

Entwurfsbeschreibung für Fremdprojekt

1. Wie wird die Model-View-Controller-Architektur umgesetzt?

Bei näherer Betrachtung der einzelnen Klassen und deren Funktionen ergibt sich, dass die MVC-Architektur in dem SyncML-Paket des Horde-Frameworks nicht umgesetzt ist.

2. Wie ist das Packages und die API strukturiert?

Das SyncML-Package ist ein aus dem Horde-Framework heraus gelöstes Modul und benötigt daher nur noch wenige Klassen. Es ist streng objektorientiert gestaltet, wobei es hier und da ein paar Funktionen ohne zugehörige Klasse gibt.

Siehe UML-Übersicht Seite 3

Der Einstiegspunkt, wo die Verarbeitung von SyncML-Anfragen stattfinden, ist die Funktion *SyncML_HandleRequest*. Sie bekommt den gesendeten Request, die Art (ob XML oder WBXML) der gesendeten Daten sowie die Zugangsdaten für das Backend. Ein Backend ist eine spezielle Datei, welche sich um die Verwaltung der Daten kümmert. In unserem Fall wäre das MySQL.

In der Funktion *SyncML_HandleRequest* wird zuerst ein Backend ausgewählt, welches auf die Datenbestände zugreifen kann. Danach wird ein Objekt von der Klasse *SyncML_ContentHandler* instantiiert und die Methode *process* aufgerufen. In ihr verzweigen sich nun die genutzten Klassen je nachdem, welche Daten interpretiert und verarbeitet werden müssen.

Die API ist auf Dateiebene folgendermaßen strukturiert:

- **Hauptordner:**
 - allgemeiner Kram

- **Unterordner Backend**
 - Hier befinden sich die verschiedenen Backends. Zum Zeitpunkt der Erstellung der Entwurfsbeschreibung gibt es eines für das Horde-Framework und ein einfaches SQL-Backend.

- **Unterordner Command**

- Hier befinden sich die sogenannten Commands, welche laut SyncML-Spezifikation für eine Synchronisierung benötigt werden. Sie stellen Klassen dar, welche u.a. die geschickten Daten des Clients bearbeiten und Informationen an den SyncML-Server weiterreichen oder das Ergebnis einer Suche an den Client zurücksenden.

- **Unterordner Device**

- Dieser Ordner enthält verschiedene Klassen, welche für je einen Client eines bestimmten Handgerätes die Funktionalität zur Verfügung stellen. U.a. befinden sich hier Klassen für die Arbeit mit Nokia-, P800- oder Synthesis-SyncML-Clients.

Die API ist auf Softwareebene folgendermaßen strukturiert:

- die API ist objektorientiert aufgebaut
- sowohl Klassenhierarchien, als auch ab gekapselte Klassen sind vorhanden

3. Wie kann das Projekt durch die Implementierung eines eigenen Triple-Store Backends auf Basis der Erfurt API und einer OntoWiki Component Extension erweitert werden? Beschreiben Sie dies anhand des Modells.

Die bestehende SyncML-Komponente des Horde-Frameworks benötigt einerseits eine Schnittstelle für den Triple-Store des OntoWikis sowie eine geänderte Authentifizierungsmethode. Die Schnittstelle für den Triple-Store muss vCards verschiedenster Versionen (1.1, 1.2, etc.) lesen und in RDF/RDFs im OntoWiki ablegen können. Die SyncML-Horde-Authentifizierung muss dahin geändert, dass eine Anmeldung im OntoWiki möglich ist. Parallel zu den Datenbeständen, muss gespeichert werden, wann die einzelnen Personendaten geändert wurden.

Anhang:

- SyncML Klassendiagramm Seite 3
- SyncML-Backend Klassendiagramm Seite 4
- SyncML-Command Klassendiagramm Seite 5
- SyncML-Device Klassendiagramm Seite 6

SyncML_Command
+ cmdName : undef
+ cmdID : undef
+ stack : undef
+ chars : undef
+ outputHandler : undef
+ SyncML_Command(outputHandler : undef)
+ startElement(uri : undef, element : undef, attrs : undef)
+ endElement(uri : undef, element : undef)
+ characters(str : undef)
+ getCommandName()
+ handleCommand(debug : undef)
+ factory(command : undef, outputHandler : undef)

SyncML_Property
+ ValEnum : undef
+ DataType : undef
+ Size : undef
+ DisplayName : undef
+ Params : undef

SyncML_PropertyParameter
+ ValEnum : undef
+ DataType : undef
+ Size : undef
+ DisplayName : undef

SyncML_DataStore
+ SourceRef : undef
+ DisplayName : undef
+ MaxGUIDSize : undef
+ Rx_Pref : undef
+ Size : undef
+ Tx_Pref : undef
+ Tx : undef
+ DSMem : undef
+ SyncCap : undef
+ getPreferredRXContentType()
+ getPreferredRXContentTypeVersion()

SyncML_ContentHandler
+ Stack : undef
+ chars : undef
+ currentCommand : undef
+ gotFinal : undef
+ xmlWriter : undef
+ wbxmlparser : undef
+ respURI : undef
+ debug : undef
+ SyncML_ContentHandler()
+ process(request : undef, contentType : undef, respURI : undef)
+ getOutput()
+ startElement(parser : undef, tag : undef, attributes : undef)
+ characters(parser : undef, chars : undef)
+ endElement(parser : undef, tag : undef)
+ splitURI(tag : undef)
+ startElement(uri : undef, element : undef, attrs : undef)
+ endElement(uri : undef, element : undef)
+ characters(str : undef)
+ handleHeader(hdr : undef)
+ handleCommand(cmd : undef)
+ handleEnd()

SyncML_State
+ sessionID : undef
+ messageID : undef
+ targetURI : undef
+ sourceURI : undef
+ version : undef
+ user : undef
+ authenticated : undef
+ uri : undef
+ uriMeta : undef
+ uriDevInf : undef
+ wbxml : undef
+ maxMsgSize : undef
+ _syncs : undef
+ serverChanges : undef
+ deviceDriver : undef
+ deviceInfo : undef
+ curSyncItem : undef
+ delayedFinal : undef
+ SyncML_State(sourceURI : undef, user : undef, sessionID : undef)
+ getVerDTD()
+ getDevInfURI()
+ getProtocolName()
+ setVersion(version : undef)
+ getURI()
+ getDevice()
+ setSync(target : undef, sync : undef)
+ getSync(target : undef)
+ getSyncs()
+ hasPendingSyncs()
+ getPendingSyncs()
+ isAllSyncsComplete()
+ handleFinal(output : undef, debug : undef)

SyncML_Sync
+ targetLocURI : undef
+ sourceLocURI : undef
+ syncType : undef
+ syncsSent : undef
+ syncsReceived : undef
+ expectingMapData : undef
+ state : undef
+ clientAnchorNext : undef
+ serverAnchorLast : undef
+ serverAnchorNext : undef
+ client_add_count : undef
+ client_replace_count : undef
+ client_delete_count : undef
+ client_addreplaces : undef
+ server_add_count : undef
+ server_replace_count : undef
+ server_delete_count : undef
+ errors : undef
+ server_adds : undef
+ server_replaces : undef
+ server_deletes : undef
+ server_task_adds : undef
+ SyncML_Sync(syncType : undef, serverURI : undef, clientURI : undef, serverAnchorLast : undef, serverAnchorNext : undef, clientAnchorNext : undef)
+ handleClientSyncItem(output : undef, item : undef)
+ createSyncOutput(output : undef)
+ retrieveChanges(syncDB : undef, adds : undef, replaces : undef, deletes : undef)
+ handleFinal(output : undef, debug : undef)
+ hasPendingElements()
+ addSyncReceived()
+ isComplete()
+ closeSync()
+ getServerLocURI()
+ getClientLocURI()
+ getClientAnchorNext()
+ getServerAnchorNext()
+ createUriMap(databaseURI : undef, cuid : undef, cuid : undef)
+ setServerChange(change : undef, id : undef)
+ setServerChange(change : undef, sid : undef, cid : undef)
+ unsetServerChange(change : undef, id : undef)
+ taskDbUriForCalendarDbUri(databaseURI : undef)

SyncML_XMLOutput
+ msg_CmdID : undef
+ output : undef
+ uri : undef
+ getOutput()
+ getOutputSize()
+ isWBXML()
+ singleton()
+ init(theOutputHandler : undef)
+ outputHeader(respURI : undef)
+ outputInit()
+ outputBodyStart()
+ outputFinal()
+ outputEnd()
+ outputStatus(cmdRef : undef, cmd : undef, data : undef, targetRef : undef, sourceRef : undef, syncAnchorNext : undef, syncAnchorLast : undef)
+ outputDevInf(cmdRef : undef)
+ writeDataStore(sourceRef : undef, mimeType : undef, version : undef, output : undef, additionalTypes : undef)
+ outputAlert(alertCode : undef, clientDB : undef, serverDB : undef, lastAnchor : undef, nextAnchor : undef)
+ outputGetDevInf()
+ outputSyncStart(clientLocURI : undef, serverLocURI : undef, contentType : undef, encodingType : undef, cuid : undef, cuid : undef)
+ outputSyncEnd()
+ singleEle(tag : undef, str : undef, uri : undef)

SyncML_Backend
+ logtext : undef
+ debugDir : undef
+ debugFiles : undef
+ logLevel : undef
+ charset : undef
+ user : undef
+ syncDeviceID : undef
+ backendMode : undef
+ SyncML_Backend(params : undef)
+ factory(driver : undef, params : undef)
+ setCharset(charset : undef)
+ getCharset()
+ getSyncDeviceID()
+ setUser(user : undef)
+ getUser()
+ setupState(state : undef)
+ sessionStart(syncDeviceID : undef, sessionID : undef, backendMode : undef)
+ sessionClose()
+ isValidDatabaseURI(databaseURI : undef)
+ getServerChanges(databaseURI : undef, from_ts : undef, to_ts : undef, adds : undef, mods : undef, dels : undef)
+ retrieveEntry(databaseURI : undef, cuid : undef, contentType : undef)
+ addEntry(databaseURI : undef, content : undef, contentType : undef, cuid : undef)
+ replaceEntry(databaseURI : undef, content : undef, contentType : undef, cuid : undef)
+ deleteEntry(databaseURI : undef, cuid : undef)
+ checkAuthentication(username : undef, credData : undef, credFormat : undef, credType : undef)
+ _checkAuthentication(username : undef, password : undef)
+ setAuthenticated(username : undef, credData : undef)
+ writeSyncAnchors(databaseURI : undef, clientAnchorNext : undef, serverAnchorNext : undef)
+ readSyncAnchors(databaseURI : undef)
+ createUriMap(databaseURI : undef, cuid : undef, cuid : undef, timestamp : undef)
+ eraseMap(databaseURI : undef)
+ logMessage(message : undef, file : undef, line : undef, priority : undef)
+ logFileType : undef, content : undef, wbxml : undef, sessionClose : undef)
+ close()
+ getCurrentTimeStamp()
+ testSetup(user : undef, pwd : undef)
+ testStart(user : undef)
+ testTearDown()
+ _normalize(databaseURI : undef)
+ getParameter(url : undef, parameter : undef, default : undef)

SyncML_SyncElement
+ contentType : undef
+ contentFormat : undef
+ content : undef
+ size : undef
+ cmdID : undef
+ elementType : undef
+ cuid : undef
+ responseCode : undef
+ sync : undef
+ SyncML_SyncElement(sync : undef, elementType : undef, cmdID : undef, size : undef)

```

+ _logText : undef
+ _debugDir : undef
+ _debugFiles : undef
+ _logLevel : undef
+ _charset : undef
+ _user : undef
+ _syncDeviceID : undef
+ _backendMode : undef
+ SyncML_Backend(params : undef)
+ factory(driver : undef, params : undef)
+ setCharset(charset : undef)
+ getCharset()
+ getSyncDeviceID()
+ setUser(user : undef)
+ getUser()
+ setupState(state : undef)
+ sessionStart(syncDeviceID : undef, sessionId : undef, backendMode : undef)
+ sessionClose()
+ isValidDatabaseURL(databaseURL : undef)
+ getServerChanges(databaseURL : undef, from_ts : undef, to_ts : undef, adds : undef, mods : undef, dels : undef)
+ retrieveEntry(databaseURL : undef, cuid : undef, contentType : undef)
+ addEntry(databaseURL : undef, content : undef, contentType : undef, cuid : undef)
+ replaceEntry(databaseURL : undef, content : undef, contentType : undef, cuid : undef)
+ deleteEntry(databaseURL : undef, cuid : undef)
+ checkAuthentication(username : undef, credData : undef, credFormat : undef, credType : undef)
+ _checkAuthentication(username : undef, password : undef)
+ writeSyncAnchors(databaseURL : undef, clientAnchorNext : undef, serverAnchorNext : undef)
+ readSyncAnchors(databaseURL : undef)
+ createUuidMap(databaseURL : undef, cuid : undef, timestamp : undef)
+ eraseMap(databaseURL : undef)
+ logMessage(message : undef, file : undef, line : undef, priority : undef)
+ logFileType : undef, content : undef, wbxmi : undef, sessionClose : undef)
+ close()
+ getCurrentTimeStamp()
+ testSetup(user : undef, pwd : undef)
+ testStart(user : undef)
+ testTearDown()
+ _normalize(databaseURL : undef)
+ getParameter(url : undef, parameter : undef, default : undef)

```

SyncML_Backend

```

+ _db : undef
+ SyncML_Backend_Horde(params : undef)
+ setCharset(charset : undef)
+ setUser(user : undef)
+ sessionStart(syncDeviceID : undef, sessionId : undef, backendMode : undef)
+ getServerChanges(databaseURL : undef, from_ts : undef, to_ts : undef, adds : undef, mods : undef, dels : undef)
+ retrieveEntry(databaseURL : undef, cuid : undef, contentType : undef)
+ addEntry(databaseURL : undef, content : undef, contentType : undef, cuid : undef)
+ replaceEntry(databaseURL : undef, content : undef, contentType : undef, cuid : undef)
+ deleteEntry(databaseURL : undef, cuid : undef)
+ _checkAuthentication(username : undef, password : undef)
+ setAuthenticated(username : undef, credData : undef)
+ writeSyncAnchors(databaseURL : undef, clientAnchorNext : undef, serverAnchorNext : undef)
+ readSyncAnchors(databaseURL : undef)
+ getUserAnchors(user : undef)
+ removeAnchor(user : undef, device : undef, database : undef)
+ createUuidMap(databaseURL : undef, cuid : undef, timestamp : undef)
+ _getCuid(databaseURL : undef, cuid : undef)
+ _getChangeTS(databaseURL : undef, cuid : undef)
+ eraseMap(databaseURL : undef)
+ logMessage(message : undef, file : undef, line : undef, priority : undef)
+ testSetup(user : undef, pwd : undef)
+ testStart(user : undef)
+ testTearDown()

```

SyncML_Backend_Sql

```

+ _db : undef
+ SyncML_Backend_Sql(params : undef)
+ isValidDatabaseURL(databaseURL : undef)
+ getServerChanges(databaseURL : undef, from_ts : undef, to_ts : undef, adds : undef, mods : undef, dels : undef)
+ retrieveEntry(databaseURL : undef, cuid : undef, contentType : undef)
+ addEntry(databaseURL : undef, content : undef, contentType : undef, cuid : undef)
+ replaceEntry(databaseURL : undef, content : undef, contentType : undef, cuid : undef)
+ deleteEntry(databaseURL : undef, cuid : undef)
+ _checkAuthentication(username : undef, password : undef)
+ setAuthenticated(username : undef, credData : undef)
+ writeSyncAnchors(databaseURL : undef, clientAnchorNext : undef, serverAnchorNext : undef)
+ readSyncAnchors(databaseURL : undef)
+ createUuidMap(databaseURL : undef, cuid : undef, timestamp : undef)
+ _getCuid(databaseURL : undef, cuid : undef)
+ _getCuid(databaseURL : undef, cuid : undef)
+ _getChangeTS(databaseURL : undef, cuid : undef)
+ eraseMap(databaseURL : undef)
+ close()
+ _generateID()
+ _checkForError(o : undef)
+ _trackDeletes(databaseURL : undef, currentSuids : undef)
+ _removeFromSuidList(databaseURL : undef, cuid : undef)
+ testSetup(user : undef, pwd : undef)
+ testStart(user : undef)
+ testTearDown()
+ addEntry_backend(user : undef, databaseURL : undef, content : undef, contentType : undef)
+ replaceEntry_backend(user : undef, databaseURL : undef, content : undef, contentType : undef)
+ deleteEntry_backend(user : undef, databaseURL : undef, cuid : undef)
+ _cleanUser(user : undef)

```



