibleValue** (double value)

Protected Member Functions

- virtual void **overwrite** (**FlexibleTimeMeasurement** ∗dur)

Protected Attributes

- double **computedValue**
- double **minimumValue**
- double **maximumValue**
- double **minFeasibleValue**
- double **maxFeasibleValue**

Private Member Functions

- void **setBoundaryValues** (double minValue, double maxValue)

2.94.1 Detailed Description

Definition at line 65 of file FlexibleTimeMeasurement.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/FlexibleTimeMeasurement.h

2.95 LinearCostFunctionDuration Class Reference

Inheritance diagram for LinearCostFunctionDuration:Collaboration diagram for LinearCostFunctionDuration:

Public Member Functions

- **LinearCostFunctionDuration** (double expectedVal, double minVal, double maxVal, LinearTimeCostFunction ∗function)
- **LinearCostFunctionDuration** (double expectedVal, LinearTimeCostFunction ∗function)
- virtual ∼**LinearCostFunctionDuration** ()
- double **getShrinkingCostRate** ()
- double **getStretchingCostRate** ()
- double **getCostValue** (double value)
- **TimeMeasurement** ∗ **duplicate** ()
- string **toString** ()

2.95.1 Detailed Description

Definition at line 69 of file LinearCostFunctionDuration.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/LinearCostFunctionDuration.h

2.96 TimeMeasurement Class Reference

Inheritance diagram for TimeMeasurement:Collaboration diagram for TimeMeasurement:

Public Member Functions

- **TimeMeasurement** (double value)
- virtual ∼**TimeMeasurement** ()
- double **getExpectedValue** ()
- virtual double **getComputedValue** ()
- virtual void **setComputedValue** (double value)
- double **getActualValue** ()
- void **setExpectedValue** (double value)
- void **setActualValue** (double value)
- virtual bool **isPredictable** ()
- virtual bool **isComputed** ()
- string **toString** ()
- virtual **TimeMeasurement** ∗ **duplicate** ()
- virtual double **getValue** ()
- void **fromString** (string str)

Protected Member Functions

- virtual void **overwrite** (**TimeMeasurement** ∗time)

Protected Attributes

- double **expectedValue**
- double **actualValue**

2.96.1 Detailed Description

Definition at line 66 of file TimeMeasurement.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/TimeMeasurement.h

2.97 UnflexibleDuration Class Reference

Inheritance diagram for UnflexibleDuration:Collaboration diagram for UnflexibleDuration:

Public Member Functions

- **UnflexibleDuration** (double **expectedValue**)
- virtual ∼**UnflexibleDuration** ()
- **TimeMeasurement** ∗ **duplicate** ()
- double **getComputedValue** ()
- void **setComputedValue** (double value)
- bool **isPredictable** ()
- bool **isComputed** ()
- double **getValue** ()

Protected Member Functions

- void **overwrite** (**TimeMeasurement** ∗time)

2.97.1 Detailed Description

Definition at line 62 of file UnflexibleDuration.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/model/UnflexibleDuration.h

2.98 FormatterActiveDevice Class Reference

Inheritance diagram for FormatterActiveDevice:Collaboration diagram for FormatterActiveDevice:

Public Member Functions

- **FormatterActiveDevice** (IDeviceLayout ∗**deviceLayout**, int x, int y, int w, int h)

Protected Member Functions

- bool **newDeviceConnected** (int newDevClass, int w, int h)
- void **connectedToBaseDevice** (unsigned int domainAddr)
- bool **receiveRemoteEvent** (int remoteDevClass, int eventType, string eventContent)
- bool **receiveRemoteContent** (int remoteDevClass, char ∗stream, int streamSize)
- bool **receiveRemoteContent** (int remoteDevClass, string contentUri)
- bool **receiveRemoteContentInfo** (string contentId, string contentUri)
- bool **userEventReceived** (IInputEvent ∗ev)

Private Attributes

- IPlayer ∗ **player**
- map< string, string > ∗ **contentsInfo**

2.98.1 Detailed Description

Definition at line 61 of file FormatterActiveDevice.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/multidevice/FormatterActiveDevice.h

2.99 FormatterBaseDevice Class Reference

Inheritance diagram for FormatterBaseDevice:Collaboration diagram for FormatterBaseDevice:

Public Member Functions

- **FormatterBaseDevice** (IDeviceLayout ∗**deviceLayout**, int x, int y, int w, int h)

Protected Member Functions

- bool **newDeviceConnected** (int newDevClass, int w, int h)
- void **connectedToBaseDevice** (unsigned int domainAddr)
- bool **receiveRemoteEvent** (int remoteDevClass, int eventType, string eventContent)
- bool **receiveRemoteContent** (int remoteDevClass, char ∗stream, int streamSize)
- bool **receiveRemoteContent** (int remoteDevClass, string contentUri)
- bool **userEventReceived** (IInputEvent ∗ev)

2.99.1 Detailed Description

Definition at line 61 of file FormatterBaseDevice.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/multidevice/FormatterBaseDevice.h

2.100 FormatterMultiDevice Class Reference

Inheritance diagram for FormatterMultiDevice:Collaboration diagram for FormatterMultiDevice:

Public Member Functions

- **FormatterMultiDevice** (IDeviceLayout ∗**deviceLayout**, int x, int y, int w, int h)
- void **setBackgroundImage** (string uri)
- void ∗ **getMainLayout** ()
- void ∗ **getFormatterLayout** (int devClass)
- string **getScreenShot** ()
- FormatterLayout ∗ **getFormatterLayout** (CascadingDescriptor ∗descriptor)
- void **prepareFormatterRegion** (ExecutionObject ∗executionObject, ISurface ∗renderedSurface)
- void **showObject** (ExecutionObject ∗executionObject)
- void **hideObject** (ExecutionObject ∗executionObject)
- void **addActiveUris** (string baseUri, vector< string > ∗uris)
- virtual void **updatePassiveDevices** ()

Protected Member Functions

- string **serializeScreen** (int devClass, IWindow ∗mapWindow)
- virtual void **postMediaContent** (int destDevClass)
- virtual bool **newDeviceConnected** (int newDevClass, int w, int h)
- virtual void **connectedToBaseDevice** (unsigned int domainAddr)=0
- virtual bool **receiveRemoteEvent** (int remoteDevClass, int eventType, string eventContent)
- virtual bool **receiveRemoteContent** (int remoteDevClass, char ∗stream, int streamSize)
- virtual bool **receiveRemoteContentInfo** (string contentId, string contentUri)
- void **renderFromUri** (IWindow ∗win, string uri)
- void **tapObject** (int devClass, int x, int y)
- virtual bool **receiveRemoteContent** (int remoteDevClass, string contentUri)
- void **updateStatus** (short code, string parameter, short type)
- virtual bool **userEventReceived** (IInputEvent ∗ev)=0

Protected Attributes

- IDeviceLayout ∗ **deviceLayout**
- map< int, FormatterLayout ∗ > ∗ **layoutManager**
- vector< string > ∗ **activeUris**
- string **activeBaseUri**
- FormatterLayout ∗ **mainLayout**
- IWindow ∗ **serialized**
- IWindow ∗ **printScreen**
- IWindow ∗ **bitMapScreen**
- int **xOffset**
- int **yOffset**
- int **defaultWidth**
- int **defaultHeight**
- int **deviceClass**
- bool **hasRemoteDevices**

Static Protected Attributes

- static ILocalDeviceManager * **dm**
- static IInputManager * **im**
- static pthread_mutex_t **mutex**
- static IRemoteDeviceManager * **rdm**
- static const int **DV_QVGA_WIDTH** = 480
- static const int **DV_QVGA_HEIGHT** = 320

2.100.1 Detailed Description

Definition at line 90 of file FormatterMultiDevice.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/multidevice/FormatterMultiDevice.h

2.101 FormatterPassiveDevice Class Reference

Inheritance diagram for FormatterPassiveDevice:Collaboration diagram for FormatterPassiveDevice:

Public Member Functions

- **FormatterPassiveDevice** (IDeviceLayout ∗**deviceLayout**, int x, int y, int w, int h)

Protected Member Functions

- void **postMediaContent** (int destDevClass)
- bool **newDeviceConnected** (int newDevClass, int w, int h)
- void **connectedToBaseDevice** (unsigned int domainAddr)
- bool **receiveRemoteEvent** (int remoteDevClass, int eventType, string eventContent)
- bool **receiveRemoteContent** (int remoteDevClass, char ∗stream, int streamSize)
- bool **receiveRemoteContent** (int remoteDevClass, string contentUri)
- bool **userEventReceived** (IInputEvent ∗ev)

2.101.1 Detailed Description

Definition at line 61 of file FormatterPassiveDevice.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/multidevice/FormatterPassiveDevice.h

2.102 IFormatterMultiDevice Class Reference

Inheritance diagram for IFormatterMultiDevice:Collaboration diagram for IFormatterMultiDevice:

Public Member Functions

- virtual ~**IFormatterMultiDevice** ()
- virtual void **setBackgroundImage** (string uri)=0
- virtual void * **getMainLayout** ()=0
- virtual void * **getFormatterLayout** (int devClass)=0
- virtual string **getScreenShot** ()=0
- virtual void **addActiveUris** (string baseUri, vector< string > *uris)=0
- virtual void **updatePassiveDevices** ()=0

Protected Member Functions

- virtual bool **newDeviceConnected** (int newDevClass, int w, int h)=0
- virtual void **connectedToBaseDevice** (unsigned int domainAddr)=0
- virtual bool **receiveRemoteEvent** (int remoteDevClass, int eventType, string eventContent)=0
- virtual bool **receiveRemoteContent** (int remoteDevClass, char *stream, int streamSize)=0
- virtual bool **receiveRemoteContent** (int remoteDevClass, string contentUri)=0
- virtual void **updateStatus** (short code, string parameter, short type)=0

2.102.1 Detailed Description

Definition at line 72 of file IFormatterMultiDevice.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/multidevice/IFormatterMultiDevice.h

2.103 IPrefetchManager Class Reference

Inheritance diagram for IPrefetchManager:

Public Member Functions

- virtual ~**IPrefetchManager** ()
- virtual void **release** ()=0
- virtual void **releaseContents** ()=0
- virtual string **createDocumentPrefetcher** (string remoteDocUri)=0
- virtual string **createSourcePrefetcher** (string localDocUri, string srcUri)=0
- virtual bool **hasIChannel** ()=0
- virtual bool **hasRemoteLocation** (string localUri)=0
- virtual string **getRemoteLocation** (string localUri)=0
- virtual string **getLocalRoot** (string remoteUri)=0
- virtual void **setSynchPrefetch** (bool synch)=0
- virtual void **getScheduledContent** (string clientUri)=0
- virtual void **getScheduledContents** ()=0

2.103.1 Detailed Description

Definition at line 62 of file IPrefetchManager.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/prefetch/IPrefetchManager.h

2.104 PrefetchManager Class Reference

Inheritance diagram for PrefetchManager:Collaboration diagram for PrefetchManager:

Public Member Functions

- void **release** ()
- void **releaseContents** ()
- string **createDocumentPrefetcher** (string remoteDocUri)
- string **createSourcePrefetcher** (string localDocUri, string srcUri)
- bool **hasIChannel** ()
- bool **hasRemoteLocation** (string localUri)
- string **getRemoteLocation** (string localUri)
- string **getLocalRoot** (string remoteUri)
- void **setSynchPrefetch** (bool **synch**)
- void **getScheduledContent** (string clientUri)
- void **getScheduledContents** ()

Static Public Member Functions

- static **IPrefetchManager** ∗ **getInstance** ()

Private Member Functions

- void **createDirectory** (string newDir)
- void **getContent** (string remoteUri, string localUri)
- void **scheduleContent** (string remoteUri, string localUri)
- void **receiveCode** (long respCode)
- void **receiveDataStream** (char ∗buffer, int size)
- void **receiveDataPipe** (int fd, int size)
- void **downloadCompleted** (const char ∗localUri)

Private Attributes

- map< string, string > ∗ **localToRemoteUris**
- map< string, string > ∗ **urisToLocalRoots**
- set< string > ∗ **scheduledRemoteUris**
- set< string > ∗ **scheduledLocalUris**
- IInteractiveChannelManager ∗ **icm**
- string **prefetchRoot**
- bool **synch**
- int **kbytes**
- int **filesDown**
- int **filesSched**

Static Private Attributes

- static **IPrefetchManager** ∗ **_instance**

2.104.1 Detailed Description

Definition at line 66 of file PrefetchManager.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/prefetch/PrefetchManager.h

2.105 PrivateBaseContext Class Reference

Inheritance diagram for PrivateBaseContext:Collaboration diagram for PrivateBaseContext:

Public Member Functions

- void **createPrivateBase** (string id)
- NclDocument * **addDocument** (string location, IDeviceLayout *deviceLayout)
- NclDocument * **embedDocument** (string docId, string nodeId, string location, IDeviceLayout *deviceLayout)
- void * **addVisibleDocument** (string location, IDeviceLayout *deviceLayout)
- string **getDocumentLocation** (string docId)
- string **getEmbeddedDocumentLocation** (string parentDocId, string nodeId)
- NclDocument * **getDocument** (string id)
- NclDocument * **getEmbeddedDocument** (string parendDocId, string nodeId)
- vector< NclDocument * > * **getDocuments** ()
- NclDocument * **removeDocument** (string id)
- NclDocument * **removeEmbeddedDocument** (string parentDocId, string nodeId)
- void **clear** ()
- LayoutRegion * **addRegion** (string documentId, string regionBaseId, string regionId, string xmlRegion)
- LayoutRegion * **removeRegion** (string documentId, string regionBaseId, string regionId)
- RegionBase * **addRegionBase** (string documentId, string xmlRegionBase)
- RegionBase * **removeRegionBase** (string documentId, string regionBaseId)
- Rule * **addRule** (string documentId, string xmlRule)
- Rule * **removeRule** (string documentId, string ruleId)
- RuleBase * **addRuleBase** (string documentId, string xmlRuleBase)
- RuleBase * **removeRuleBase** (string documentId, string ruleBaseId)
- Transition * **addTransition** (string documentId, string xmlTransition)
- Transition * **removeTransition** (string documentId, string transitionId)
- TransitionBase * **addTransitionBase** (string documentId, string xmlTransitionBase)
- TransitionBase * **removeTransitionBase** (string documentId, string transitionBaseId)
- Connector * **addConnector** (string documentId, string xmlConnector)
- Connector * **removeConnector** (string documentId, string connectorId)
- ConnectorBase * **addConnectorBase** (string documentId, string xmlConnectorBase)
- ConnectorBase * **removeConnectorBase** (string documentId, string connectorBaseId)
- GenericDescriptor * **addDescriptor** (string documentId, string xmlDescriptor)
- GenericDescriptor * **removeDescriptor** (string documentId, string descriptorId)
- DescriptorBase * **addDescriptorBase** (string documentId, string xmlDescriptorBase)
- DescriptorBase * **removeDescriptorBase** (string documentId, string descriptorBaseId)
- Base * **addImportBase** (string documentId, string docBaseId, string xmlImportBase)
- Base * **removeImportBase** (string documentId, string docBaseId, string documentURI)
- NclDocument * **addImportedDocumentBase** (string documentId, string xmlImportedDocumentBase)
- NclDocument * **removeImportedDocumentBase** (string documentId, string importedDocumentBaseId)
- NclDocument * **addImportNCL** (string documentId, string xmlImportNCL)
- NclDocument * **removeImportNCL** (string documentId, string documentURI)
- Node * **addNode** (string documentId, string compositeId, string xmlNode)
- InterfacePoint * **addInterface** (string documentId, string nodeId, string xmlInterface)
- Link * **addLink** (string documentId, string compositeId, string xmlLink)
- PrivateBase * **getPrivateBase** ()

Private Member Functions

- NclDocument * **compileDocument** (string location, IDeviceLayout *deviceLayout)
- NclDocument * **getBaseDocument** (string documentId)
- void * **compileEntity** (string location, NclDocument *document, void *parentObject)
- Base * **getBase** (NclDocument *document, string baseId)

Private Attributes

- map< string, NclDocument * > * **baseDocuments**
- map< string, NclDocument * > * **visibleDocuments**
- map< string, **EmbeddedNclData** * > * **embeddedDocuments**
- map< string, string > * **idToLocation**
- map< NclDocument *, IDeviceLayout * > * **layouts**
- set< IDeviceLayout * > * **layoutsGB**
- PrivateBase * **privateBase**

2.105.1 Detailed Description

Definition at line 113 of file PrivateBaseContext.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/privatebase/PrivateBaseContext.h

2.106 PrivateBaseManager Class Reference

Inheritance diagram for PrivateBaseManager:Collaboration diagram for PrivateBaseManager:

Public Member Functions

- void **createPrivateBase** (string id)
- void **release** ()
- NclDocument * **addDocument** (string id, string location, IDeviceLayout *deviceLayout)
- NclDocument * **embedDocument** (string id, string docId, string nodeId, string location, IDeviceLayout *deviceLayout)
- void * **addVisibleDocument** (string id, string location, IDeviceLayout *deviceLayout)
- string **getDocumentLocation** (string id, string docId)
- string **getEmbeddedDocumentLocation** (string baseId, string parentDocId, string nodeId)
- NclDocument * **getDocument** (string id, string docId)
- NclDocument * **getEmbeddedDocument** (string baseId, string parendDocId, string nodeId)
- vector< NclDocument * > * **getDocuments** (string id)
- NclDocument * **removeDocument** (string id, string docId)
- NclDocument * **removeEmbeddedDocument** (string baseId, string parentDocId, string nodeId)
- void **clear** (string id)
- LayoutRegion * **addRegion** (string id, string documentId, string regionBaseId, string regionId, string xmlRegion)
- LayoutRegion * **removeRegion** (string id, string documentId, string regionBaseId, string regionId)
- RegionBase * **addRegionBase** (string id, string documentId, string xmlRegionBase)
- RegionBase * **removeRegionBase** (string id, string documentId, string regionBaseId)
- Rule * **addRule** (string id, string documentId, string xmlRule)
- Rule * **removeRule** (string id, string documentId, string ruleId)
- RuleBase * **addRuleBase** (string id, string documentId, string xmlRuleBase)
- RuleBase * **removeRuleBase** (string id, string documentId, string ruleBaseId)
- Transition * **addTransition** (string id, string documentId, string xmlTransition)
- Transition * **removeTransition** (string id, string documentId, string transitionId)
- TransitionBase * **addTransitionBase** (string id, string documentId, string xmlTransitionBase)
- TransitionBase * **removeTransitionBase** (string id, string documentId, string transitionBaseId)
- Connector * **addConnector** (string id, string documentId, string xmlConnector)
- Connector * **removeConnector** (string id, string documentId, string connectorId)
- ConnectorBase * **addConnectorBase** (string id, string documentId, string xmlConnectorBase)
- ConnectorBase * **removeConnectorBase** (string id, string documentId, string connectorBaseId)
- GenericDescriptor * **addDescriptor** (string id, string documentId, string xmlDescriptor)
- GenericDescriptor * **removeDescriptor** (string id, string documentId, string descriptorId)
- DescriptorBase * **addDescriptorBase** (string id, string documentId, string xmlDescriptorBase)
- DescriptorBase * **removeDescriptorBase** (string id, string documentId, string descriptorBaseId)
- Base * **addImportBase** (string id, string documentId, string docBaseId, string xmlImportBase)
- Base * **removeImportBase** (string id, string documentId, string docBaseId, string documentURI)
- NclDocument * **addImportedDocumentBase** (string id, string documentId, string xmlImportedDocumentBase)
- NclDocument * **removeImportedDocumentBase** (string id, string documentId, string importedDocumentBaseId)
- NclDocument * **addImportNCL** (string id, string documentId, string xmlImportNCL)
- NclDocument * **removeImportNCL** (string id, string documentId, string documentURI)
- Node * **addNode** (string id, string documentId, string compositeId, string xmlNode)

- InterfacePoint * **addInterface** (string id, string documentId, string nodeId, string xmlInterface)
- Link * **addLink** (string id, string documentId, string compositeId, string xmlLink)
- PrivateBase * **getPrivateBase** (string id)

Static Public Member Functions

- static **PrivateBaseManager** * **getInstance** ()

Private Member Functions

- **PrivateBaseContext** * **getPrivateBaseContext** (string id)
- void **lockTable** ()
- void **unlockTable** ()

Private Attributes

- map< string, **PrivateBaseContext** * > * **privateBases**
- pthread_mutex_t **mutexTable**

Static Private Attributes

- static **PrivateBaseManager** * **_instance**

2.106.1 Detailed Description

Definition at line 61 of file PrivateBaseManager.h.

The documentation for this class was generated from the following file:

- gingancl-cpp/include/privatebase/PrivateBaseManager.h

2.107 AgentListener Class Reference

Inheritance diagram for AgentListener:

Public Member Functions

- virtual void **volUp** ()=0
- virtual void **volDown** ()=0
- virtual void **channelUp** ()=0
- virtual void **channelDown** ()=0
- virtual void **channelChange** (const char ∗info)=0
- virtual void **nclChange** (const char ∗info)=0

2.107.1 Detailed Description

Definition at line 10 of file AgentListener.h.

2.107.2 Member Function Documentation

2.107.2.1 virtual void volUp () [pure virtual]

Método acionado quando o volume é incrementado.

Implemented in **IAgentListener** (p. **??**).

2.107.2.2 virtual void volDown () [pure virtual]

Método acionado quando o volume é decrementado.

Implemented in **IAgentListener** (p. **??**).

2.107.2.3 virtual void channelUp () [pure virtual]

Método acionado quando o canal é incrementado.

Implemented in **IAgentListener** (p. **??**).

2.107.2.4 virtual void channelDown () [pure virtual]

Método acionado quando o canal é decrementado.

Implemented in **IAgentListener** (p. **??**).

2.107.2.5 virtual void channelChange (const char ∗ *info*) [pure virtual]

Método acionado quando ocorre uma mudança de canal.

Parameters:

info informações do canal anterior e do novo canal sintonizado.

Implemented in **IAgentListener** (p. **??**).

2.107.2.6 virtual void nclChange (const char * *info*) [pure virtual]

Método acionado quando ocorre uma interação com uma aplicação NCL.

Parameters:

info informações da interação realizada.

Implemented in **IAgentListener** (p. **??**).

The documentation for this class was generated from the following file:

- Recommender/AgentListener/include/AgentListener.h

2.108 Channel Class Reference

Inheritance diagram for Channel:Collaboration diagram for Channel:

Public Member Functions

- **Channel** (int **code**, string **name**)
- string **generateSql** ()
- int **getIdChannel** ()
- int **getCode** ()
- void **setCode** (int **code**)
- string **getName** ()
- void **setName** (string **name**)

Static Public Member Functions

- static void **setIdChannel** (int id)
- static void **incIdChannel** ()

Private Attributes

- int **code**
- string **name**

Static Private Attributes

- static int **idChannel**

2.108.1 Detailed Description

Definition at line 14 of file Channel.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/Channel.h

2.109 Context Class Reference

Collaboration diagram for Context:

Public Member Functions

- **Context** (string **id**, **Context** ∗**context**, **Document** ∗**document**)
- string **generateSql** ()
- int **getIdContext** ()
- string **getId** ()
- void **setId** (string **id**)
- **Context** ∗ **getContext** ()
- void **setContext** (**Context** ∗**context**)
- **Document** ∗ **getDocument** ()
- void **setDocument** (**Document** ∗**document**)

Static Public Member Functions

- static void **setIdContext** (int **id**)
- static void **incIdContext** ()

Private Attributes

- string **id**
- **Context** ∗ **context**
- **Document** ∗ **document**

Static Private Attributes

- static int **idContext**

2.109.1 Detailed Description

Definition at line 16 of file Context.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/Context.h

2.110 Database Class Reference

Collaboration diagram for Database:

Public Member Functions

- **Database** (const char ∗**User**, const char ∗Password, const char ∗**Database**)
- ∼**Database** ()
- bool **createDatabase** ()
- bool **deleteDatabase** ()
- bool **closeDatabase** ()
- void **setMaxSize** (long MaxSize)
- long **getMaxSize** ()
- long **getCurrentSize** ()
- bool **query** (const char ∗sql)
- bool **query** (const char ∗sql, vector< string > ∗head, vector< string > ∗data)

Private Attributes

- string **user**
- string **password**
- string **database**
- long **maxSize**
- sqlite3 ∗ **db**

  Abstração para o banco de dados sqlite.

2.110.1 Detailed Description

Definition at line 37 of file Database.h.

2.110.2 Constructor & Destructor Documentation

2.110.2.1 Database (const char ∗ *User*, const char ∗ *Password*, const char ∗ *Database*)

Construtor. Devolve uma instância de **Database** (p. **??**).

Parameters:

user nome de usuário do banco de dados.

password senha do banco de dados.

database nome do banco de dados para conexão ou criação.

2.110.2.2 ∼Database ()

Destrutor.

2.110.3 Member Function Documentation

2.110.3.1 bool createDatabase ()

Cria o banco de dados com o nome, usuário e senha especificados no construtor.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.110.3.2 bool deleteDatabase ()

Apaga o banco de dados com o nome, usuário e senha especificados no construtor.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.110.3.3 bool closeDatabase ()

Fecha a conexão com o banco de dados especificado no construtor.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.110.3.4 void setMaxSize (long *MaxSize*)

Configura o tamanho máximo que o banco de dados poderá ocupar no disco. Deve ser acionado antes da chamada do método **createDatabase()** (p. **??**).

Parameters:

maxSize tamanho máximo (em bytes) que o banco de dados poderá ocupar no disco.

2.110.3.5 long getMaxSize ()

Retorna o tamanho máximo que o banco de dados pode ocupar no disco.

Returns:

tamanho máximo (em bytes) que o banco de dados pode ocupar no disco.

2.110.3.6 long getCurrentSize ()

Retorna o tamanho atual do banco de dados no disco.

Returns:

tamanho atual (em bytes) do banco de dados no disco.

2.110.3.7 bool query (const char * *sql*)

Executa um comando SQL.

Parameters:

sql comando SQL.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.110.3.8 bool query (const char * *sql*, vector< string > * *head*, vector< string > * *data*)

Executa um comando SQL.

Parameters:

sql comando SQL.

head variável para retorno do cabeçalho da consulta SQL.

data variável para retorno dos dados da consulta SQL.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

The documentation for this class was generated from the following file:

- Recommender/Database/include/Database.h

2.111 DataMining Class Reference

Collaboration diagram for DataMining:

Public Member Functions

- **DataMining** ()
- **DataMining** (const char ∗algorithm)
- **DataMining** (const char ∗lib, const char ∗constructor, const char ∗destructor)
- virtual ∼**DataMining** ()
- bool **setAlgorithm** (const char ∗algorithm)
- const char ∗ **getAlgorithm** ()
- vector< char ∗ > ∗ **getAlgorithms** ()
- bool **setParams** (vector< char ∗ > params)
- bool **prepare** (const char ∗data, const char ∗delimiter)
- bool **prepare** (**Database** ∗db, const char ∗table)
- bool **start** ()
- bool **stop** ()
- void **setOutput** (char ∗data, long size, const char ∗delimiter)
- void **setOutput** (**Database** ∗db, const char ∗table)
- void **setOutput** (const char ∗file)

Public Attributes

- **MiningAlgorithm** ∗ **miningAlgorithm**

Private Types

- enum **MiningAlgoritmKeys** { **Apriori** = 1 }

Private Member Functions

- void **setAvaibleMiningAlgorithms** ()

Private Attributes

- map< char ∗, int > **avaibleMiningAlgorithms**
- string **currentAlgorithm**
- void ∗(∗ **algorithmConstructor**)(void)
- void ∗ **algorithmPointer**
- void(∗ **algorithmDestructor**)(void ∗)

2.111.1 Detailed Description

Definition at line 26 of file DataMining.h.

2.111.2 Member Enumeration Documentation

2.111.2.1 enum MiningAlgoritmKeys [private]

Enumerator:

Apriori

Definition at line 137 of file DataMining.h.

2.111.3 Constructor & Destructor Documentation

2.111.3.1 DataMining ()

Construtor. Devolve uma instância de **DataMining** (p. **??**).

2.111.3.2 DataMining (const char * *algorithm*)

Construtor. Devolve uma instância de **DataMining** (p. **??**).

Parameters:

algorithm algoritmo que será usado para o processamento dos dados.

2.111.3.3 DataMining (const char * *lib*, const char * *constructor*, const char * *destructor*)

Construtor. Devolve uma instância de **DataMining** (p. **??**).

Parameters:

lib nome da biblioteca que contém o algoritmo a ser executado.

constructor construtor do algoritmo que devolve uma instância para a interface **MiningAlgorithm** (p. **??**).

destructor destrutor do algoritmo.

2.111.3.4 virtual ~DataMining () [virtual]

Destrutor.

2.111.4 Member Function Documentation

2.111.4.1 bool setAlgorithm (const char * *algorithm*)

Define o algoritmo que será usado para o processamento dos dados.

Parameters:

algorithm algoritmo que será usado para o processamento dos dados.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.111.4.2 const char* getAlgorithm ()

Retorna o algoritmo que será usado para o processamento dos dados.

Returns:

algoritmo que será usado para o processamento dos dados.

2.111.4.3 vector<char*>* getAlgorithms ()

Retorna todos os algoritmos disponíveis para processamento dos dados.

Returns:

todos os algoritmos disponíveis para processamento dos dados.

2.111.4.4 bool setParams (vector< char * > *params*)

Inicializa o algoritmo com os parâmetros especificados.

Parameters:

params parâmetros para o algoritmo.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.111.4.5 bool prepare (const char * *data*, const char * *delimiter*)

Prepara os dados para serem processados pelo algoritmo.

Parameters:

data dados a serem pre-processados.
delimiter delimitador dos dados.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.111.4.6 bool prepare (Database * *db*, const char * *table*)

Prepara os dados para serem processados pelo algoritmo.

Parameters:

db banco de dados que contém a tabela a ser pre-processada.
table tabela com os dados a serem pre-processados.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.111.4.7 bool start ()

Executa o algoritmo.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.111.4.8 bool stop ()

Pára a execução do algoritmo.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.111.4.9 void setOutput (char * *data*, long *size*, const char * *delimiter*)

Define a saída dos dados processados.

Parameters:

data variável para armazenamento dos dados processados.
size tamanho (em bytes) da variável para armazenamento dos dados.
delimiter delimitador dos dados.

2.111.4.10 void setOutput (Database * *db*, const char * *table*)

Define a saída dos dados processados.

Parameters:

db banco de dados que contém a tabela onde serão armazenados os dados processados.
table tabela onde serão armazenados os dados processados.

2.111.4.11 void setOutput (const char * *file*)

Define a saída dos dados processados.

Parameters:

file caminho do arquivo para armazenamento dos dados processados.

2.111.4.12 void setAvaibleMiningAlgorithms () [private]

Inicializa o mapa com os nomes dos algoritmos de mineração de dados disponíveis.

The documentation for this class was generated from the following file:

- Recommender/DataMining/include/DataMining.h

2.112 Device Class Reference

Collaboration diagram for Device:

Public Member Functions

- **Device** (string **prof**, string **serialNumber**, string **tp**)
- string **generateSql** ()
- int **getIdDevice** ()
- string **getProf** ()
- void **setProf** (string **prof**)
- string **getSerialNumber** ()
- void **setSerialNumber** (string **serialNumber**)
- string **getTp** ()
- void **setTp** (string **tp**)

Static Public Member Functions

- static void **setIdDevice** (int id)
- static void **incIdDevice** ()

Private Attributes

- string **prof**
- string **serialNumber**
- string **tp**

Static Private Attributes

- static int **idDevice**

2.112.1 Detailed Description

Definition at line 15 of file Device.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/Device.h

2.113 Document Class Reference

Collaboration diagram for Document:

Public Member Functions

- **Document** (string **id**, **NclStateMachine** ∗**nclStateMachine**)
- string **generateSql** ()
- int **getIdDocument** ()
- string **getId** ()
- void **setId** (string **id**)
- **NclStateMachine** ∗ **getNclStateMachine** ()
- void **setNclStateMachine** (**NclStateMachine** ∗**nclStateMachine**)

Static Public Member Functions

- static void **setIdDocument** (int **id**)
- static void **incIdDocument** ()

Private Attributes

- string **id**
- **NclStateMachine** ∗ **nclStateMachine**

Static Private Attributes

- static int **idDocument**

2.113.1 Detailed Description

Definition at line 16 of file Document.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/Document.h

2.114 eit Class Reference

`#include <EIT.h>`

Inheritance diagram for eit:Collaboration diagram for eit:

Public Member Functions

- char * **getDescriptor_conteudo_genero** ()
- char * **getDescriptor_conteudo_subgenero** ()
- char * **getDescriptor_event_short_sinopse** ()
- char * **getEnd_time** ()
- char * **getStart_time** ()
- void **setDescriptor_conteudo_genero** (const char *__descriptor_conteudo_genero__)
- void **setDescriptor_conteudo_subgenero** (const char *__descriptor_conteudo_subgenero__)
- void **setDescriptor_event_short_sinopse** (const char *__descriptor_event_short_sinopse__)
- void **setEnd_time** (const char *__end_time__)
- void **setStart_time** (const char *__start_time__)

Private Attributes

- char **start_time** [24]

  inicio do programa - máxmo de 12 bytes

- char **end_time** [24]

  término do programa - máximo de 12 bytes

- char **descriptor_conteudo_genero** [5]

  gênero do programa - máximo de 4 bytes

- char **descriptor_conteudo_subgenero** [5]

  sub-gênero do programa - máximo de 4 bytes

- char **descriptor_event_short_sinopse** [129]

  sinopse do programa máximo de 128 bytes

2.114.1 Detailed Description

Fornece uma abstração para a tabela Eit. A tabela Eit é responsável em prover informações sobre os serviços disponibilizados pelos provedores de serviços

Author:

Paulo Muniz de Ávila

Definition at line 24 of file EIT.h.

2.114.2 Member Function Documentation

2.114.2.1 char* getDescriptor_conteudo_genero ()

Recuperar a informação com o gênero do serviço Ver Norma NBR 15603-2 para maiores informações

Returns:

um char * com o gênero do serviço

2.114.2.2 char* getDescriptor_conteudo_subgenero ()

Recuperar as informações com o sub-gênero do serviço Ver Norma NBR 15603-2 para maiores informações

Returns:

um char * com o sub-gênero do serviço

2.114.2.3 char* getDescriptor_event_short_sinopse ()

Recuperar as informações de sinopse do serviço

Returns:

Um ponteiro de 128 bytes com a sinopse

2.114.2.4 char* getEnd_time ()

Recuperar o horário de término do programa

Returns:

char * com o horário informado pelo provedor de serviço

2.114.2.5 char* getStart_time ()

Recuperar o horário de início do programa

Returns:

char * com o horário informado pelo provedor de serviço

2.114.2.6 void setDescriptor_conteudo_genero (const char * *descriptor_conteudo_genero*)

Armazenar o gênero no objeto da classe Eit

Parameters:

descriptor_conteudo_genero char * com o código do gênero

2.114.2.7 void setDescriptor_conteudo_subgenero (const char * *descriptor_conteudo_subgenero*)

Armazenar o sub-gênero no objeto da classe Eit

Parameters:

descriptor_conteudo_subgenero char * com o código do gênero

2.114.2.8 void setDescriptor_event_short_sinopse (const char * *descriptor_event_short_sinopse*)

Armazenar uma sinopse no objeto da classe Eit

Parameters:

descriptor_event_short_sinopse um char * com 128 bytes descrevendo a sinopse

2.114.2.9 void setEnd_time (const char * *end_time*)

Armazenar o horário de término do serviço no objeto da classe Eit

Parameters:

end_time um char * com o término do serviço

2.114.2.10 void setStart_time (const char * *start_time*)

Armazenar o horário de início do serviço no objeto da classe Eit

Parameters:

start_time um char * com o horário de início do programa

The documentation for this class was generated from the following file:

- Recommender/EIT/include/EIT.h

2.115 IAgentListener Class Reference

Inheritance diagram for IAgentListener:Collaboration diagram for IAgentListener:

Public Member Functions

- virtual void **volUp** ()=0
- virtual void **volDown** ()=0
- virtual void **channelUp** ()=0
- virtual void **channelDown** ()=0
- virtual void **channelChange** (const char ∗info)=0
- virtual void **nclChange** (const char ∗info)=0

2.115.1 Detailed Description

Definition at line 8 of file IAgentListener.h.

2.115.2 Member Function Documentation

2.115.2.1 virtual void volUp () [pure virtual]

Método acionado quando o volume é incrementado.

Implements **AgentListener** (p. **??**).

2.115.2.2 virtual void volDown () [pure virtual]

Método acionado quando o volume é decrementado.

Implements **AgentListener** (p. **??**).

2.115.2.3 virtual void channelUp () [pure virtual]

Método acionado quando o canal é incrementado.

Implements **AgentListener** (p. **??**).

2.115.2.4 virtual void channelDown () [pure virtual]

Método acionado quando o canal é decrementado.

Implements **AgentListener** (p. **??**).

2.115.2.5 virtual void channelChange (const char ∗ *info*) [pure virtual]

Método acionado quando ocorre uma mudança de canal.

Parameters:

info informações do canal anterior e do novo canal sintonizado.

Implements **AgentListener** (p. **??**).

2.115.2.6 virtual void nclChange (const char * *info*) [pure virtual]

Método acionado quando ocorre uma interação com uma aplicação NCL.

Parameters:

info informações da interação realizada.

Implements **AgentListener** (p. **??**).

The documentation for this class was generated from the following file:

- Recommender/AgentListener/include/IAgentListener.h

2.116 IMiningAlgorithmApriori Class Reference

Inheritance diagram for IMiningAlgorithmApriori:Collaboration diagram for IMiningAlgorithmApriori:

Public Types

- typedef unsigned long **itemtype**

  define o tipo itemtype utilizado pelo algoritmo apriori para contagem dos elementos

Public Member Functions

- **IMiningAlgorithmApriori** ()
- ∼**IMiningAlgorithmApriori** ()
- bool **setParams** (vector< char * > params)
- bool **prepare** (const char *data, const char *delimiter)
- bool **prepare** (**Database** *db, const char *table)
- bool **start** ()
- bool **stop** ()
- void **setOutput** (char *data, long size, const char *delimiter)
- void **setOutput** (**Database** *db, const char *table)
- void **setOutput** (const char *file)
- void **execute** (string file, double min_sup, double min_confi)
- void **encode** (char *fileCodeTable, char *fileOutMining, char *fileOut)
- void **code_file_process** (char *code_file_name)
- void **code_file_process_out_Algorithm** (char *outAlgorithm)
- void **code_file_process_out** (char *outAlgorithm)
- void **write_decoded_item** (**itemtype** item)
- void **decode** ()
- vector< **si** * > * **restoreCacheInformation** (string nomeCache)
- vector< **User** * > * **getUserInformation** ()
- vector< std::string > **prepareDataUser** (vector< **User** * > *user)
- void **saveUserInformation** (vector< string > user, string file)
- vector< **User** * > * **mountObject** (vector< string > retornaConsulta, vector< string > retornaHead)
- string **getValueByTime** (string dataColectedByUser)
- string **getPercentualByTime** (const string dataColectedByUser, vector< **si** * > *dataCacheByTable)
- vector< string > **getPercentualByTime** (const string nomePrograma, double tempoUsuario, string dataInicio, vector< **si** * > *dataCacheByTable)
- int **getRandom** (int min, int max)
- int **ReadFile** (char *arquivo)
- vector< string > **getLine** ()
- void **setLine** (vector< string > **line**)
- vector< string > **searchProgrambyId** (string genero, string subgenero, string per_dia, vector< string > *programacao)
- string **getGenProgram** (string gen, string subgen)
- void **controlDuplicate** ()
- void **saveRecommender** (string arquivo, vector< string > *user, string horario)
- void **prepareToSaveRecommender** ()

Static Public Member Functions

- static std::string **verifyDuration** (double duration)

Public Attributes

- int **count**
- vector< string > **line**
- string * **inData**
- string * **inDataDelimiter**
- **Database** * **inDB**
- string * **inTableName**
- char * **outDataBuffer**
- long * **outDataSize**
- string * **outDataDelimiter**
- **Database** * **outDB**
- string * **outTableName**
- string * **outFileName**
- double **minSup**
- double **minConfi**
- string **siCacheFileName**
- vector< **User** * > * **userInfo**
- FILE * **code_file_algorithm**

  abstração para o arquivo que contém os dados da interação do usuário

- ofstream **code_file_out**
- map< **itemtype**, string > **codearray**

  conta a quantidade de itens (programas) de acordo com a base de dados do usuário

- string **imprimir**
- ofstream **outfile**

Static Public Attributes

- static const int **DESPREZIVEL** = 10
- static const int **POUCO** = 30
- static const int **MEDIO** = 65
- static const int **ALTO** = 100

2.116.1 Detailed Description

Definition at line 54 of file IMiningAlgorithmApriori.h.

2.116.2 Constructor & Destructor Documentation

2.116.2.1 IMiningAlgorithmApriori ()

Construtor.

2.116.2.2 ~IMiningAlgorithmApriori ()

Destrutor.

2.116.3 Member Function Documentation

2.116.3.1 bool setParams (vector< char * > *params*) [virtual]

Inicializa o algoritmo com os parâmetros especificados.

Parameters:

params parâmetros para o algoritmo: params[0]: min_sup parametros do algoritmo. Utilizado para desconsiderar poucas interações do usuário. params[1]: min_confi Confiabilidade dos dados. Quanto maior esse parâmetro, menos regras são produzidas. params[2]: banco de dados tabela contendo a tabela SI (uma abstração para as tabelas EIT e SDT).

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

Implements **MiningAlgorithm** (p. **??**).

2.116.3.2 bool prepare (const char * *data*, const char * *delimiter*) [virtual]

Prepara os dados para serem processados pelo algoritmo.

Parameters:

data dados a serem pre-processados.

delimiter delimitador dos dados.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

Implements **MiningAlgorithm** (p. **??**).

2.116.3.3 bool prepare (Database * *db*, const char * *table*) [virtual]

Prepara os dados para serem processados pelo algoritmo.

Parameters:

db banco de dados que contém a tabela a ser pre-processada.

table tabela com os dados a serem pre-processados.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

Implements **MiningAlgorithm** (p. **??**).

2.116.3.4 bool start () [virtual]

Executa o algoritmo.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

Implements **MiningAlgorithm** (p. **??**).

2.116.3.5 bool stop () [virtual]

Pára a execução do algoritmo.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

Implements **MiningAlgorithm** (p. **??**).

2.116.3.6 void setOutput (char * *data*, long *size*, const char * *delimiter*) [virtual]

Define a saída dos dados processados.

Parameters:

data variável para armazenamento dos dados processados.

size tamanho (em bytes) da variável para armazenamento dos dados.

delimiter delimitador dos dados.

Implements **MiningAlgorithm** (p. **??**).

2.116.3.7 void setOutput (Database * *db*, const char * *table*) [virtual]

Define a saída dos dados processados.

Parameters:

db banco de dados que contém a tabela onde serão armazenados os dados processados.

table tabela onde serão armazenados os dados processados.

Implements **MiningAlgorithm** (p. **??**).

2.116.3.8 void setOutput (const char * *file*) [virtual]

Define a saída dos dados processados.

Parameters:

file caminho do arquivo para armazenamento dos dados processados.

Implements **MiningAlgorithm** (p. **??**).

2.116.3.9 void execute (string *file*, double *min_sup*, double *min_confi*)

Executa o algoritmo apriori

Parameters:

file arquivo que contém as interações do usuário (comportamento do usuário)
min_sup parametros do algoritmo. Utilizado para desconsiderar poucas interações do usuário.
min_confi Confiabilidade dos dados. Quanto maior esse parâmetro, menos regras são produzidas.

2.116.3.10 void encode (char * *fileCodeTable*, char * *fileOutMining*, char * *fileOut*)

O algoritmo apriori produz a saída toda em formato numérico. Esse método mapea a saída do algoritmo numérico para strings.

Parameters:

fileCodeTable tabela usada para o mapeamento string –> numérico
fileOutMining arquivo gerado pelo algoritmo apriori (numérico)
fileOut arquivo de saída mapeado numérico –> string

2.116.3.11 void write_decoded_item (itemtype *item*)

Adiciona o item decodificado à string que será impressa ao final da execução.

Parameters:

item elemento que será escrito.

2.116.3.12 void decode ()

Decodifica o arquivo que contém os dados da interação do usuário com o middleware.

2.116.3.13 vector<si*>* restoreCacheInformation (string *nomeCache*)

Recupera as informações das tabelas EIT e SDT

Parameters:

nomeCache o nome do cache que contém as informações das tabelas EIT & SDT

Returns:

um ponteiro para um vector contendo os registros recuperados.

2.116.3.14 vector<User*>* getUserInformation ()

Converte informações relacionais do banco de dados em objetos da classe **User** (p. **??**).

Returns:

um ponteiro de vector <User*>

2.116.3.15 vector<std::string> prepareDataUser (vector< User * > * *user*)

Converte objetos da classe **User** (p. **??**) em strings para serem submetidas aos algoritmos de mineração

Parameters:

user um ponteiro para um vector<User *> *user

Returns:

string com os objetos da classe **User** (p. **??**) convertidos no padrão basket_file do algoritmo apriori.

2.116.3.16 void saveUserInformation (vector< string > *user*, string *file*)

Realiza a persistência das informações processadas pelo método prepareDataUser

Parameters:

user recebe uma string com os dados a serem persistidos na

file recebe uma string que informa o path e nome do arquivo a ser criado em disco. em disco.

2.116.3.17 vector<User *>* mountObject (vector< string > *retornaConsulta*, vector< string > *retornaHead*)

Monta objetos da classe **User** (p. **??**) a partir das tuplas passadas por referência.

Parameters:

retornaConsulta : dados contidos nas tuplas.

retornaHead : nome das tabelas.

Returns:

vetor com os usuários.

2.116.3.18 string getValueByTime (string *dataColectedByUser*)

Calcular o período do dia em que um programa foi sintonizado. Exemplo: Programa sintonizado das 09:30hs até 11:00hs. retorna: manhã.

Parameters:

dataColectedByUser string contendo as informações do programa, horário, data.

Returns:

string informando em qual horário o programa foi sintonizado

2.116.3.19 string getPercentualByTime (const string *dataColectedByUser*, vector< si * > * *dataCacheByTable*)

Calcular o percentual de tempo que o usuário sintonizou determinando programa.

Parameters:

dataColectedByUser string contendo as informações do programa, horário, data.

dataCacheByTable vector <si*> * um ponteiro para os dados restaurados da tabela EIT & SDT.

Returns:

string informando: pouco, medio,alto

2.116.3.20 vector<string> getPercentualByTime (const string *nomePrograma*, double *tempoUsuario*, string *dataInicio*, vector< si * > * *dataCacheByTable*)

Calcular o percentual de tempo que o usuário sintonizou determinando programa.

Parameters:

nomePrograma string contendo as informações do programa para comparação com as informações da EIT no cache.

tempoUsuario tempoUsuario double contendo o tempo em que o usuário ficou sintonizado no programa

dataInicio string contendo a data atual para recuperar os programas da tabela EIT & SDT

dataCacheByTable vector si* um ponteiro para os dados restaurados da tabela EIT & SDT.

Returns:

vector contendo o tempo que o usuário assitiu cada programa.

2.116.3.21 int getRandom (int *min*, int *max*)

Quando não existirem dados suficiente para o algoritmo apriori processar é necessário preencher o arquivo de saída lua com informações randômicas.

Parameters:

min informa o menor valor gerado

max informa o máximo valor gerado

Returns:

int onde esse valor é maior igual ao minimo e menor igual ao máximo.

2.116.3.22 static std::string verifyDuration (double *duration*) [static]

Retorna uma string representando o tempo que usuário permaneceu conectado em determinado programa.

Parameters:

duration double que representa o percentual de tempo em que o usuário ficou conectado em um determinado programa.

Returns:

string que pode ser: pouco, medio, alto

2.116.3.23 int ReadFile (char * *arquivo*)

Leitura de arquivo texto do disco

Parameters:

arquivo nome do arquivo a ser lido

Returns:

número de bytes lidos do arquivo

2.116.3.24 vector<string> searchProgrambyId (string *genero*, string *subgenero*, string *per_dia*, vector< string > * *programacao*)

Busca na tabela EIT o programa recomendado

Parameters:

genero string que informa o genero do programa

subgenero string que informa o sub-genero do programa

per_dia string que informa o dia para procurar na tabela EIT

programacao a tabela EIT

Returns:

2.116.3.25 string getGenProgram (string *gen*, string *subgen*)

retorna o Genero do Programa a partir do codigo

Parameters:

gen string com o genero

Returns:

o genero, por exemplo, Jornalismo

2.116.3.26 void controlDuplicate ()

Evita que ocorram duplicações nas recomedações

2.116.3.27 void saveRecommender (string *arquivo*, vector< string > * *user*, string *horario*)

Salva a recomendação no arquivo de formato lua. Exemplo de arquivo salvo: NOITE{ Genero Cod = "0x0", SubGenero Cod = "0x02", Inicio = "2008-03-12 23:32:00", Fim = "2008-03-13 00:29:00", Programa = "REPORTER RECORD SG", Emissora = "006", }

Parameters:

arquivo nome do arquivo para salvar as recomendações

user recomendações ofertadas para o usuário

horario que pode ser: madrugada, manha, tarde e noite

2.116.3.28 void prepareToSaveRecommender ()

Prepara o banco para receber as novas recomendacoes

2.116.4 Member Data Documentation

2.116.4.1 ofstream code_file_out

abstração para o arquivo de saída produzido pelo algoritmo apriori. O arquivo de saída ainda não está pronto, necessitando ser submetido aos algoritmos do módulo FilterAgent

See also:

br.ufscar.lince.ginga.recommender.filteragent

Definition at line 352 of file IMiningAlgorithmApriori.h.

The documentation for this class was generated from the following file:

- Recommender/MiningAlgorithm/include/IMiningAlgorithmApriori.h

2.117 Iteraction Class Reference

Inheritance diagram for Iteraction:Collaboration diagram for Iteraction:

Public Member Functions

- **Iteraction** (**Program** ∗**program**, **Volume** ∗**volume**, **Key** ∗**key**, string **user**, string **time**, string **type**)
- string **generateSql** (string tecla, string volum, string **program**)
- int **getIdIteraction** ()
- **Program** ∗ **getProgram** ()
- void **setProgram** (**Program** ∗**program**)
- **Volume** ∗ **getVolume** ()
- void **setVolume** (**Volume** ∗**volume**)
- **Key** ∗ **getKey** ()
- void **setKey** (**Key** ∗**key**)
- **Device** ∗ **getDevice** ()
- void **setDevice** (**Device** ∗**device**)
- string **getUser** ()
- void **setUser** (string **user**)
- string **getTime** ()
- void **setTime** (string **time**)
- string **getType** ()
- void **setType** (string **type**)

Static Public Member Functions

- static void **setIdIteraction** (int id)
- static void **incIdIteraction** ()

Private Attributes

- **Program** ∗ **program**
- **Volume** ∗ **volume**
- **Key** ∗ **key**
- **Device** ∗ **device**
- string **user**
- string **time**
- string **type**

Static Private Attributes

- static int **idIteraction**

2.117.1 Detailed Description

Definition at line 21 of file Iteraction.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/Iteraction.h

2.118 IteractionBase Class Reference

Inheritance diagram for IteractionBase:

2.118.1 Detailed Description

Definition at line 15 of file IteractionBase.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/IteractionBase.h

2.119 Key Class Reference

Inheritance diagram for Key:Collaboration diagram for Key:

Public Member Functions

- **Key** (string **code**, string **action**)
- string **generateSql** ()
- int **getIdKey** ()
- string **getCode** ()
- void **setCode** (string **code**)
- string **getAction** ()
- void **setAction** (string **action**)
- string **getType** ()

Static Public Member Functions

- static void **setIdKey** (int id)
- static void **incIdKey** ()

Private Attributes

- string **code**
- string **action**

Static Private Attributes

- static int **idKey**

2.119.1 Detailed Description

Definition at line 16 of file Key.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/Key.h

2.120 LocalAgent Class Reference

Inheritance diagram for LocalAgent:Collaboration diagram for LocalAgent:

Public Member Functions

- **LocalAgent** ()
- ∼**LocalAgent** ()
- **Database** ∗ **getDatabase** ()
- bool **stop** ()
- bool **addAgentListener** (**AgentListener** ∗listener)
- bool **removeAgentListener** (**AgentListener** ∗listener)
- void **run** ()
- bool **getEstado** ()
- void **setEstado** (bool **estado**)
- string **getValue** (string varAmbiente)
- void **setPressedKey** (int **pressedKey**)
- int **getPressedKey** ()
- bool **addIteraction** (**Iteraction** ∗iteraction)
- bool **removeIteraction** (**Iteraction** ∗iteraction)
- string **getKeyName** (int keyCode)
- void **setTuner** (ITuner ∗**tuner**)
- void **setPrimaryKey** ()
- void **setNewUser** (string ID, string userGenre, string locationZip, string age)

Static Public Member Functions

- static **LocalAgent** ∗ **getInstance** ()

Private Attributes

- vector< **AgentListener** ∗ > ∗ **listeners**
- vector< **Iteraction** ∗ > **iters**
- **Database** ∗ **db**
- ITuner ∗ **tuner**
- int **pressedKey**
- bool **estado**
- IComponentManager ∗ **cm**
- IStateManager ∗ **stateManager**
- IPresentationState ∗ **ps**
- **Document** ∗ **document**
- **NclStateMachine** ∗ **nclStateMachine**
- **Context** ∗ **context**
- **Media** ∗ **media**
- **User** ∗ **user**

Static Private Attributes

- static **LocalAgent** ∗ **_instance**

2.120.1 Detailed Description

Definition at line 68 of file LocalAgent.h.

2.120.2 Constructor & Destructor Documentation

2.120.2.1 LocalAgent ()

Construtor vazio

2.120.2.2 ~LocalAgent ()

Destrutor.

2.120.3 Member Function Documentation

2.120.3.1 static LocalAgent* getInstance () [static]

Retorna a instancia única da classe

2.120.3.2 Database* getDatabase ()

Retorna o banco de dados onde são armazendas as informações coletadas.

Returns:

banco de dados onde são armazendas as informações coletadas.

2.120.3.3 bool stop ()

Pára a execução do **LocalAgent** (p. **??**).

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.120.3.4 bool addAgentListener (AgentListener * *listener*)

Adiciona um **AgentListener** (p. **??**) ao **LocalAgent** (p. **??**).

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.120.3.5 bool removeAgentListener (AgentListener * *listener*)

Remove um **AgentListener** (p. **??**) do **LocalAgent** (p. **??**).

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/LocalAgent.h

2.121 Media Class Reference

Collaboration diagram for Media:

Public Member Functions

- **Media** (string **id**, string **status**, string **time**, **Context** ∗**context**, **Document** ∗**document**)
- string **generateSql** ()
- int **getIdMedia** ()
- string **getId** ()
- void **setId** (string **id**)
- string **getStatus** ()
- void **setStatus** (string **status**)
- string **getTime** ()
- void **setTime** (string **time**)
- **Context** ∗ **getContext** ()
- void **setContext** (**Context** ∗**context**)
- **Document** ∗ **getDocument** ()
- void **setDocument** (**Document** ∗**document**)

Static Public Member Functions

- static void **setIdMedia** (int **id**)
- static void **incIdMedia** ()

Private Attributes

- string **id**
- string **status**
- string **time**
- **Context** ∗ **context**
- **Document** ∗ **document**

Static Private Attributes

- static int **idMedia**

2.121.1 Detailed Description

Definition at line 17 of file Media.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/Media.h

2.122 Meta Class Reference

Inheritance diagram for Meta:Collaboration diagram for Meta:

Public Member Functions

- **Meta** (**Program** ∗**program**, string **meta**)
- string **generateSql** ()
- int **getIdMeta** ()
- **Program** ∗ **getProgram** ()
- void **setProgram** (**Program** ∗**program**)
- string **getMeta** ()
- void **setMeta** (string **meta**)

Static Public Member Functions

- static void **setIdMeta** (int id)
- static void **incIdMeta** ()

Private Attributes

- **Program** ∗ **program**
- string **meta**

Static Private Attributes

- static int **idMeta**

2.122.1 Detailed Description

Definition at line 17 of file Meta.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/Meta.h

2.123 MiningAlgorithm Class Reference

Inheritance diagram for MiningAlgorithm:

Public Member Functions

- virtual bool **setParams** (vector< char * > params)=0
- virtual bool **prepare** (const char *data, const char *delimiter)=0
- virtual bool **prepare** (**Database** *db, const char *table)=0
- virtual bool **start** ()=0
- virtual bool **stop** ()=0
- virtual void **setOutput** (char *data, long size, const char *delimiter)=0
- virtual void **setOutput** (**Database** *db, const char *table)=0
- virtual void **setOutput** (const char *file)=0

2.123.1 Detailed Description

Definition at line 20 of file MiningAlgorithm.h.

2.123.2 Member Function Documentation

2.123.2.1 virtual bool setParams (vector< char * > *params*) [pure virtual]

Inicializa o algoritmo com os parâmetros especificados.

Parameters:

params parâmetros para o algoritmo.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

Implemented in **IMiningAlgorithmApriori** (p. **??**).

2.123.2.2 virtual bool prepare (const char * *data*, const char * *delimiter*) [pure virtual]

Prepara os dados para serem processados pelo algoritmo.

Parameters:

data dados a serem pre-processados.

delimiter delimitador dos dados.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

Implemented in **IMiningAlgorithmApriori** (p. **??**).

2.123.2.3 virtual bool prepare (Database * *db*, const char * *table*) [pure virtual]

Prepara os dados para serem processados pelo algoritmo.

Parameters:

db banco de dados que contém a tabela a ser pre-processada.

table tabela com os dados a serem pre-processados.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

Implemented in **IMiningAlgorithmApriori** (p. **??**).

2.123.2.4 virtual bool start () [pure virtual]

Executa o algoritmo.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

Implemented in **IMiningAlgorithmApriori** (p. **??**).

2.123.2.5 virtual bool stop () [pure virtual]

Pára a execução do algoritmo.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

Implemented in **IMiningAlgorithmApriori** (p. **??**).

2.123.2.6 virtual void setOutput (char * *data*, long *size*, const char * *delimiter*) [pure virtual]

Define a saída dos dados processados.

Parameters:

data variável para armazenamento dos dados processados.

size tamanho (em bytes) da variável para armazenamento dos dados.

delimiter delimitador dos dados.

Implemented in **IMiningAlgorithmApriori** (p. **??**).

2.123.2.7 virtual void setOutput (Database * *db*, const char * *table*) [pure virtual]

Define a saída dos dados processados.

Parameters:

db banco de dados que contém a tabela onde serão armazenados os dados processados.

table tabela onde serão armazenados os dados processados.

Implemented in **IMiningAlgorithmApriori** (p. **??**).

2.123.2.8 virtual void setOutput (const char * *file*) [pure virtual]

Define a saída dos dados processados.

Parameters:

file caminho do arquivo para armazenamento dos dados processados.

Implemented in **IMiningAlgorithmApriori** (p. **??**).

The documentation for this class was generated from the following file:

- Recommender/MiningAlgorithm/include/MiningAlgorithm.h

2.124 NclStateMachine Class Reference

Collaboration diagram for NclStateMachine:

Public Member Functions

- **NclStateMachine** (**Iteraction** ∗**iteraction**)
- string **generateSql** ()
- int **getIdNclStateMachine** ()
- **Iteraction** ∗ **getIteraction** ()
- void **setIteraction** (**Iteraction** ∗**iteraction**)

Static Public Member Functions

- static void **setIdNclStateMachine** (int id)
- static void **incIdNclStateMachine** ()

Private Attributes

- **Iteraction** ∗ **iteraction**

Static Private Attributes

- static int **idNclStateMachine**

2.124.1 Detailed Description

Definition at line 16 of file NclStateMachine.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/NclStateMachine.h

2.125 Program Class Reference

Inheritance diagram for Program:Collaboration diagram for Program:

Public Member Functions

- **Program** (int **age**, int **code**, **Channel** ∗**channel**, string **name**, string **genre**, string **subGenre**)
- string **generateSql** ()
- int **getIdProgram** ()
- int **getAge** ()
- void **setAge** (int **age**)
- int **getCode** ()
- void **setCode** (int **code**)
- **Channel** ∗ **getChannel** ()
- void **setChannel** (**Channel** ∗**channel**)
- string **getName** ()
- void **setName** (string **name**)
- string **getGenre** ()
- void **setGenre** (string **genre**)
- string **getSubGenre** ()
- void **setSubGenre** (string **subGenre**)

Static Public Member Functions

- static void **setIdProgram** (int id)
- static void **incIdProgram** ()

Private Attributes

- int **age**
- int **code**
- **Channel** ∗ **channel**
- string **name**
- string **genre**
- string **subGenre**

Static Private Attributes

- static int **idProgram**

2.125.1 Detailed Description

Definition at line 17 of file Program.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/Program.h

2.126 Scheduler Class Reference

Collaboration diagram for Scheduler:

Public Member Functions

- **Scheduler** ()
- ∼**Scheduler** ()
- bool **schedule** (const char ∗lib, const char ∗constructor, const char ∗destructor, vector< char ∗ > ∗params, const char ∗begin, const char ∗end, int period)
- bool **load** (const char ∗file)
- bool **store** (const char ∗file)
- bool **should_stop** (void)
- void **stop** (void)

Static Public Member Functions

- static **Scheduler** ∗ **getInstance** ()

Public Attributes

- double **timeTomin**

Private Attributes

- void ∗ **temp**
- volatile bool **stopflag**

Static Private Attributes

- static **Scheduler** ∗ **_instance**

2.126.1 Detailed Description

Definition at line 50 of file Scheduler.h.

2.126.2 Constructor & Destructor Documentation

2.126.2.1 Scheduler ()

Construtor. Devolve uma instância de **Scheduler** (p. **??**).

2.126.2.2 ∼Scheduler ()

Destrutor.

2.126.3 Member Function Documentation

2.126.3.1 static Scheduler* getInstance () [static]

Retorna uma instância da classe Agendadora

Returns:

Um objeto do tipo da classe **Scheduler** (p. **??**)

2.126.3.2 bool schedule (const char * *lib*, const char * *constructor*, const char * *destructor*, vector< char * > * *params*, const char * *begin*, const char * *end*, int *period*)

Agenda a execução de uma aplicação.

Parameters:

lib nome da biblioteca que contém a aplicação a ser executada.

constructor construtor da aplicação que devolve uma instância para a interface **SchedulerItem** (p. **??**).

destructor destrutor da aplicação.

params parâmetros para a aplicação.

begin data e horário para iniciar a execução da aplicação (formato: dd/MM/aaaa-hh:mm:ss).

end data e horário para o fim da execução da aplicação (formato: dd/MM/aaaa-hh:mm:ss).

period periodicidade do agendamento: 1 - uma única vez; 2 - uma vez por dia; 3 - indefinidamente; 4 - uma vez por semana; 5 - uma vez por mês.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.126.3.3 bool load (const char * *file*)

Carrega para a memória as informações de agendamentos armazenadas em arquivo.

Parameters:

file arquivo XML com informações de agendamentos.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.126.3.4 bool store (const char * *file*)

Salva em arquivo as informações de agendamentos em memória.

Parameters:

file arquivo XML para destino das informações de agendamentos.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.126.4 Member Data Documentation

2.126.4.1 Scheduler* _instance [static, private]

singleton

Definition at line 125 of file Scheduler.h.

The documentation for this class was generated from the following file:

- Recommender/Scheduler/include/Scheduler.h

2.127 SchedulerItem Class Reference

Public Member Functions

- virtual bool **init** (vector< char * > *params)=0
- virtual bool **start** ()=0
- virtual bool **stop** ()=0
- virtual bool **pause** ()=0

2.127.1 Detailed Description

Definition at line 14 of file SchedulerItem.h.

2.127.2 Member Function Documentation

2.127.2.1 virtual bool init (vector< char * > * *params*) [pure virtual]

Inicializa a aplicação com os parâmetros especificados.

Parameters:

params parâmetros para a aplicação.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.127.2.2 virtual bool start () [pure virtual]

Coloca a aplicação em execução.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.127.2.3 virtual bool stop () [pure virtual]

Pára a aplicação.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

2.127.2.4 virtual bool pause () [pure virtual]

Coloca a aplicação em modo pause.

Returns:

true, caso a operação seja executada com sucesso, ou false, caso contrário.

The documentation for this class was generated from the following file:

- Recommender/SchedulerItem/include/SchedulerItem.h

2.128 sdt Class Reference

`#include <SDT.h>`

Inheritance diagram for sdt:Collaboration diagram for sdt:

Public Member Functions

- char * **getServiceName** ()
- char * **getServiceProviderName** ()
- void **setServiceName** (const char *__serviceName__)
- void **setServiceProviderName** (const char *__serviceProviderName__)

Private Attributes

- char **serviceProviderName** [128]
- char **serviceName** [128]

2.128.1 Detailed Description

Criar um abstração para a tabela **sdt** (p. **??**) responsável em informar o nome do provedor de serviços e o serviço associado

Author:

Paulo Muniz de Ávila

Definition at line 22 of file SDT.h.

2.128.2 Member Function Documentation

2.128.2.1 char* getServiceName ()

Recuperar o nome do serviço, no sistema analógico o programa

Returns:

char * com o nome do serviço

2.128.2.2 char* getServiceProviderName ()

Recuperar o nome do provedor de serviços, no sistema analógico a emissora

Returns:

char * com o nome do provedor de serviços

2.128.2.3 void setServiceName (const char * *serviceName*)

Armazenar no objeto da classe Sdt o nome do serviço

Parameters:

serviceName char * com o nome do serviço

2.128.2.4 void setServiceProviderName (const char * *serviceProviderName*)

Armanezar no objeto da classe Sdt o provedor de Serviço

Parameters:

serviceProviderName char * com o nome do provedor de serviço

The documentation for this class was generated from the following file:

- Recommender/SDT/include/SDT.h

2.129 si Class Reference

Inheritance diagram for si:Collaboration diagram for si:

2.129.1 Detailed Description

Definition at line 26 of file SI.h.

The documentation for this class was generated from the following file:

- Recommender/MiningAlgorithm/include/SI.h

2.130 SystemResource Class Reference

Collaboration diagram for SystemResource:

Public Member Functions

- long **getCPUClock** ()
- int **getCPUCount** ()
- long **getMemorySize** ()
- long **getMemoryFree** ()
- off_t **getDiskSize** ()
- off_t **getDiskFree** ()
- int **reserve** (const char ∗applicationId, int resource, long count)
- bool **freeReserve** (const char ∗applicationId, int reserveId)
- bool **allocate** (const char ∗applicationId, int reserveId)
- bool **freeAllocation** (const char ∗applicationId, int reserveId)
- **SystemResource** ()
- ∼**SystemResource** ()

Private Member Functions

- void **initDataMembers** ()

Private Attributes

- utsname **sn**
- sysinfo **info**
- float **CPUClock**
- int **CPUCount**

2.130.1 Detailed Description

Definition at line 40 of file SystemResource.h.

2.130.2 Constructor & Destructor Documentation

2.130.2.1 SystemResource ()

Construtor.

2.130.2.2 ∼SystemResource ()

Destrutor.

2.130.3 Member Function Documentation

2.130.3.1 long getCPUClock ()

Retorna a frequência da CPU.

Returns:

frequência da CPU em Hz.

2.130.3.2 int getCPUCount ()

Retorna quantidade de CPUs da máquina.

Returns:

quantidade de CPUs da máquina.

2.130.3.3 long getMemorySize ()

Retorna a quantidade total de memória da máquina.

Returns:

total de memória em bytes.

2.130.3.4 long getMemoryFree ()

Retorna a quantidade de memória livre na máquina.

Returns:

total de memória livre (em bytes).

2.130.3.5 off_t getDiskSize ()

Retorna a quantidade total de espaço de armazenamento da máquina.

Returns:

total de espaço de armazenamento da máquina (em bytes).

2.130.3.6 off_t getDiskFree ()

Retorna a quantidade de espaço de armazenamento livre da máquina.

Returns:

espaço de armazenamento livre da máquina (em bytes).

2.130.3.7 int reserve (const char * *applicationId*, int *resource*, long *count*)

Reserva um recurso do sistema para uma aplicação.

Parameters:

applicationId identificação da aplicação que solicita o recurso.

resource identificação do recurso solicitado: 1 - CPU; 2 - Memória; 3 - Disco.

count quantidade do recurso solicitada. O retorno deste método devolve um identificador para que o recurso seja liberado mais tarde.

Returns:

maior que 0, se a operação foi executada com sucesso, ou -1, caso contrário.

2.130.3.8 bool freeReserve (const char * *applicationId*, int *reserveId*)

Desfaz uma reserva de recurso do sistema.

Parameters:

applicationId identificação da aplicação que solicitou o recurso.

reserveId identificador do recurso que será liberado.

Returns:

true, se a operação foi executada com sucesso, ou false, caso contrário.

2.130.3.9 bool allocate (const char * *applicationId*, int *reserveId*)

Faz uso de um recurso do sistema reservado anteriormente.

Parameters:

applicationId identificação da aplicação que solicitou o recurso.

reserveId identificador do recurso reservado.

Returns:

true, se a operação foi executada com sucesso, ou false, caso contrário.

2.130.3.10 bool freeAllocation (const char * *applicationId*, int *reserveId*)

Libera o uso de um recurso do sistema.

Parameters:

applicationId identificação da aplicação que solicitou o recurso.

reserveId identificador do recurso reservado.

Returns:

true, se a operação foi executada com sucesso, ou false, caso contrário.

2.130.3.11 void initDataMembers () [private]

Obtém os valores dos atributos da classe relativos à CPU.

The documentation for this class was generated from the following file:

- Recommender/SystemResource/include/SystemResource.h

2.131 Thread Class Reference

Inheritance diagram for Thread:Collaboration diagram for Thread:

Public Member Functions

- virtual void **start** ()
- bool **sleep** (long int seconds)
- bool **usleep** (long int milliseconds)
- void **wakeUp** ()
- void **lock** ()
- void **unlock** ()
- void **waitForUnlockCondition** ()
- bool **unlockConditionSatisfied** ()

Protected Member Functions

- virtual void **run** ()=0

Protected Attributes

- pthread_mutex_t **threadMutex**
- bool **isSleeping**
- pthread_mutex_t **threadFlagMutex**
- pthread_cond_t **threadFlagConditionVariable**
- bool **isWaiting**
- pthread_mutex_t **threadFlagMutexLockUntilSignal**
- pthread_cond_t **threadFlagCVLockUntilSignal**
- pthread_attr_t **tattr**

Static Private Member Functions

- static void ∗ **function** (void ∗ptr)

Private Attributes

- pthread_t **threadId_**

2.131.1 Detailed Description

Definition at line 62 of file Thread.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/thread/Thread.h

2.132 User Class Reference

`#include <User.h>`

Collaboration diagram for User:

Public Member Functions

- int **getCode** ()
- void **setCode** (int setcode)
- string **getKey** ()
- void **setKey** (string setKey)
- string **getDocid** ()
- void **setDocid** (string setdocid)
- string **getFilename** ()
- void **setFilename** (string setfilename)
- string **getHour** ()
- void **setHour** (string sethour)
- string **getMinute** ()
- void **setMinute** (string setminute)
- string **getSecond** ()
- void **setSecond** (string setsecond)
- string **getDay** ()
- void **setDay** (string setday)
- string **getMonth** ()
- void **setMonth** (string setmonth)
- string **getYear** ()
- void **setYear** (string setyear)
- string **getWeekDay** ()
- void **setWeekDay** (string setweekday)
- double **getTimeDuration** ()
- void **setTimeDuration** (double settimeduration)
- string **getFormatDate** ()
- string **getFormatTime** ()
- **User** (string **ID**, string **userGenre**, string **locationZip**, string **locationLatitude**, string **locationLongitude**, string **userBirth**, string **docId**, string **filename**)
- string **generateSql** ()
- string **generateSql2** ()
- string **generateSql3** ()
- string **generateSql4** ()
- string **generateSql5** (int id)
- int **getIdUser** ()
- void **setIdUser** (int id)
- void **incIdUser** ()
- string **getID** ()
- void **setID** (string **ID**)
- string **getUserGenre** ()
- void **setUserGenre** (string **userGenre**)
- string **getLocationZip** ()
- void **setLocationZip** (string **locationZip**)

- string **getLocationLatitude** ()
- void **setLocationLatitude** (string **locationLatitude**)
- string **getLocationLongitude** ()
- void **setLocationLongitude** (string **locationLongitude**)
- string **getUserBirth** ()
- void **setUserBirth** (string **userBirth**)
- **Device** ∗ **getDevice** ()
- void **setDevice** (**Device** ∗**device**)
- string **getDocId** ()
- void **setDocId** (string **docId**)
- string **getFilename** ()
- void **setFilename** (string **filename**)

Private Attributes

- int **code**
- string **keycode**
- string **doc_id**
- string **filename**
- string **hour**
- string **minute**
- string **second**
- string **day**
- string **month**
- string **year**
- string **dayweek**
- double **time**
- string **ID**
- string **userGenre**
- string **locationZip**
- string **locationLatitude**
- string **locationLongitude**
- string **userBirth**
- **Device** ∗ **device**
- string **docId**

Static Private Attributes

- static int **idUser**

2.132.1 Detailed Description

Class **User** (p. **??**) Representa todas as informações coletadas a partir da interação do usuário com o set-top box. Exemplo: codigo da tecla pressionada, nome do programa sendo assistido, hora e data.

Author:

Paulo Muniz de Ávila

Definition at line 75 of file User.h.

2.132.2 Member Function Documentation

2.132.2.1 int getCode ()

retorna o identificador do provedor de serviços

Returns:

um int que representa o provedor de serviço

2.132.2.2 void setCode (int *setcode*)

Armazena o provedor de serviços sintonizado no objeto da classe user

Parameters:

setcode o código que representa o provedor de serviço

2.132.2.3 string getKey ()

Retorna a tecla pressionada pelo usuário

Returns:

string informando a tecla

See also:

codemap

2.132.2.4 void setKey (string *setKey*)

Armazena a tecla pressionada pelo usuário no objeto da classe **User** (p. **??**)

Parameters:

setKey string que representa a tecla

See also:

codemap

2.132.2.5 string getDocid ()

Retorna o identificador do serviço sintonizado

Returns:

string que representa o serviço (canal no sistema analógico)

2.132.2.6 void setDocid (string *setdocid*)

Armazena o serviço sintonizado no objeto da classe **User** (p. **??**)

Parameters:

setdocid string que identifica o serviço sintonizado

2.132.2.7 string getFilename ()

Retorna o nome do programa sintonizado pelo usuário

Returns:

string que representa o nome do programa. **Tabela EIT, descritor de conteúdo**

2.132.2.8 void setFilename (string *setfilename*)

Armazena o nome do programa sintonizado no objeto da classe **User** (p. **??**)

Parameters:

setfilename nome do programa obtido da tabela EIT

2.132.2.9 string getHour ()

Retorna a hora atual no set-top box

Returns:

string informando a hora de 0 a 24.

2.132.2.10 void setHour (string *sethour*)

Armazena a hora atual do set-top box no objeto da classe **User** (p. **??**)

Parameters:

sethour string representando a hora de 0 a 24.

2.132.2.11 string getMinute ()

Retorna os minutos da hora atual do set-top box

Returns:

string informando o minuto de 0 a 59.

2.132.2.12 void setMinute (string *setminute*)

Armazena os minutos atuais do set-top box no objeto da classe **User** (p. **??**)

Parameters:

setminute string representando os minutos de 0 a 59.

2.132.2.13 string getSecond ()

Retorna os segundos da hora atual do set-top box

Returns:

string representando os segundos de 0 a 59.

2.132.2.14 void setSecond (string *setsecond*)

Armazena os segundos atuais da hora do set-top box no objeto da classe **User** (p. **??**)

Parameters:

setsecond string representando os segundos de 0 a 59

2.132.2.15 string getDay ()

Retorna o dia de acordo com a data setada no set-top box

Returns:

string que representa os dias de 1 a 31.

2.132.2.16 void setDay (string *setday*)

Armazena o dia de acordo com a data atual do set-top Box no objeto da classe **User** (p. **??**)

Parameters:

setday string que representa o dia atual de 1 a 31

2.132.2.17 string getMonth ()

Retorna o mês atual de acordo com a data no set-top box

Returns:

string que representa o mês de 1 a 12.

2.132.2.18 void setMonth (string *setmonth*)

Armazena o mês de acordo com a data atual do set-top Box no objeto da classe **User** (p. **??**)

Parameters:

setmonth string que representa o mês de 1 a 12.

2.132.2.19 string getYear ()

Retorna o ano atual de acordo com a data no set-top box

Returns:

string que informa o ano de 1900-2027.

2.132.2.20 void setYear (string *setyear*)

Armazena o ano de acordo com a data atual do set-top Box no objeto da classe **User** (p. **??**)

Parameters:

setyear string que representa o ano de 1900-2027

2.132.2.21 string getWeekDay ()

Retorna o dia da semana. Ex: domingo = 1, segunda = 2, ...

Returns:

string que representa o dia da semana 1 a 7.

2.132.2.22 void setWeekDay (string *setweekday*)

Armazena o dia da semana da data atual do set-top-box no objeto da classe **User** (p. **??**)

Parameters:

setweekday string que representa o dia da semana de 1 a 7

2.132.2.23 double getTimeDuration ()

Retorna o tempo que um determinado serviço ficou sintonizado

Returns:

double que representa os segundos que um determinado serviço ficou sintonizado

2.132.2.24 void setTimeDuration (double *settimeduration*)

Armazena o tempo que um determinado serviço ficou sintonizado no objeto da classe **User** (p. **??**)

Parameters:

settimeduration double representando o tempo em segundos

2.132.2.25 string getFormatDate ()

Retorna uma string com a data formatada como dd-mm-yyyy

Returns:

string com data preparada para inserção no banco sqlite

2.132.2.26 string getFormatTime ()

Retorna uma string com a hora formatada como hh:mm:ss

Returns:

string com a hora preparada para inserção no banco sqlite

The documentation for this class was generated from the following files:

- Recommender/MiningAlgorithm/include/User.h
- Recommender/LocalAgent/include/User.h

2.133 Volume Class Reference

Inheritance diagram for Volume:Collaboration diagram for Volume:

Public Member Functions

- **Volume** (int **level**, bool **mute**)
- string **generateSql** ()
- int **getIdVolume** ()
- int **getLevel** ()
- void **setLevel** (int **level**)
- int **getMute** ()
- void **setMute** (bool **mute**)

Static Public Member Functions

- static void **setIdVolume** (int id)
- static void **incIdVolume** ()

Private Attributes

- int **level**
- bool **mute**

Static Private Attributes

- static int **idVolume**

2.133.1 Detailed Description

Definition at line 16 of file Volume.h.

The documentation for this class was generated from the following file:

- Recommender/LocalAgent/include/Volume.h

2.134 CODEBOOK Struct Reference

Collaboration diagram for CODEBOOK:

Public Attributes

- char **Versionstr** [MaxVersionLength]
- int **BlockSizeX**
- int **BlockSizeY**
- int **CodebookSize**
- int **TotalFreq**
- int **BytesPerElement**
- int **MinValue**
- int **MaxValue**
- int **Preprocessing**
- char **GenerationMethod** [MaxGenMethodLength]
- **BOOKTYPE Book**
- int **AllocatedSize**

2.134.1 Detailed Description

Definition at line 46 of file cb.h.

The documentation for this struct was generated from the following file:

- Recommender/MiningAlgorithm/include/ikmeans/cb.h

2.135 COUNTER Struct Reference

Collaboration diagram for COUNTER:

Public Attributes

- llong ∗ **counter**
- int **freq**

2.135.1 Detailed Description

Definition at line 37 of file cb.h.

The documentation for this struct was generated from the following file:

- Recommender/MiningAlgorithm/include/ikmeans/cb.h

2.136 Edge Struct Reference

`#include <Trie.hpp>`

Collaboration diagram for Edge:

Public Attributes

- itemtype **label**
- **Trie** ∗ **subtrie**

2.136.1 Detailed Description

Struct que representa um **Trie** (p. **??**)

Definition at line 82 of file Trie.hpp.

The documentation for this struct was generated from the following file:

- Recommender/MiningAlgorithm/include/apriori23/Trie.hpp

2.137 EmbeddedNclData Struct Reference

Collaboration diagram for EmbeddedNclData:

Public Attributes

- string **embeddedNclNodeId**
- string **embeddedNclNodeLocation**
- NclDocument * **embeddedDocument**

2.137.1 Detailed Description

Definition at line 102 of file PrivateBaseContext.h.

The documentation for this struct was generated from the following file:

- gingancl-cpp/include/privatebase/PrivateBaseContext.h

2.138 Graph Struct Reference

Collaboration diagram for Graph:

Public Attributes

- int **nvec**
- int **k**
- int **dim**
- int **maxcoord**
- int **mincoord**
- **GraphVector** ∗∗ **vectors**

2.138.1 Detailed Description

Definition at line 88 of file cb.h.

The documentation for this struct was generated from the following file:

- Recommender/MiningAlgorithm/include/ikmeans/cb.h

2.139 GraphVector Struct Reference

Collaboration diagram for GraphVector:

Public Attributes

- int **index**
- int * **data**
- int * **kindices**

2.139.1 Detailed Description

Definition at line 83 of file cb.h.

The documentation for this struct was generated from the following file:

- Recommender/MiningAlgorithm/include/ikmeans/cb.h

2.140 Input_Output_Manager Class Reference

`#include <Input_Output_Manager.hpp>`

Inheritance diagram for Input_Output_Manager:Collaboration diagram for Input_Output_Manager:

Public Member Functions

- **Input_Output_Manager** (ifstream &**basket_file**, const char *output_file_name)
- int **read_in_a_line** (set< itemtype > &basket)

  Realizar a leitura de uma linha do arquivo de entrada (uma transação).

- unsigned long **find_frequent_items** (const double min_supp, vector< unsigned long > &support_of_items)

  Determina a frequencia de cada item.

- void **basket_recode** (const set< itemtype > &original_basket, vector< itemtype > &new_basket)

  Criar um arquivo temporário com os itemsets mais frequentes.

- void **write_out_basket** (const set< itemtype > &itemset)

  Escreve um itemset para um arquivo.

- void **write_out_basket_and_counter** (const set< itemtype > &itemset, const unsigned long counter)

  Escreve o itemset e seu contador de frequencia para um arquivo.

- void **rewind** ()

Private Attributes

- ifstream & **basket_file**

  Arquivo que contém as transações. Arquivo de entrada do algoritmo.

- vector< itemtype > **new_code**
- vector< itemtype > **new_code_inverse**

2.140.1 Detailed Description

Essa classe é responsável pelas operações de entrada e saída (I/O Operations)

Definition at line 78 of file Input_Output_Manager.hpp.

2.140.2 Member Data Documentation

2.140.2.1 vector<itemtype> new_code [private]

Vector que representa a frequência de cada item.

Definition at line 115 of file Input_Output_Manager.hpp.

2.140.2.2 vector<itemtype> new_code_inverse [private]

O inverso de new_code vector.

Definition at line 120 of file Input_Output_Manager.hpp.

The documentation for this class was generated from the following file:

- Recommender/MiningAlgorithm/include/apriori23/Input_Output_Manager.hpp

2.141 IterativeKMeans Class Reference

`#include <ikmeans.h>`

Collaboration diagram for IterativeKMeans:

Public Member Functions

- **IterativeKMeans** (vector< vector< int > > **points**)
- vector< int > & **getClusterIds** ()
- vector< vector< int > > & **getCentroids** ()
- bool **cluster** (int k)
- bool **cluster** (int Minclus, int Maxclus)

Private Member Functions

- void **matrix2CB** (int **mat, int vecCount, int vecLen, **CODEBOOK** *CB)
- double **RMSE** (double *x, double *y, int start, int end, double a, double b)
- void **getFunction** (double *x, double *y, int start, int end, double *a, double *b)
- int **LMethod** (double *x, double *y, int N)
- int **predictedNumberClusters** (double *bicValues, int Minclus, int Maxclus)

Private Attributes

- vector< vector< int > > **points**
- vector< int > **clusterid**
- vector< vector< int > > **centroids**

2.141.1 Detailed Description

Iterative Clustering with prediction of number of clusters. Cluster method is k-means. Prediction of number of clusters uses L-Method and BIC. For a description of L-Method, see "Determining the Number of Clusters/Segments in Hierarchical Clustering/ Segmentation Algorithms", by Stan Salvador and Philip Chan.

Author:

Marco Cristo and Angelo Filipe. K-means code and some auxiliary code by Marco Tuononen (available at `http://cs.joensuu.fi/~mtuonon/`). BIC code by Qinpei Zhao (available at `http://cs.joensuu.fi/~zhao/Software/`).

Definition at line 28 of file ikmeans.h.

2.141.2 Constructor & Destructor Documentation

2.141.2.1 IterativeKMeans (vector< vector< int > > *points*) `[inline]`

Constructor for **IterativeKMeans** (p. **??**).

Definition at line 98 of file ikmeans.h.

References points.

2.141.3 Member Function Documentation

2.141.3.1 void matrix2CB (int ∗∗ *mat*, int *vecCount*, int *vecLen*, CODEBOOK ∗ *CB*) [private]

Convert integer matrix to internal KMeans Format.

Parameters:

mat Pre-alocated integer matrix (int ∗∗m)

vecCount Number of lines in m (number of vectors)

vecLen Number of columns in m (vector dimension)

2.141.3.2 double RMSE (double ∗ *x*, double ∗ *y*, int *start*, int *end*, double *a*, double *b*) [private]

Returns root mean squared error between function ax + b and set of points (x,y) in interval [start,end).

Parameters:

x Vector of x coordinates.

y Vector of y coordinates.

start Interval start.

end Interval end.

a Function slope.

b Function intercept.

Returns:

Root mean squared error.

2.141.3.3 void getFunction (double ∗ *x*, double ∗ *y*, int *start*, int *end*, double ∗ *a*, double ∗ *b*) [private]

Finds slope a and intercept b of best fit function of a set of points (x,y) in interval [start,end).

Parameters:

x Vector of x coordinates.

y Vector of y coordinates.

start Interval start.

end Interval end.

a Function slope.

b Function intercept.

Returns:

Point which corresponds to line "knee".

2.141.3.4 int LMethod (double * *x*, double * *y*, int *N*) [private]

Finds "knee" of a line (given by its points [x,y]) using L-method. For a description of L-Method, see "Determining the Number of Clusters/Segments in Hierarchical Clustering/ Segmentation Algorithms", by Stan Salvador and Philip Chan.

Parameters:

x Vector of x coordinates.

y Vector of y coordinates.

N Number of points.

Returns:

Point which corresponds to line "knee".

2.141.3.5 int predictedNumberClusters (double * *bicValues*, int *Minclus*, int *Maxclus*) [private]

Predicts the number of clusters considering the BIC values associated with several different partitions. The bicValues correspond to partitions from Minclus to Maxclus clusters. This method can only be applied to a set with at least 4 partitions (4 points in the line function).

Parameters:

bicValues Vector of BIC values.

Minclus Minimum number of clusters.

Maxclus Maximum number of clusters.

Returns:

Number of clusters.

2.141.3.6 vector<int>& getClusterIds () [inline]

Returns cluster ids found after last clustering.

Returns:

Vector of cluster ids.

Definition at line 107 of file ikmeans.h.

References clusterid.

2.141.3.7 vector<vector<int> >& getCentroids () [inline]

Returns centroids found after last clustering.

Returns:

Vector of centroids.

Definition at line 115 of file ikmeans.h.

References centroids.

2.141.3.8 bool cluster (int *k*)

Cluster the vectors in points into k clusters.

Parameters:

k Number of clusters.

Returns:

True, if clustering succeded.

2.141.3.9 bool cluster (int *Minclus*, int *Maxclus*)

Perform clustering on vectors in points, choosing the best partition from Minclus to Maxclus clusters.

Parameters:

Minclus Minimum number of clusters.

Maxclus Maximum number of clusters.

Returns:

True, if clustering succeded.

2.141.4 Member Data Documentation

2.141.4.1 vector<int> clusterid [private]

Points (vectors) to be clustered

Definition at line 32 of file ikmeans.h.

Referenced by getClusterIds().

2.141.4.2 vector<vector<int> > centroids [private]

Cluster ids assigned to each point

Definition at line 33 of file ikmeans.h.

Referenced by getCentroids().

The documentation for this class was generated from the following file:

- Recommender/MiningAlgorithm/include/ikmeans/ikmeans.h

2.142 PARTITIONING Struct Reference

Collaboration diagram for PARTITIONING:

Public Attributes

- char **Versionstr** [MaxVersionLength]
- int **PartitionCount**
- int **TSsize**
- int ∗ **Map**
- int ∗ **First**
- int ∗ **Next**
- **COUNTER** ∗ **CC**
- int ∗ **Uniques**
- int **Vsize**
- char **GenerationMethod** [MaxGenMethodLength]
- int **AllocatedSize**

2.142.1 Detailed Description

Definition at line 66 of file cb.h.

The documentation for this struct was generated from the following file:

- Recommender/MiningAlgorithm/include/ikmeans/cb.h

2.143 sqlite3_api_routines Struct Reference

Collaboration diagram for sqlite3_api_routines:

Public Attributes

- void *(* **aggregate_context**)(sqlite3_context *, int nBytes)
- int(* **aggregate_count**)(sqlite3_context *)
- int(* **bind_blob**)(sqlite3_stmt *, int, const void *, int n, void(*)(void *))
- int(* **bind_double**)(sqlite3_stmt *, int, double)
- int(* **bind_int**)(sqlite3_stmt *, int, int)
- int(* **bind_int64**)(sqlite3_stmt *, int, sqlite_int64)
- int(* **bind_null**)(sqlite3_stmt *, int)
- int(* **bind_parameter_count**)(sqlite3_stmt *)
- int(* **bind_parameter_index**)(sqlite3_stmt *, const char *zName)
- const char *(* **bind_parameter_name**)(sqlite3_stmt *, int)
- int(* **bind_text**)(sqlite3_stmt *, int, const char *, int n, void(*)(void *))
- int(* **bind_text16**)(sqlite3_stmt *, int, const void *, int, void(*)(void *))
- int(* **bind_value**)(sqlite3_stmt *, int, const sqlite3_value *)
- int(* **busy_handler**)(sqlite3 *, int(*)(void *, int), void *)
- int(* **busy_timeout**)(sqlite3 *, int ms)
- int(* **changes**)(sqlite3 *)
- int(* **close**)(sqlite3 *)
- int(* **collation_needed**)(sqlite3 *, void *, void(*)(void *, sqlite3 *, int eTextRep, const char *))
- int(* **collation_needed16**)(sqlite3 *, void *, void(*)(void *, sqlite3 *, int eTextRep, const void *))
- const void *(* **column_blob**)(sqlite3_stmt *, int iCol)
- int(* **column_bytes**)(sqlite3_stmt *, int iCol)
- int(* **column_bytes16**)(sqlite3_stmt *, int iCol)
- int(* **column_count**)(sqlite3_stmt *pStmt)
- const char *(* **column_database_name**)(sqlite3_stmt *, int)
- const void *(* **column_database_name16**)(sqlite3_stmt *, int)
- const char *(* **column_decltype**)(sqlite3_stmt *, int i)
- const void *(* **column_decltype16**)(sqlite3_stmt *, int)
- double(* **column_double**)(sqlite3_stmt *, int iCol)
- int(* **column_int**)(sqlite3_stmt *, int iCol)
- sqlite_int64(* **column_int64**)(sqlite3_stmt *, int iCol)
- const char *(* **column_name**)(sqlite3_stmt *, int)
- const void *(* **column_name16**)(sqlite3_stmt *, int)
- const char *(* **column_origin_name**)(sqlite3_stmt *, int)
- const void *(* **column_origin_name16**)(sqlite3_stmt *, int)
- const char *(* **column_table_name**)(sqlite3_stmt *, int)
- const void *(* **column_table_name16**)(sqlite3_stmt *, int)
- const unsigned char *(* **column_text**)(sqlite3_stmt *, int iCol)
- const void *(* **column_text16**)(sqlite3_stmt *, int iCol)
- int(* **column_type**)(sqlite3_stmt *, int iCol)
- sqlite3_value *(* **column_value**)(sqlite3_stmt *, int iCol)
- void *(* **commit_hook**)(sqlite3 *, int(*)(void *), void *)
- int(* **complete**)(const char *sql)
- int(* **complete16**)(const void *sql)

- int(∗ **create_collation**)(sqlite3 ∗, const char ∗, int, void ∗, int(∗)(void ∗, int, const void ∗, int, const void ∗))
- int(∗ **create_collation16**)(sqlite3 ∗, const void ∗, int, void ∗, int(∗)(void ∗, int, const void ∗, int, const void ∗))
- int(∗ **create_function**)(sqlite3 ∗, const char ∗, int, int, void ∗, void(∗xFunc)(sqlite3_context ∗, int, sqlite3_value ∗∗), void(∗xStep)(sqlite3_context ∗, int, sqlite3_value ∗∗), void(∗xFinal)(sqlite3_context ∗))
- int(∗ **create_function16**)(sqlite3 ∗, const void ∗, int, int, void ∗, void(∗xFunc)(sqlite3_context ∗, int, sqlite3_value ∗∗), void(∗xStep)(sqlite3_context ∗, int, sqlite3_value ∗∗), void(∗xFinal)(sqlite3_context ∗))
- int(∗ **create_module**)(sqlite3 ∗, const char ∗, const **sqlite3_module** ∗, void ∗)
- int(∗ **data_count**)(sqlite3_stmt ∗pStmt)
- sqlite3 ∗(∗ **db_handle**)(sqlite3_stmt ∗)
- int(∗ **declare_vtab**)(sqlite3 ∗, const char ∗)
- int(∗ **enable_shared_cache**)(int)
- int(∗ **errcode**)(sqlite3 ∗db)
- const char ∗(∗ **errmsg**)(sqlite3 ∗)
- const void ∗(∗ **errmsg16**)(sqlite3 ∗)
- int(∗ **exec**)(sqlite3 ∗, const char ∗, sqlite3_callback, void ∗, char ∗∗)
- int(∗ **expired**)(sqlite3_stmt ∗)
- int(∗ **finalize**)(sqlite3_stmt ∗pStmt)
- void(∗ **free**)(void ∗)
- void(∗ **free_table**)(char ∗∗result)
- int(∗ **get_autocommit**)(sqlite3 ∗)
- void ∗(∗ **get_auxdata**)(sqlite3_context ∗, int)
- int(∗ **get_table**)(sqlite3 ∗, const char ∗, char ∗∗∗, int ∗, int ∗, char ∗∗)
- int(∗ **global_recover**)(void)
- void(∗ **interruptx**)(sqlite3 ∗)
- sqlite_int64(∗ **last_insert_rowid**)(sqlite3 ∗)
- const char ∗(∗ **libversion**)(void)
- int(∗ **libversion_number**)(void)
- void ∗(∗ **malloc**)(int)
- char ∗(∗ **mprintf**)(const char ∗,...)
- int(∗ **open**)(const char ∗, sqlite3 ∗∗)
- int(∗ **open16**)(const void ∗, sqlite3 ∗∗)
- int(∗ **prepare**)(sqlite3 ∗, const char ∗, int, sqlite3_stmt ∗∗, const char ∗∗)
- int(∗ **prepare16**)(sqlite3 ∗, const void ∗, int, sqlite3_stmt ∗∗, const void ∗∗)
- void ∗(∗ **profile**)(sqlite3 ∗, void(∗)(void ∗, const char ∗, sqlite_uint64), void ∗)
- void(∗ **progress_handler**)(sqlite3 ∗, int, int(∗)(void ∗), void ∗)
- void ∗(∗ **realloc**)(void ∗, int)
- int(∗ **reset**)(sqlite3_stmt ∗pStmt)
- void(∗ **result_blob**)(sqlite3_context ∗, const void ∗, int, void(∗)(void ∗))
- void(∗ **result_double**)(sqlite3_context ∗, double)
- void(∗ **result_error**)(sqlite3_context ∗, const char ∗, int)
- void(∗ **result_error16**)(sqlite3_context ∗, const void ∗, int)
- void(∗ **result_int**)(sqlite3_context ∗, int)
- void(∗ **result_int64**)(sqlite3_context ∗, sqlite_int64)
- void(∗ **result_null**)(sqlite3_context ∗)
- void(∗ **result_text**)(sqlite3_context ∗, const char ∗, int, void(∗)(void ∗))
- void(∗ **result_text16**)(sqlite3_context ∗, const void ∗, int, void(∗)(void ∗))

- void(∗ **result_text16be**)(sqlite3_context ∗, const void ∗, int, void(∗)(void ∗))
- void(∗ **result_text16le**)(sqlite3_context ∗, const void ∗, int, void(∗)(void ∗))
- void(∗ **result_value**)(sqlite3_context ∗, sqlite3_value ∗)
- void ∗(∗ **rollback_hook**)(sqlite3 ∗, void(∗)(void ∗), void ∗)
- int(∗ **set_authorizer**)(sqlite3 ∗, int(∗)(void ∗, int, const char ∗, const char ∗, const char ∗, const char ∗), void ∗)
- void(∗ **set_auxdata**)(sqlite3_context ∗, int, void ∗, void(∗)(void ∗))
- char ∗(∗ **snprintf**)(int, char ∗, const char ∗,...)
- int(∗ **step**)(sqlite3_stmt ∗)
- int(∗ **table_column_metadata**)(sqlite3 ∗, const char ∗, const char ∗, const char ∗, char const ∗∗, char const ∗∗, int ∗, int ∗, int ∗)
- void(∗ **thread_cleanup**)(void)
- int(∗ **total_changes**)(sqlite3 ∗)
- void ∗(∗ **trace**)(sqlite3 ∗, void(∗xTrace)(void ∗, const char ∗), void ∗)
- int(∗ **transfer_bindings**)(sqlite3_stmt ∗, sqlite3_stmt ∗)
- void ∗(∗ **update_hook**)(sqlite3 ∗, void(∗)(void ∗, int, char const ∗, char const ∗, sqlite_int64), void ∗)
- void ∗(∗ **user_data**)(sqlite3_context ∗)
- const void ∗(∗ **value_blob**)(sqlite3_value ∗)
- int(∗ **value_bytes**)(sqlite3_value ∗)
- int(∗ **value_bytes16**)(sqlite3_value ∗)
- double(∗ **value_double**)(sqlite3_value ∗)
- int(∗ **value_int**)(sqlite3_value ∗)
- sqlite_int64(∗ **value_int64**)(sqlite3_value ∗)
- int(∗ **value_numeric_type**)(sqlite3_value ∗)
- const unsigned char ∗(∗ **value_text**)(sqlite3_value ∗)
- const void ∗(∗ **value_text16**)(sqlite3_value ∗)
- const void ∗(∗ **value_text16be**)(sqlite3_value ∗)
- const void ∗(∗ **value_text16le**)(sqlite3_value ∗)
- int(∗ **value_type**)(sqlite3_value ∗)
- char ∗(∗ **vmprintf**)(const char ∗, va_list)
- int(∗ **overload_function**)(sqlite3 ∗, const char ∗zFuncName, int nArg)
- int(∗ **prepare_v2**)(sqlite3 ∗, const char ∗, int, sqlite3_stmt ∗∗, const char ∗∗)
- int(∗ **prepare16_v2**)(sqlite3 ∗, const void ∗, int, sqlite3_stmt ∗∗, const void ∗∗)
- int(∗ **clear_bindings**)(sqlite3_stmt ∗)
- int(∗ **create_module_v2**)(sqlite3 ∗, const char ∗, const **sqlite3_module** ∗, void ∗, void(∗xDestroy)(void ∗))
- int(∗ **bind_zeroblob**)(sqlite3_stmt ∗, int, int)
- int(∗ **blob_bytes**)(sqlite3_blob ∗)
- int(∗ **blob_close**)(sqlite3_blob ∗)
- int(∗ **blob_open**)(sqlite3 ∗, const char ∗, const char ∗, const char ∗, sqlite3_int64, int, sqlite3_blob ∗∗)
- int(∗ **blob_read**)(sqlite3_blob ∗, void ∗, int, int)
- int(∗ **blob_write**)(sqlite3_blob ∗, const void ∗, int, int)
- int(∗ **create_collation_v2**)(sqlite3 ∗, const char ∗, int, void ∗, int(∗)(void ∗, int, const void ∗, int, const void ∗), void(∗)(void ∗))
- int(∗ **file_control**)(sqlite3 ∗, const char ∗, int, void ∗)
- sqlite3_int64(∗ **memory_highwater**)(int)
- sqlite3_int64(∗ **memory_used**)(void)
- sqlite3_mutex ∗(∗ **mutex_alloc**)(int)

- void(∗ **mutex_enter**)(sqlite3_mutex ∗)
- void(∗ **mutex_free**)(sqlite3_mutex ∗)
- void(∗ **mutex_leave**)(sqlite3_mutex ∗)
- int(∗ **mutex_try**)(sqlite3_mutex ∗)
- int(∗ **open_v2**)(const char ∗, sqlite3 ∗∗, int, const char ∗)
- int(∗ **release_memory**)(int)
- void(∗ **result_error_nomem**)(sqlite3_context ∗)
- void(∗ **result_error_toobig**)(sqlite3_context ∗)
- int(∗ **sleep**)(int)
- void(∗ **soft_heap_limit**)(int)
- **sqlite3_vfs** ∗(∗ **vfs_find**)(const char ∗)
- int(∗ **vfs_register**)(**sqlite3_vfs** ∗, int)
- int(∗ **vfs_unregister**)(**sqlite3_vfs** ∗)
- int(∗ **xthreadsafe**)(void)
- void(∗ **result_zeroblob**)(sqlite3_context ∗, int)
- void(∗ **result_error_code**)(sqlite3_context ∗, int)
- int(∗ **test_control**)(int,...)
- void(∗ **randomness**)(int, void ∗)
- sqlite3 ∗(∗ **context_db_handle**)(sqlite3_context ∗)
- int(∗ **extended_result_codes**)(sqlite3 ∗, int)
- int(∗ **limit**)(sqlite3 ∗, int, int)
- sqlite3_stmt ∗(∗ **next_stmt**)(sqlite3 ∗, sqlite3_stmt ∗)
- const char ∗(∗ **sql**)(sqlite3_stmt ∗)
- int(∗ **status**)(int, int ∗, int ∗, int)

2.143.1 Detailed Description

Definition at line 38 of file sqlite3ext.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3ext.h

2.144 sqlite3_file Struct Reference

Collaboration diagram for sqlite3_file:

Public Attributes

- **sqlite3_io_methods** ∗ **pMethods**

2.144.1 Detailed Description

Definition at line 491 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.145 sqlite3_index_info Struct Reference

Collaboration diagram for sqlite3_index_info:

Public Attributes

- int **nConstraint**
- **sqlite3_index_info::sqlite3_index_constraint** * **aConstraint**
- int **nOrderBy**
- **sqlite3_index_info::sqlite3_index_orderby** * **aOrderBy**
- **sqlite3_index_info::sqlite3_index_constraint_usage** * **aConstraintUsage**
- int **idxNum**
- char * **idxStr**
- int **needToFreeIdxStr**
- int **orderByConsumed**
- double **estimatedCost**

Classes

- struct **sqlite3_index_constraint**
- struct **sqlite3_index_constraint_usage**
- struct **sqlite3_index_orderby**

2.145.1 Detailed Description

Definition at line 4243 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.146 sqlite3_index_info::sqlite3_index_constraint Struct Reference

Collaboration diagram for sqlite3_index_info::sqlite3_index_constraint:

Public Attributes

- int **iColumn**
- unsigned char **op**
- unsigned char **usable**
- int **iTermOffset**

2.146.1 Detailed Description

Definition at line 4246 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.147 sqlite3_index_info::sqlite3_index_constraint_usage Struct Reference

Collaboration diagram for sqlite3_index_info::sqlite3_index_constraint_usage:

Public Attributes

- int **argvIndex**
- unsigned char **omit**

2.147.1 Detailed Description

Definition at line 4258 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.148 sqlite3_index_info::sqlite3_index_orderby Struct Reference

Collaboration diagram for sqlite3_index_info::sqlite3_index_orderby:

Public Attributes

- int **iColumn**
- unsigned char **desc**

2.148.1 Detailed Description

Definition at line 4253 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.149 sqlite3_io_methods Struct Reference

Collaboration diagram for sqlite3_io_methods:

Public Attributes

- int **iVersion**
- int(∗ **xClose**)(**sqlite3_file** ∗)
- int(∗ **xRead**)(**sqlite3_file** ∗, void ∗, int iAmt, sqlite3_int64 iOfst)
- int(∗ **xWrite**)(**sqlite3_file** ∗, const void ∗, int iAmt, sqlite3_int64 iOfst)
- int(∗ **xTruncate**)(**sqlite3_file** ∗, sqlite3_int64 size)
- int(∗ **xSync**)(**sqlite3_file** ∗, int flags)
- int(∗ **xFileSize**)(**sqlite3_file** ∗, sqlite3_int64 ∗pSize)
- int(∗ **xLock**)(**sqlite3_file** ∗, int)
- int(∗ **xUnlock**)(**sqlite3_file** ∗, int)
- int(∗ **xCheckReservedLock**)(**sqlite3_file** ∗, int ∗pResOut)
- int(∗ **xFileControl**)(**sqlite3_file** ∗, int op, void ∗pArg)
- int(∗ **xSectorSize**)(**sqlite3_file** ∗)
- int(∗ **xDeviceCharacteristics**)(**sqlite3_file** ∗)

2.149.1 Detailed Description

Definition at line 583 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.150 sqlite3_mem_methods Struct Reference

Collaboration diagram for sqlite3_mem_methods:

Public Attributes

- void ∗(∗ **xMalloc**)(int)
- void(∗ **xFree**)(void ∗)
- void ∗(∗ **xRealloc**)(void ∗, int)
- int(∗ **xSize**)(void ∗)
- int(∗ **xRoundup**)(int)
- int(∗ **xInit**)(void ∗)
- void(∗ **xShutdown**)(void ∗)
- void ∗ **pAppData**

2.150.1 Detailed Description

Definition at line 976 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.151 sqlite3_module Struct Reference

Collaboration diagram for sqlite3_module:

Public Attributes

- int **iVersion**
- int(∗ **xCreate**)(sqlite3 ∗, void ∗pAux, int argc, const char ∗const ∗argv, **sqlite3_vtab** ∗∗ppVTab, char ∗∗)
- int(∗ **xConnect**)(sqlite3 ∗, void ∗pAux, int argc, const char ∗const ∗argv, **sqlite3_vtab** ∗∗ppVTab, char ∗∗)
- int(∗ **xBestIndex**)(**sqlite3_vtab** ∗pVTab, **sqlite3_index_info** ∗)
- int(∗ **xDisconnect**)(**sqlite3_vtab** ∗pVTab)
- int(∗ **xDestroy**)(**sqlite3_vtab** ∗pVTab)
- int(∗ **xOpen**)(**sqlite3_vtab** ∗pVTab, **sqlite3_vtab_cursor** ∗∗ppCursor)
- int(∗ **xClose**)(**sqlite3_vtab_cursor** ∗)
- int(∗ **xFilter**)(**sqlite3_vtab_cursor** ∗, int idxNum, const char ∗idxStr, int argc, sqlite3_value ∗∗argv)
- int(∗ **xNext**)(**sqlite3_vtab_cursor** ∗)
- int(∗ **xEof**)(**sqlite3_vtab_cursor** ∗)
- int(∗ **xColumn**)(**sqlite3_vtab_cursor** ∗, sqlite3_context ∗, int)
- int(∗ **xRowid**)(**sqlite3_vtab_cursor** ∗, sqlite3_int64 ∗pRowid)
- int(∗ **xUpdate**)(**sqlite3_vtab** ∗, int, sqlite3_value ∗∗, sqlite3_int64 ∗)
- int(∗ **xBegin**)(**sqlite3_vtab** ∗pVTab)
- int(∗ **xSync**)(**sqlite3_vtab** ∗pVTab)
- int(∗ **xCommit**)(**sqlite3_vtab** ∗pVTab)
- int(∗ **xRollback**)(**sqlite3_vtab** ∗pVTab)
- int(∗ **xFindFunction**)(**sqlite3_vtab** ∗pVtab, int nArg, const char ∗zName, void(∗∗pxFunc)(sqlite3_context ∗, int, sqlite3_value ∗∗), void ∗∗ppArg)
- int(∗ **xRename**)(**sqlite3_vtab** ∗pVtab, const char ∗zNew)

2.151.1 Detailed Description

Definition at line 4162 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.152 sqlite3_mutex_methods Struct Reference

Collaboration diagram for sqlite3_mutex_methods:

Public Attributes

- int(∗ **xMutexInit**)(void)
- int(∗ **xMutexEnd**)(void)
- sqlite3_mutex ∗(∗ **xMutexAlloc**)(int)
- void(∗ **xMutexFree**)(sqlite3_mutex ∗)
- void(∗ **xMutexEnter**)(sqlite3_mutex ∗)
- int(∗ **xMutexTry**)(sqlite3_mutex ∗)
- void(∗ **xMutexLeave**)(sqlite3_mutex ∗)
- int(∗ **xMutexHeld**)(sqlite3_mutex ∗)
- int(∗ **xMutexNotheld**)(sqlite3_mutex ∗)

2.152.1 Detailed Description

Definition at line 4815 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.153 sqlite3_pcache_methods Struct Reference

Collaboration diagram for sqlite3_pcache_methods:

Public Attributes

- void ∗ **pArg**
- int(∗ **xInit**)(void ∗)
- void(∗ **xShutdown**)(void ∗)
- sqlite3_pcache ∗(∗ **xCreate**)(int szPage, int bPurgeable)
- void(∗ **xCachesize**)(sqlite3_pcache ∗, int nCachesize)
- int(∗ **xPagecount**)(sqlite3_pcache ∗)
- void ∗(∗ **xFetch**)(sqlite3_pcache ∗, unsigned key, int createFlag)
- void(∗ **xUnpin**)(sqlite3_pcache ∗, void ∗, int discard)
- void(∗ **xRekey**)(sqlite3_pcache ∗, void ∗, unsigned oldKey, unsigned newKey)
- void(∗ **xTruncate**)(sqlite3_pcache ∗, unsigned iLimit)
- void(∗ **xDestroy**)(sqlite3_pcache ∗)

2.153.1 Detailed Description

Definition at line 5291 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.154 sqlite3_vfs Struct Reference

Collaboration diagram for sqlite3_vfs:

Public Attributes

- int **iVersion**
- int **szOsFile**
- int **mxPathname**
- **sqlite3_vfs** ∗ **pNext**
- const char ∗ **zName**
- void ∗ **pAppData**
- int(∗ **xOpen**)(**sqlite3_vfs** ∗, const char ∗**zName**, **sqlite3_file** ∗, int flags, int ∗pOutFlags)
- int(∗ **xDelete**)(**sqlite3_vfs** ∗, const char ∗**zName**, int syncDir)
- int(∗ **xAccess**)(**sqlite3_vfs** ∗, const char ∗**zName**, int flags, int ∗pResOut)
- int(∗ **xFullPathname**)(**sqlite3_vfs** ∗, const char ∗**zName**, int nOut, char ∗zOut)
- void ∗(∗ **xDlOpen**)(**sqlite3_vfs** ∗, const char ∗zFilename)
- void(∗ **xDlError**)(**sqlite3_vfs** ∗, int nByte, char ∗zErrMsg)
- void(∗(∗ **xDlSym**)(**sqlite3_vfs** ∗, void ∗, const char ∗zSymbol))(void)
- void(∗ **xDlClose**)(**sqlite3_vfs** ∗, void ∗)
- int(∗ **xRandomness**)(**sqlite3_vfs** ∗, int nByte, char ∗zOut)
- int(∗ **xSleep**)(**sqlite3_vfs** ∗, int microseconds)
- int(∗ **xCurrentTime**)(**sqlite3_vfs** ∗, double ∗)
- int(∗ **xGetLastError**)(**sqlite3_vfs** ∗, int, char ∗)

2.154.1 Detailed Description

Definition at line 762 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.155 sqlite3_vtab Struct Reference

Collaboration diagram for sqlite3_vtab:

Public Attributes

- const **sqlite3_module** ∗ **pModule**
- int **nRef**
- char ∗ **zErrMsg**

2.155.1 Detailed Description

Definition at line 4339 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.156 sqlite3_vtab_cursor Struct Reference

Collaboration diagram for sqlite3_vtab_cursor:

Public Attributes

- **sqlite3_vtab** ∗ **pVtab**

2.156.1 Detailed Description

Definition at line 4364 of file sqlite3.h.

The documentation for this struct was generated from the following file:

- Recommender/Database/include/sqlite3.h

2.157 transT Struct Reference

Collaboration diagram for transT:

Public Attributes

- ::**br::pucrio::telemidia::ginga::ncl::model::presentation::FormatterRegion** ∗ **fr**
- Transition ∗ **t**

2.157.1 Detailed Description

Definition at line 229 of file FormatterRegion.h.

The documentation for this struct was generated from the following file:

- gingancl-cpp/include/model/FormatterRegion.h

2.158 Trie Class Reference

`#include <Trie.hpp>`

Collaboration diagram for Trie:

Public Member Functions

- **Trie** (const unsigned long init_counter)
- const **Trie** ∗ **is_included** (const set< itemtype > &an_itemset, set< itemtype >::const_iterator item_it) const

  Decide se o itemset será ou não incluído na árvore.

- void **find_candidate** (vector< itemtype >::const_iterator it_basket_upper_bound, const itemtype distance_from_candidate, vector< itemtype >::const_iterator it_basket, const unsigned long counter_incr=1)

  Incrementa o contator dos itemsets.

- void **delete_infrequent** (const double min_occurrence, const itemtype distance_from_candidate)

  Apaga as arvores que contém os itens que ocorrem poucas vezes.

- void **show_content_preorder** () const

  exibe a árvore em pré-ordem

Private Member Functions

- void **add_empty_state** (const itemtype item, const unsigned long init_counter=0)

  adiciona um nó vazio na árvore

Private Attributes

- unsigned long **counter**

  contator que armazena o número de nós na trie (Árvore)

- vector< **Edge** > **edgevector**
- itemtype **maxpath**

  armazena o tamanho do maior caminho até os nós folha da árvore (trie) inicia no nó raiz

Friends

- class **Apriori_Trie**

2.158.1 Detailed Description

A classe trie é uma abstração para uma estrutura de dados recursivas. Cada nó raíz representa um itemset.

Definition at line 93 of file Trie.hpp.

2.158.2 Member Data Documentation

2.158.2.1 vector<Edge> edgevector [private]

edgevector armazena os nós raízes das trie

edgevector é organizado. Em linhas gerais é um vector de tries que são estruturas de dados do tipo árvores

Definition at line 142 of file Trie.hpp.

The documentation for this class was generated from the following file:

- Recommender/MiningAlgorithm/include/apriori23/Trie.hpp

2.159 XStr Class Reference

Collaboration diagram for XStr:

Public Member Functions

- **XStr** (const char ∗const toTranscode)
- ∼**XStr** ()
- const XMLCh ∗ **unicodeForm** () const

Private Attributes

- XMLCh ∗ **fUnicodeForm**

2.159.1 Detailed Description

Definition at line 52 of file createXML.h.

The documentation for this class was generated from the following file:

- Recommender/ExportXML/include/createXML.h