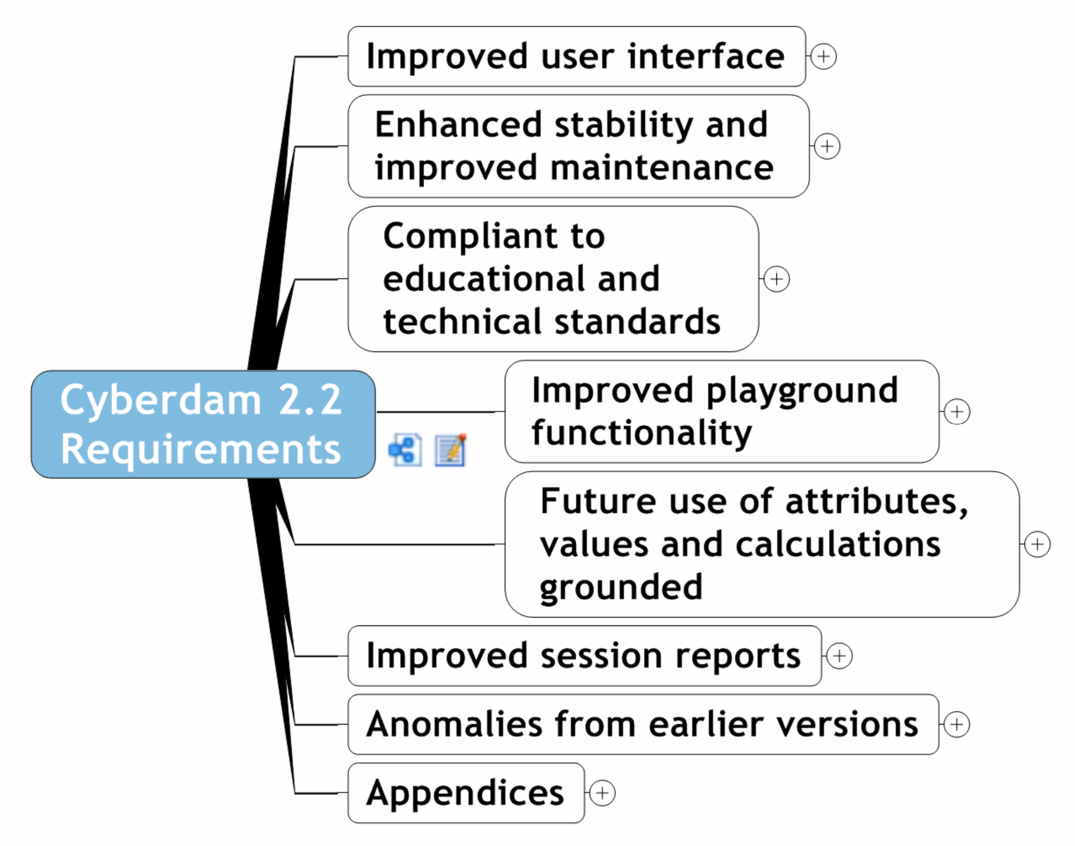
Stichting RechtenOnline

Project Leren in een Virtuele Wereld

Cyberdam 2.2 Requirements


Cyberdam 2.2 Requirements
Improved user interface
Enhanced stability and improved maintenance
Compliant to educational and technical standards
Improved playground functionality
Future use of attributes, values and calculations grounded
Improved session reports
Anomalies from earlier versions
Appendices

3rd draft 16 June 2008 by Pieter van der Hijden

Projectleiding:

Diny Peters
Pieter van der Hijden

Projectpartners:

E-Merge
Hogeschool Rotterdam
Hogeschool Utrecht
Strathclyde University Glasgow
TU-Delft
Universiteit Leiden
Thieme Meulenhoff
IJsfontein

1 Improved user interface

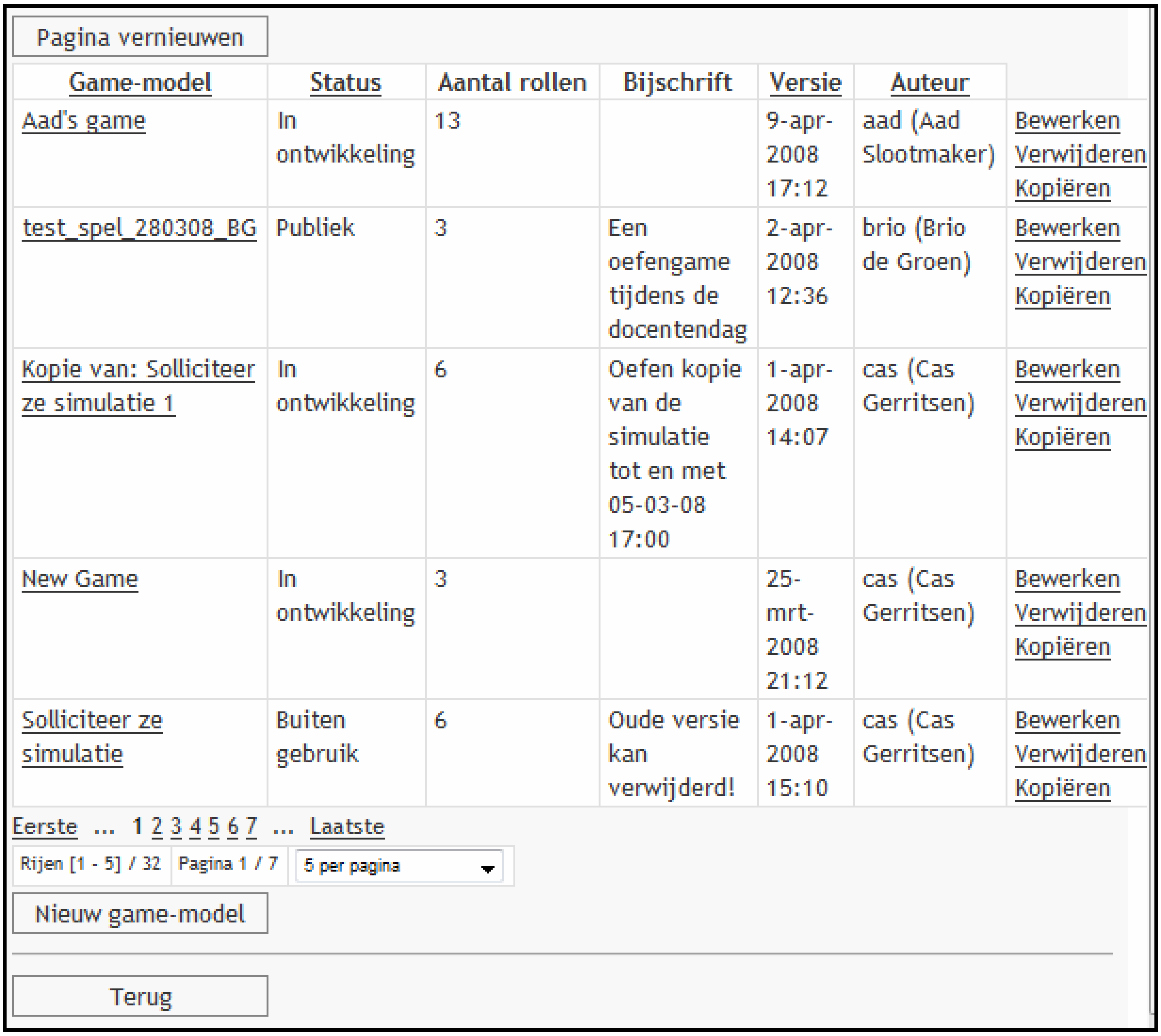
1.1 Organise critical walkthrough

During a half-day meeting with the client, IJsfontein will organise a critical walkthrough of the program from usability and user interface point of view.

We expect that the outcomes of this meeting are suggestions for improvements/changes of the Cyberdam-application. They may be subdivided into three categories:

- Quick wins that can be implemented relatively simply as part of the Cyberdam 2.2 effort, like:
  - Harmonise use of buttons, hyperlinks and icons
  - Improve the create/edit manifest user interface
  - Improve the create/edit game session user interface
  - Use rich text for e-mail notifications
- Improvements/changes that take more time (and need more budget); they will be candidates to be included in Cyberdam 2.3.
- Graphical design comments; they will be considered during the specification of Cyberdam 2.3.

1.2 Harmonise use of buttons, hyperlinks and icons



Game-model	Status	Aantal rollen	Bijschrift	Versie	Auteur	
Aad's game	In ontwikkeling	13		9-apr-2008 17:12	aad (Aad Slootmaker)	Bewerken Verwijderen Kopiëren
test_spel_280308_BG	Publiek	3	Een oefengame tijdens de docentendag	2-apr-2008 12:36	brio (Brio de Groen)	Bewerken Verwijderen Kopiëren
Kopie van: Solliciteer ze simulatie 1	In ontwikkeling	6	Oefen kopie van de simulatie tot en met 05-03-08 17:00	1-apr-2008 14:07	cas (Cas Gerritsen)	Bewerken Verwijderen Kopiëren
New Game	In ontwikkeling	3		25-mrt-2008 21:12	cas (Cas Gerritsen)	Bewerken Verwijderen Kopiëren
Solliciteer ze simulatie	Buiten gebruik	6	Oude versie kan verwijderd!	1-apr-2008 15:10	cas (Cas Gerritsen)	Bewerken Verwijderen Kopiëren

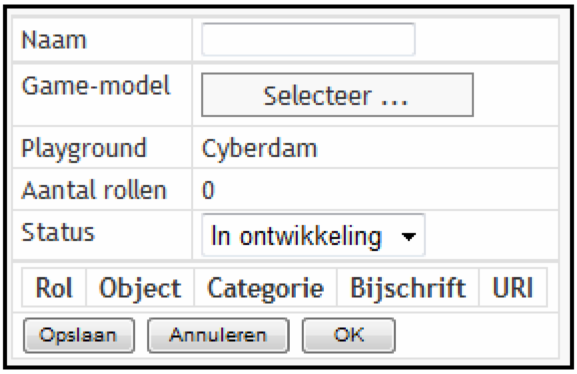
Eerste ... 1 2 3 4 5 6 7 ... Laatste

Rijen [1 - 5] / 32 Pagina 1 / 7 5 per pagina

At this moment some actions are triggered by clicking a hyperlink, other actions by clicking a button. We need to follow a single philosophy here.

To make the interface less cluttered with texts, action hyperlinks maybe replaced by little icons (plus a hovering text).

1.3 Improve the create/edit manifest user interface



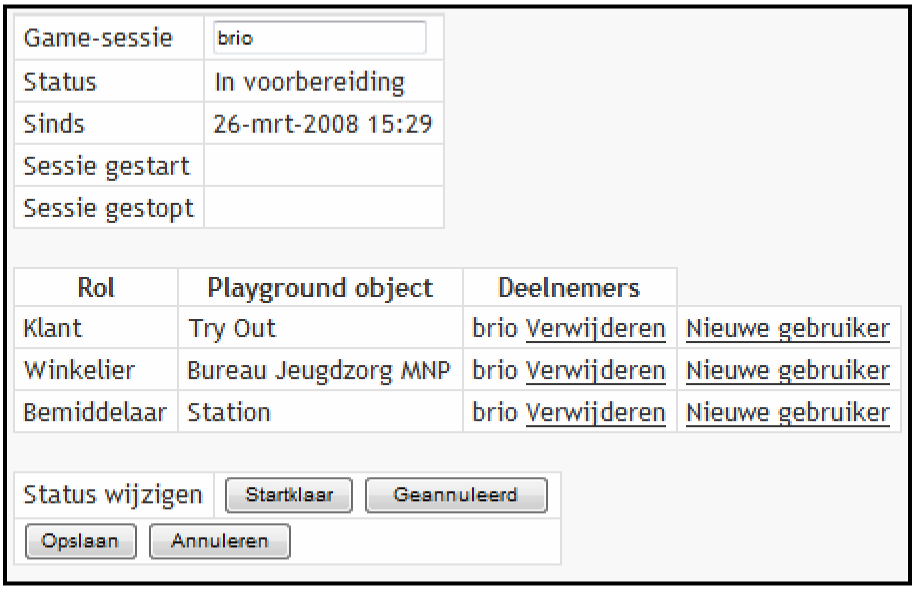
Naam	
Game-model	Selecteer ...
Playground	Cyberdam
Aantal rollen	0
Status	In ontwikkeling

Rol	Object	Categorie	Bijschrift	URI

Comments:

- The screenshot above comes from Cyberdam 2.0 and may not reflect the actual situation in Cyberdam 2.1.
- The function of the three buttons is unclear.
- Once a game model has been selected it cannot be changed anymore. Suggestion: Accept a looser way of filling in the manifest. Add a validate button to check its consistency and give warnings. Imply a validation when the user is trying to save. Only accept saving with status "under development" as long as validation does not return OK.

1.4 Improve the create/edit game session user interface



Game-sessie	brio
Status	In voorbereiding
Sinds	26-mrt-2008 15:29
Sessie gestart	
Sessie gestopt	

Rol	Playground object	Deelnemers	
Klant	Try Out	brio Verwijderen	Nieuwe gebruiker
Winkelier	Bureau Jeugdzorg MNP	brio Verwijderen	Nieuwe gebruiker
Bemiddelaar	Station	brio Verwijderen	Nieuwe gebruiker

Comments:

- Four different buttons to end this dialogue.
- Two locations dealing with the session status.

1.5 Use rich text for e-mail notifications

When a rich text is used as e-mail notification message, the message is sent as visible HTML. Problem: how to handle styles.

1.6 Give examples of multimedia use in game models

The FCK editor used to enter instructions and messages offers the opportunity to include multimedia. Some examples of its potential are needed.

1.7 Realise some quick wins (outcomes of critical walkthrough)

Reserve some time to realise some quick wins.

2 Enhanced stability and improved maintenance

2.1 Give game models an inspect option (read-only)

It makes sense to have the option to inspect a game model, apart from the option to edit it.

When a game model is public, all users with privilege = game author may inspect it. Only the owner (and LCMS admin and system admin) may edit it.

Even for users who are allowed to edit the game model, it would be welcome to give them the inspect function as editing (with autosaving) easily may result in unintended changes of the model.

2.2 Give manifests an inspect option (read-only)

It makes sense to have the option to inspect a game manifest, apart from the option to edit it.

When a game manifest is public, all users with privilege = game manifest composer may inspect it. Only the owner (and LCMS admin and system admin) may edit it.

Even for users who are allowed to edit the game model, it would be welcome to give them the inspect function as editing (with autosaving) easily may result in unintended changes of the manifest.

2.3 Give playgrounds an inspect option (read-only)

It makes sense to have the option to inspect a playground object, apart from the option to edit it.

When a playground object is public, all users with privilege = playground editor may inspect it. Only the owner (and LCMS admin and system admin) may edit it.

2.4 Make finished sessions read-only

Once a game session reaches its end, either a normal end (status=finished) or an abnormal end (status=aborted). participants should no longer be able to continue their play. It should be possible for them to inspect their session home page, see the messages, read the files and any uncompleted tasks. However, it should be impossible to send new messages, upload new files, delete or undelete messages or files or to complete any uncompleted task.

2.5 Make it possible to remove session data

At this moment deleting a game session results in setting a non started game session to status=cancelled or setting a started game session to status=aborted. However, it should be possible to really delete a session including all data involved (uploaded files, messages, etc.). Of course this delete operation should be guarded by a verbose are-you-sure dialogue.

2.6 Make it possible to export, import, create and delete language packs

At this moment the applications works with two language packs. It is required to be able to add a new language pack, to edit its description, to delete an existing one, to export a language pack and to import one. Importing implies overwriting an existing pack (after an are-you-sure dialogue).

For each language pack it must be clear from which other language pack it will inherit texts in the case the current pack cannot provide them. If the other pack has not been specified, the application will display the multilanguage key between brackets.

3 Compliant to educational and technical standards

3.1 Educational

3.1.1 Add metadata to game model

The game model form (metadata page) should contain the following fields:

- Name (existing),
- Caption (existing),
- URI (new),
- Language (new),
- Description (this is a new field, the existing "description" field will be renamed to "background),
- Keywords (new),
- Contribution (new),
- Size (new),
- Background (existing field, previously known as "description")
- Status (existing).

See the Appendices for further details.

3.1.2 Export game model meta data

To add game models to online catalogues of learning objects, a valid XML file with metadata is needed.

The application will generate a standardised metadata file based on the metadata entered by the user. A few known characteristics of the game model may be added automatically.

The application will generate the XML file by filling in the metadata into a template. The system administrator can enter the text of this template as a system parameter.

3.1.3 Add metadata to playground

The playground editing form should contain the following fields:

- Name (existing),
- Caption (existing),
- URI (existing),
- Language (new),
- Description (this is a new field, the existing "description" field will be renamed to "background),
- Keywords (new),
- Contribution (new),
- Size (new),
- Background (existing field, previously known as "description")
- Status (existing).

See the Appendices for further details.

3.1.4 Export playground meta data

To add playgrounds to online catalogues of learning objects, a valid XML file with metadata is needed.

The application will generate a standardised metadata file based on the metadata entered by the user. A few known characteristics of the playground may be added automatically.

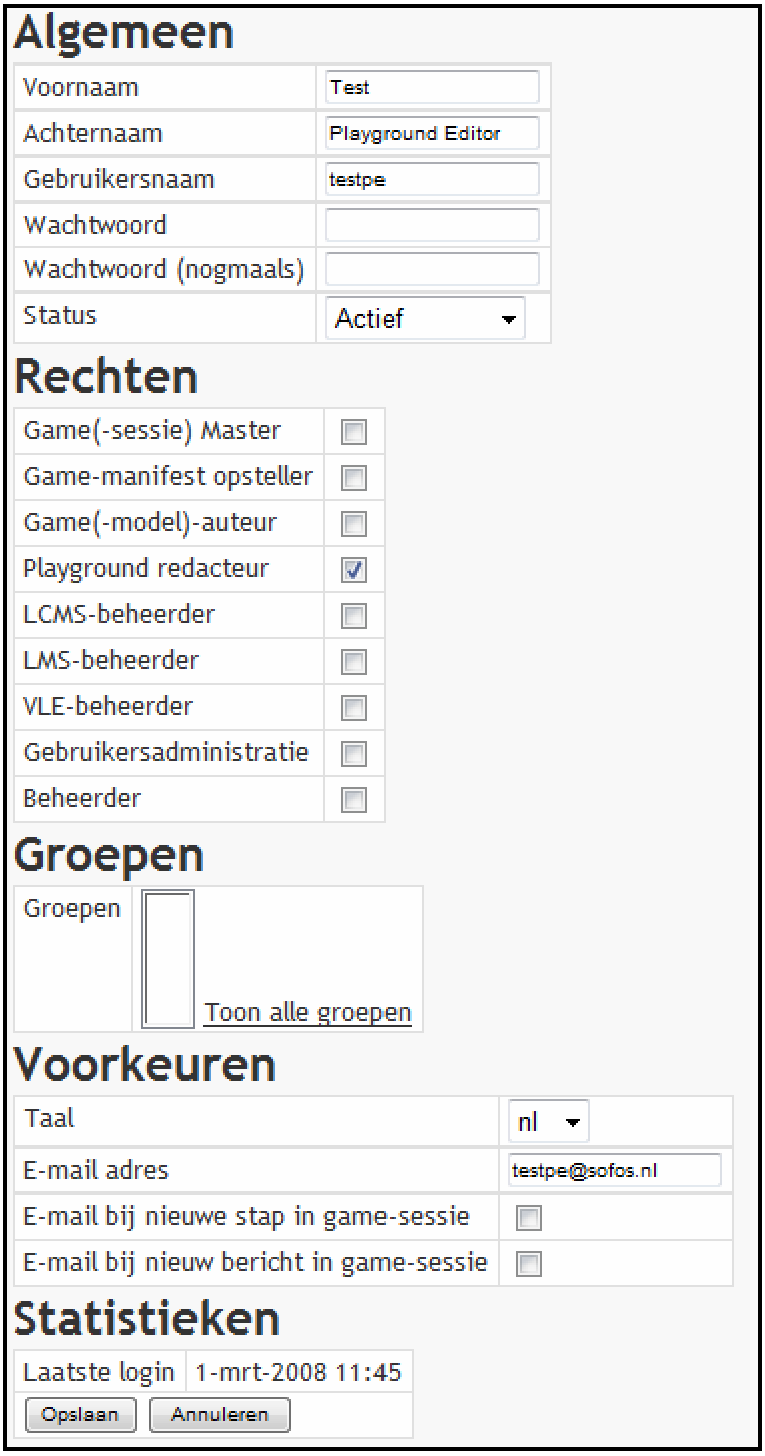
The application will generate the XML file by filling in the metadata into a template. The system administrator can enter the text of this template as a system parameter.

3.2 Technical

3.2.1 Use Moodle site as optional portal to the Cyberdam games

The Cyberdam website (www.cyberdam.nl) has been built upon the popular open source Course Management System Moodle. This system contains a range of functions for registering users, maintenance of user data, allocating privileges, enrolling for courses and managing subgroups.

From a trusted Moodle site like www.cyberdam.nl a registered user must be able to register/login automatically to a Cyberdam application. The Moodle site informs the Cyberdam application on almost all data normally filled in in when a new user is registered. The only two exceptions are the preferences for e-mail notfications. They do not exist in Moodle. In stead the default values as specified by the Cyberdam system variables will be used.



Algemeen

Voornaam	Test
Achternaam	Playground Editor
Gebruikersnaam	testpe
Wachtwoord	
Wachtwoord (nogmaals)	
Status	Actief

Rechten

Game(-sessie) Master	☐
Game-manifest opsteller	☐
Game(-model)-auteur	☐
Playground redacteur	☑
LCMS-beheerder	☐
LMS-beheerder	☐
VLE-beheerder	☐
Gebruikersadministratie	☐
Beheerder	☐

Groepen

Groepen	Toon alle groepen

Voorkeuren

Taal	nl
E-mail adres	testpe@sofos.nl
E-mail bij nieuwe stap in game-sessie	☐
E-mail bij nieuw bericht in game-sessie	☐

Statistieken

Laatste login	1-mrt-2008 11:45

The following fields are involved:

- username
- password
- first name
- last name
- rights (these are the roles the Moodle user has in the course from where the Cyberdam application is called)
- group membership (only one group: the name of this group is the short name of the Moodle course from which the application is called)

- language
- e-mail address

3.2.2 Create standards based connection to mainstream VLE's

Many higher education institutes use an LDAP server as its central source for authenticated users and their rights. Some others use Active Directory. It would make the integration of Cyberdam into education less difficult if the Cyberdam application could use either or both LDAP and Active Directory. This could imply that there would exist three options for registering and logging into the Cyberdam application:

- native registration (manually or via file upload by a user with user registration rights),
- registration via a trusted Moodle portal like www.cyberdam.nl(see above),
- registration via LDAP or Active Directory.

We have to investigate what the implications are of extending the Cyberdam application to cooperate with an LDAP or Active Directory server.

4 Improved playground functionality

4.1 Make playground map more data driven, i.e. variable object category names

The actual map mechanism uses built in data for playground object categories. Mechanism and data have to be unbundled. The category names have to be maintained by the LCMS administrator as part of the playground data.

Each playground object falls in a category (industry, government, etc.). We have one set of categories now, but since we will be using playgrounds in a bigger context this is not sufficient. We might use maps in the form of organizational charts at some point, leading to a totally different classification.

Change the following:

- Add a table 'categories' with fields id, description and playground id
- On the change playground screen, add link 'change categories'
- The link opens a new 'categories' screen, with the categories listed as usual. This means:
    1. Header with title, breadcrumb, etc.
    2. Introduction text
    3. Table with all fields as columns: id, description and playground name
    4. Each row has links 'modify' and 'delete'
    5. Below the table a button 'add'
    6. Normal paging system
    7. 'Back' button
- Add or modify happens in a new screen 'category' with
    1. fields:
        1. id (modifiable!)
        2. description
        3. playground: a dropdown box with all playground names. The id's are not visible. Initially the option 'please select...' is shown.
    2. Buttons 'save' and 'cancel'
    3. Validation that all fields are filled and the id is unique
- On the screen 'playground object' the category selection is now from the set from this playground
- The multi language keys for the categories are no longer needed: the playground itself is already language dependent.

4.2 Easy readonly access to playgrounds for various user roles

During game sessions participants can inspect the playground map, the playground directory, the playground objects and the playground introduction text. From 2.1 they can even see more playgrounds in this way (if they have been included into the game manifest).

For game authors, game manifest composers, game masters, LCMS admins, system admins it makes sense to have this function at hand when needed.

These users will need the inspect function especially when involved in other Cyberdam- functions like composing a manifest. In order not to disturb these functions-in-progress, the inspect function could be implemented as a second window.

In the current implementation, the playground information is accessible through three menu options (per playground) in the session. While reading and writing messages or doing other work in the game, or when composing games, you can't see the map for reference. And when viewing the map, it must be closed to see detailed information on objects. It would be beneficial to see both at the same time.

We will introduce a new window for this. The window can be opened using a link in the header. What playgrounds are visible depends on the user rights. Participants can only open the window once a session is opened, and will see the playgrounds used in that session. All other users will see all playgrounds at any time.

The window looks very similar to the way playgrounds are shown in the session now: menu on the left and main view to the right. The menu contains exactly the same as in the session: an entry per playground that can be opened up to show three sub entries: map, address book and description. The options also work the same. The window opens with a default size of 800x800 which fits the current map nicely, and positioned at 100x100. It is freely resizable and movable.

5 Future use of attributes, values and calculations grounded

5.1 Introduction

From the Cyberdam application point of view, the state of a Cyberdam game session consists of the name of the current step-of-play, the activities for each of the roles during this step and a stock of messages exchanged and files uploaded during the current step and the previous ones.

We want to extend this state concept by attributes that have certain values. These serve various purposes:

- they give game sessions a memory for scores, amount of capital, number of inhabitants, pollution level, annual production and whatever "stock" a game author wants to record (from moment to moment) and to display to the participants when appropriate;
- they make dynamic feedback to game sessions possible (once their values can be fed to a calculation module and the outcomes of the calculation can be fed back to attributes again);
- they make dynamic game session progress possible and make game content variable (e.g. certain values may trigger a move of the game-session to another step-of-play, or change a condition to show or hide an activity).
- In Cyberdam 2.2 we want to ground this concept by realising some basic functions. In future versions extra functions will be added.

In Cyberdam 2.2 we want to realise:

- Create attributes and assign initial values
- Give game masters the option to inspect and/or change attribute values (via a new activity type that might be useful for participants as well).
- Insert attribute values in texts to be displayed.

5.2 Create attributes and assign initial values

- there will be system wide attributes, game model attributes and game model role attributes
    1. system wide attribute are predefined; to start there are two: calendar date and clock time;
    2. game model attributes are predefined (e.g. model.name) or declared by the game author as part of the game model (name of actual step-of-play; time spent in actual step-of-play);
    3. model role attributes are predefined (e.g. role.name) or declared by the game author as part of the game model;
- the system assigns values to predefined attributes automatically;
- the game author assigns initial values to self declared attributes (character strings or real numbers).

5.3 Give game masters the option to inspect and/or change attribute values

- We introduce a new type of activity: the form. The activity consists of an instruction (rich text) followed by a number of fields that can be filled in and in fact set certain variables plus (optional) attachments.
- From the session control page, the game master can go to a debug-page. This page lists all the variables and their actual values.

5.4 Insert attribute values in texts to be displayed

- The game author can include the values of attributes in activity instruction texts; e.g. {burgomaster.score} will be replaced by the value of attribute score belonging to role burgomaster.

6 Improved session reports

6.1 Display session history

- The session control page should be extended by a Session Log page that displays a table with all session events (messages sent and files uploaded) (in fact the information the participants have, but ordered by date+time). The following columns are provided:
  - user name
  - role name
  - date+time
  - step-of-play
  - type (message/upload)
  - for uploads: file name, actual name
  - for messages: receiver(s), subject
- Clicking an item on the list should display the corresponding item in the session home page of the role.

6.2 Create game session synopsis

The game master can create a session synopsis. It consists of a report with the following data:

- header with the name of the game master who asked for the synopsis plus date and time,
- URL of the current system, application version and build date, database name,
- the content of the game session colophon,
- a list of game roles and the users playing these roles (at the moment the synopsis was generated),
- the session history (at the moment the session synopsis was generated)

This information can be generated as an XML file, a txt file or a PDF file.

6.3 Create game session archive

The game master can create a session archive. It consists of a zip file with the following content:

- game session synopsis,
- a listing of the full text of all messages with a relative link to attachments (i.e. uploaded files),
- a folder called roles
- a subfolder for each rol containing all files uploaded by that role.

This information will be generated as a ZIP file.

7 Appendices

7.1 More on metadata

Both for playgrounds and for game models a metadata file is needed. The next appendix contains a sample metadata file for playgrounds. The metadata file for game models has exactly the same structure. Some fields have different values.

To implement the metadata we will provide two templates:

- template for game model metadata,
- template for playground metadata.

The major part of the playground metadata file consists of fixed text. The next paragraphs list the variable data that have to be inserted into the template.

7.1.1 Game model metadata

The table lists the variable data that have to be inserted into the game model metadata template.

LOM 1.0 Category	Name	Example	Source	Comments
1.1.2	General.Identifier.Entry	Cyberdam	New field, URI, 80 chars.	Position in form below Caption field.
1.2	General.Title	Cyberdam	Game model name	Cyberdam 2.0
1.3	General.Language	nl	New field, 2 chars	Position in form: below URI
1.4	General.Description	Text, max. 200 chars.	New field, 200 chars.	Position in form below Language; note: the current description field better could be called Background.
1.5	General.Keywords	Text, max. 1000 chars.	New field, text max chars.	Position in form below the new Description entryfield.
2.1	LifeCycle.Version	Date ddmmjjjj	Last mutation date in different format.	Cyberdam 2.0
2.2	LifeCycle.Status	final	game model status = under construction or private - > "draft", public-> "final", obsolete->"unavailable"	Cyberdam 2.0
2.3	LifeCycle.Contribute.Role.Entity	Project Leren in een Virtuele Wereld o.l.v. Diny Peters en Pieter van der Hijden	New field, text max, 200 chars.	Position in form below the new Keywords field
4.2	Technical.Size	3137536	New field, text max 10 digits	Number of bytes. Position in form below the

				Contribution field.

The game model form should start with the following fields: Name, Caption, URI, Language, (New) Description, Keywords, Contribution, Size, Background (= old description), etc.

7.1.2 Playground metadata

The table lists the variable data that have to be inserted into the playground metadata template.

LOM 1.0 Category	Name	Example	Source	Comments
1.1.2	General.Identifier.Entry	Cyberdam	Playground URI	Cyberdam 2.1
1.2	General.Title	Cyberdam	Playground Name	Cyberdam 2.1
1.3	General.Language	nl	New field, 2 chars	Position in form: below URI
1.4	General.Description	Text, max. 200 chars.	New field, 200 chars.	Position in form below Language; note: the current description field better could be called Background.
1.5	General.Keywords	Text, max. 1000 chars.	New field, text max chars.	Position in form below the new Description entryfield.
2.1	LifeCycle.Version	Date ddmmjjjj	Last mutation date in different format.	Cyberdam 2.1
2.2	LifeCycle.Status	final	playground status = under construction -> "draft", public-> "final", obsolete->"unavailable"	Cyberdam 2.1
2.3	LifeCycle.Contribute.Role.Entity	Project Leren in een Virtuele Wereld o.l.v. Diny Peters en Pieter van der Hijden	New field, text max, 200 chars.	Position in form below the new Keywords field
4.2	Technical.Size	3137536	New field, text max 10 digits	Number of bytes

The playground form should start with the following fields: Name, Caption, URI, Language, (New) Description, Keywords, Contribution, Size, Background (= old description), etc.

7.2 Sample metadata file for playground

```
<?xml version="1.0" encoding="UTF-8"?>

<!--This is a Reload version 2.5.2 Metadata document-->

<!--Spawned from the Reload Metadata Generator - http://www.reload.ac.uk-->
```

```
<lom xmlns="http://ltsc.ieee.org/xsd/LOM"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ltsc.ieee.org/xsd/LOM lom.xsd">
  <general>
    <identifier>
      <catalog>playground.cyberdam.nl</catalog>
      <entry>Cyberdam</entry>
    </identifier>
    <title>
      <string>Cyberdam</string>
    </title>
    <language>nl</language>
    <description>
      <string>Cyberdam is een virtuele, middelgrote, Nederlandse stad die
bestaat uit een plattegrond met 150 objecten, een adresboek met objecten en
gedetailleerde gegevens daarover.</string>
    </description>
    <keyword>
      <string>"virtuele stad" "virtuele wereld" Cyberdam"
playground</string>
    </keyword>
    <aggregationLevel>
      <source>LOMv1.0</source>
      <value>2</value>
    </aggregationLevel>
  </general>
  <lifeCycle>
    <version>
      <string>22022008</string>
    </version>
    <status>
      <source>LOMv1.0</source>
      <value>final</value>
    </status>
    <contribute>
      <role>
        <source>LOMv1.0</source>
        <value>editor</value>
      </role>
      <entity>Project Leren in een Virtuele Wereld o.l.v. Diny Peters en
Pieter van der Hijden</entity>
      <date>
        <description>
```

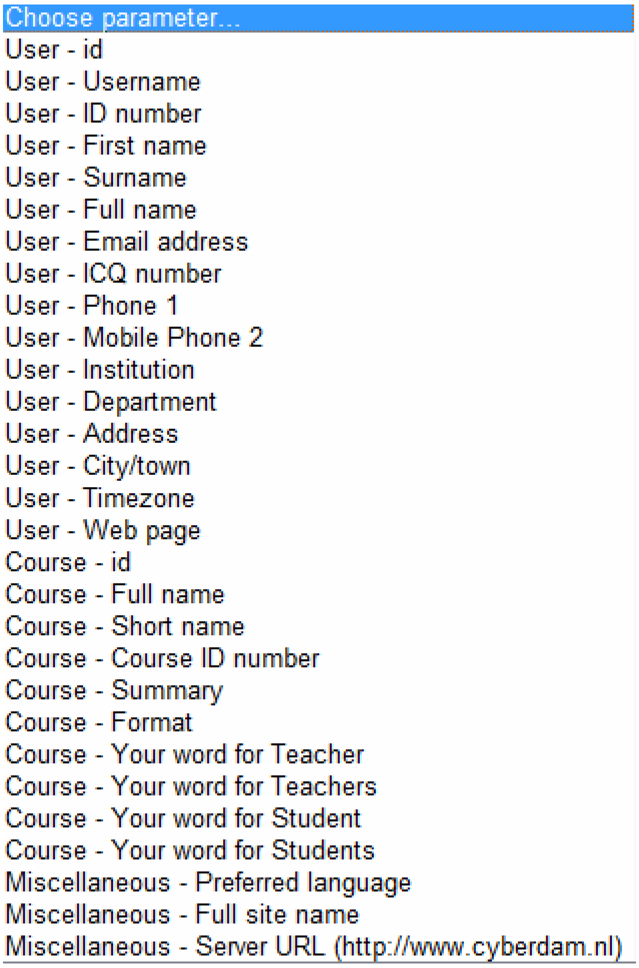
```
          <string>2008</string>
        </description>
      </date>
    </contribute>
  </lifeCycle>
  <metaMetadata>
    <metadataSchema>LOMv1.0</metadataSchema>
    <metadataSchema>czp_povpbve_v1p2</metadataSchema>
    <metadataSchema>SCORMv1.2</metadataSchema>
  </metaMetadata>
  <technical>
    <format>application/zip</format>
    <size>3137536</size>
    <location>http://www.cyberdam.nl</location>
    <installationRemarks>
      <string>Gebruiker met LCMS Admin rechten kan playground zip file
uploaden naar Cyberdam Java applicatie op de server.</string>
    </installationRemarks>
    <otherPlatformRequirements>
      <string>Cyberdam 2.1 of hoger, Java 1.6, Tomcat, Apache, MySQL op de
server</string>
    </otherPlatformRequirements>
    <duration />
  </technical>
  <educational>
    <learningResourceType>
<source>http://download.edustandaard.nl/xsd/vdex_learningresourcetype_czp.x
ml</source>
      <value>informatiebron</value>
    </learningResourceType>
    <intendedEndUserRole>
      <source>LOMv1.0</source>
      <value>learner</value>
    </intendedEndUserRole>
    <typicalAgeRange>
      <string>18+</string>
    </typicalAgeRange>
  </educational>
  <rights>
    <cost>
      <source>LOMv1.0</source>
      <value>no</value>
    </cost>
    <copyrightAndOtherRestrictions>
      <source>LOMv1.0</source>
      <value>yes</value>
    </copyrightAndOtherRestrictions>
    <description>
      <string>De Creative Commons Naamsvermelding-Niet-commercieel-Gelijk
delen 3.0 Nederland Licentie is van toepassing op dit werk. Ga naar
http://creativecommons.org/licenses/by-nc-sa/3.0/nl/ of stuur een brief
naar Creative Commons, 171 Second Street, Suite 300, San Francisco,
California, 94105, VS om deze licentie te bekijken.</string>
    </description>
  </rights>
  <classification>
    <purpose>
      <source>LOMv1.0</source>
      <value>educational objective</value>
    </purpose>
    <taxonPath>
      <source>
        <string>-</string>
      </source>
    </taxonPath>
    <description>
      <string>Gemeenschappelijke context (common playground) voor online
rollenspellen ontwikkeld en gespeeld met de Cyberdam-applicatie.</string>
    </description>
  </classification>
</lom>
```

7.3 More on Moodle link

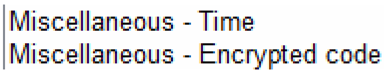
There are two ways to extract the required data from Moodle:

- let Moodle query its database and generate the required data
- use the Moodle hyperlink parameters

Within the Moodle system a course developer can enter a hyperlink to an external web site and provide it with the following parameters:



Choose parameter...
User - id
User - Username
User - ID number
User - First name
User - Surname
User - Full name
User - Email address
User - ICQ number
User - Phone 1
User - Mobile Phone 2
User - Institution
User - Department
User - Address
User - City/town
User - Timezone
User - Web page
Course - id
Course - Full name
Course - Short name
Course - Course ID number
Course - Summary
Course - Format
Course - Your word for Teacher
Course - Your word for Teachers
Course - Your word for Student
Course - Your word for Students
Miscellaneous - Preferred language
Miscellaneous - Full site name
Miscellaneous - Server URL (http://www.cyberdam.nl)



Miscellaneous - Time
Miscellaneous - Encrypted code

Unfortunately, not all data required can be extracted in this way.

From the Moodle documentation at http://docs.moodle.org/en/mod/resource/file.

"The parameter settings are completely optional, and are only useful when you need to pass some Moodle information to the resource file or web site. If you define any parameters they will be passed to the resource as part of the URL (using the GET method). The left column allows you to choose information to send, and the right column allows you to give it a name. Note that the user information will be from the user who is looking at this resource, and the course information is from the course that this resource is part of."

Example: calling the Cyberdam application with username, first name and last name as parameters results in the following link: