ijsfontein

Pazzanistraat 10
Westergasfabriek
1014 DB Amsterdam
T. +31 (0)20 420 07 43
F. +31 (0)20 421 45 98
www.ijsfontein.nl

Functional specifications

Cyberdam 2.1

Place	Amsterdam
Date	10 April 2008
Author	Marc de Boer
Position	Technical Manager
Document	Functional specification Cyberdam 2.1 version 0.4.doc
Version	0.4

Inhoudsopgave

1 Introduction

This document describes the additions and changes to Cyberdam 2.0 that will result in version 2.1.

For the specifications of Cyberdam 2.0 please consult the separate documents, but please keep in mind that it is the actual product that we are referring to, not the specifications. This is important because not everything envisioned for 2.0 was realized and also not all aspects, particularly graphic and interaction design, were properly described beforehand.

The contents of this document is in parts a high-level description, not a full functional design. And specifically, there are no screen designs. We need to find out whether this will be necessary as most of the screens are similar to other existing ones.

1.1 General hints

- Cyberdam is a multi language application. Any texts generated by the system should have a multi language key. The key itself is in English and follows naming conventions.
- Any destructive action, e.g. delete, should be guarded by an are-you- sure dialog.
- All pages have a standard structure with title, breadcrumbs, intro text, help text, etc. and use a common set of stylesheets.

1.2 Change history

In version 0.2:

- Added that rights should be checked on import
- The URI prefix requirements were changed, since the existing objects already conflict with that scheme.
- The Wiki is canceled

In version 0.3:

- On import, ownership is set to current user
- Manifests no longer need be exported or imported

In version 0.4:

- If a playground object has an external URL the X,Y and OnMap fields can be disabled
- imported elements receive the status 'under construction'
- Overwrite on playground import is not possible. If name conflicts, should rename
- Roles without objects have no link on the game session page

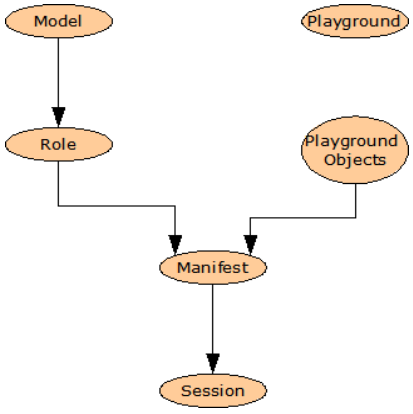
2 Overall goals

The two major goals of this upgrade are:

- The map in Cyberdam 2.0 is a good vehicle for some games, but it has quickly become apparent that there is no single map, or even map concept that satisfies everyone. Therefore we want to be able to handle multiple maps. This automatically leads to having a way of creating maps relatively easily.
- Cyberdam is used in many different locations, and so installed on many servers. We want a way to share data amongst servers.

3 In detail

3.1 Current situation



The data model can be abstracted as follows:

- There is only one playground. Therefore it doesn't really exist as an entity in any model.
- There is a list of playground objects, that implicitly belong to the one playground. There is no formal relationship between the two, only the conceptual understanding that the objects coordinates (x,y) mean a map position
- Models have a list of roles. Roles are not shared between models.
- The manifest binds playground objects to roles. Each role has exactly one object, and each object can have zero, one or more roles.
- A manifest can be used in any number of sessions.

3.2 Multiple playgrounds

We don't want a wild growth of playgrounds, and want to be able to do identify them uniquely. Therefore we will add an identifier to the playgrounds and let that be used in exchange of playgrounds, although they won't be centrally dispensed.

We also want to be able to use an external map. For that there will be external URI's.

Detailed changes are:

- Create a new list of playgrounds. Each playground has the following characteristics:
  - Name
  - Unique identifier. Arbitrary text string, for example 'Cybermaribo'.
  - Caption
  - Description. This is a large rich text, see existing 'Cyberdam intro' page
  - Version number
  - Date last modified
  - Last modified by
  - Link to an application that shows the map. This can be a Flash file, or any

arbitrary external link.
    - Status. A choice of “under construction”, “public” or “obsolete”. A manifest can only can refer to a playground when it has status "public".
- We need a screen listing all playgrounds. Elements:
    - A table with columns name, caption, identifier and date last modified
    - On every line buttons modify and delete
    - Add playground button
- A new screen modifying playground. Elements:
    - Entry for each of the fields
    - Cancel and save buttons
    - Modify objects button

  Only LCMS administrators and system administrators are allowed to modify playgrounds. Rights on playground objects continue to be allocated to playground-editors, LCMS administrators and system administrators.
- On the change object screen:
    - An extra field 'external URL', allowing entry of a arbitrary value. Idea is to enter references to maps on other site that we can link to.
    - Neither the coordinates nor the URL are mandatory. In this way, we can have playgrounds without a map.
    - If the external URL is entered, the X,Y and OnMap fields can be disabled.
- When creating or changing a manifest the user should select playground and object, not only object. Also, this should be made optional (extra option 'none' in the list) meaning that there is no map object.
- In the game session, the menu now contains three options: Cyberdam map, Cyberdam address book and Cyberdam Intro. When more that one playground exists, we will need a second level here: for each playground used in the session, there should be the playground name on level one and the three options mentioned above. If there is no playground, all options disappear.
- Also on the game session, if the role does not have a playground object associated with it, the link between brackets does not apply and is not shown.

3.3 Import and export

To share data between users, we need a function that exports elements to a form that can be shared digitally outside the system and read back elsewhere. In this case we will do this for games and for playgrounds.

Exports and imports should respect access privileges on the data.

Since there is no way of matching user names from the exporting to the importing system, all user relationships (creation, modification, etc) is set to the current user.
Imported models and playgrounds receive the status 'under construction'.

3.3.1 Export game

This function writes the coherent and closed set of data for a model with its list of roles to a single file. The function is called using a new button on each line of the game model list.

This generates output as XML in UTF-8 encoding. To able to handle future changes in the data structure we need to tag the XML files with a version number. Since models can have attachments, the function will export not only to an XML document but also a set of binaries. These will be wrapped up in a zip file (no folders) and offered for saving as a single entity.

Effectively, the browsers offers the user a 'save/cancel' dialog, so the file can easily be stored locally.

3.3.2 Import game

Obviously the converse of the export, this reads in the model with the corresponding roles. This is called by a button on the model list that calls the browser to open a file selection dialog. Any file can be selected.

When reading in the model is created as new. No attempt is made to link or merge the new data with existing models.

On completion of an import a summary is shown with the count of elements of each type.

3.3.3 Export playground

This function writes the playground with objects to a single file. Similarly to games, the function is called with a button on the playground list.

Since playgrounds can have a Flash file and playground objects have pictures, the function will export not only to an XML document but also a set of binaries. These will be wrapped up in a zip file (no folders) and offered for saving as a single entity. The XML is also version tagged.

3.3.4 Import playground

On importing the playground the unique identifier is checked. If a playground with that id already exists, a dialog screen is shown with a text field that allows entering a different name. The text field is pre-filled with the current name. There are two buttons, ok to confirm and import, and cancel to abort the whole operation.

On completion of an import a summary is shown with all playground data (except the description) and the number of playground objects.

3.4 Link to the map

To bring the map more into the game, we add a link to the map to the playground object

screen in the session. This should link directly to the object. If a URL has been entered, that will be the link. If only coordinates are entered this should link to the Flash map showing the selected object in the center. This means we need to change the Flash code so that it:

- accepts coordinates on the URL
- automatically centers around that point with the middle zoom factor active
- the info panel for the object is open

If neither URL nor coordinates are entered, the link does not exist. Also, if the object is not marked as 'visible on the map', the link is not shown.