INFRASTRUCTURE AND INTEGRATED TOOLS FOR PERSONALIZED LEARNING OF READING SKILL

D3.3 – System Specifications

Document identifier	**System Specifications**
Date	**2017-04-26**
WP	**3**
Partners	**NTUA, UCL, ULBS**
WP Lead Partner	**NTUA**
Document status	**V1.2 / Final**

Grant Agreement number:

731724 — iRead H2020-ICT-2016-2017/H2020-ICT-2016-1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 731724

Deliverable Number	D3.3
Deliverable Title	System Specifications
Deliverable version number	V1.2
Work package	3
Task	T3.3 Technical requirements
Nature of the deliverable	Report (R) /
Dissemination level	Confidential
Date of Version	2017-04-26

Author(s)	A. Symvonis, K. Karpouzis
Contributor(s)	C. Raftopoulou, C. Mihu
Reviewer(s)	I. Mihu
Abstract	This Deliverable describes the system specifications of the iRead infrastructure based on the nature and content of information contained and circulated, the need for privacy and security, the requirements for adaptation in terms of content and activity selection, and the technological requirements imposed by the need to deploy in schools (where internet connectivity is often limited and not reliable).
Keywords	System requirements, software architecture, security, privacy, content selection, adaptation

Document Status Sheet

Issue	Date	Comment	Author
V0.1	2017-02-28		K. Karpouzis
V0.5	2017-03-25	Contributions on security	C. Mihu, I. Mihu
V0.9		Contributions by A. Symvonis	

Table of contents

1. EXECUTIVE SUMMARY 6

2. OPERATIONAL DESCRIPTION 7

2.1. User Profiles 7

Type of Input Documents 7
Personalisation - Adaptive Features 7
Text Classification 8
Recording usage history 8
Offline functionality 9

3. CATEGORIES OF DATA 10

User Related Data 10
Resource Data Bank 10
Learning Material 11
Expert Data 11
Data Sensitivity 11

4. USE CASE SCENARIOS 13

Category A: iRead used at school 13

A-01: "School registration in the system" 13
A-02: "Teacher creates student account" 14
A-03: "Student plays with iRead Games" 17
A-04: "Student uses iRead Reader-app" 19
A-05: "Student overviews her progress" 20
A-06(a): "Teacher monitors a student's progress" 21
A-06(b): "Teacher monitors the progress of her class/school" 22
A-07: "Teacher selects learning material for a specific student" 22

Category B: Independent use of iRead 24

Category C: External apps using existing user-models 24

C-01: "Personalized File-explorer" 24
C-02: "eBook store with personalized search functionality" 26
C-03: "Personalized meta-search Engine" 27
C-04: "Collaboration of School with eBook publisher" 27

Category D: Extending the iRead system 28

D-01: "iRead supporting new domain models" 28
D-02: "Add a domain model for a new language" 29
D-03: "Add or update language resources for existing domain models" 31
D-04: "Update the iRead Games by incorporating new activities" 31
D-05: "New Game that wants to uses existing user models" 31
D-06: "New type of user is inserted" 31

D-07(a): "A New metric is needed (for analytics)" 31
D-07(b): "A New metric is needed (for content classification)" 31
D-08: "Provide access to language resources/services (without authentication)" 32

Category E: Automatic update of iRead system 32

E-01: "Update user-model" 32
E-02: "Domain-model update" 32
E-03: "Processing for analytics" 33

Category F: Other uses of iRead and its infrastructure 33

F-01: "Dedicated server iRead installation" 33

5. CLIENT-SERVER ARCHITECTURE 35

Advantages of a server-based architecture 35
System architecture 36

6. SYSTEM SPECIFICATIONS 38

Hardware – Software Specifications 39
Server 39
Client (Tablet/desktops/laptops) 39
Interoperability 40
System Security 40

Table of figures

Figure 1: UML use case for A-01 14
Figure 2: UML use case diagram for A-02 16
Figure 3: UML diagram for playing the literacy games 19
Figure 4: UML diagram for the reader application 20
Figure 5: UML diagram for assisted selection of reading material 23
Figure 6: UML diagram for independent use of a file explorer app 25
Figure 7: UML diagram for assisted eBook selection 26
Figure 8: UML use case diagram for adding a new domain model 29
Figure 9: Use case diagram for adding new language models 30
Figure 10: Abstract system architecure 36
Figure 11: Server/Client Architecture 38

1. EXECUTIVE SUMMARY

This Deliverable describes the system specifications of the iRead infrastructure based on the nature and content of information contained and circulated, the need for privacy and security, the requirements for adaptation in terms of content and activity selection, and the technological requirements imposed by the need to deploy in schools (where internet connectivity is often limited and not reliable).

The first sections describe what is expected from the system in terms of functionality, the nature of data and documents stored in the database and used by the applications, and some additional requirements, mostly related to how the iRead functionalities should be available when internet connectivity is not reliable and what kinds of usage history should be logged in order to allow for adaptation and proper content selection.

In the following, the server-side system architecture is described, identifying the components that materialise the functionalities mentioned before, followed by hardware requirements for the server-side and clients. Since this is a four-year project, requirements for the clients may change (but not radically), since currently available tablets may become obsolete.

2. OPERATIONAL DESCRIPTION

2.1. User Profiles

For each user the system should maintain a number of "user profiles" (one per language) which contains information about the usage history and current status of the user with respect to reading and learning activities (types of problems for the particular language, degree of severity, usual reading speed, etc.), cognitive age, interests, user preferences related to the set-up of the system, etc.

Initial profile. The initial profile can be built with the help of an expert or an initial placement test, while a default or lower-resolution (with fewer attributes and levels of current expertise) profile may be used whenever a roaming profile for that user cannot be found (e.g. when communication with the server is not possible). This placement test will be offered by the iRead platform, but may also be performed using external tools (not necessarily digital).

Profile Updates. Because of the requirement of offline operation, user profile updates will take place in real time (hence the need for an offline profile). Depending on the use case, profile updates may take place after the server identifies the need to update, based on the progress of the user. The precise update mechanism will be described in a later deliverable, but the technical provisions for this capability should already be integrated in the infrastructure.

Type of Input Documents

iRead aims to go beyond the state of the art of previous projects (which used textual information from text or PDF files) and utilise multimedia and web documents, besides discrete words or small phrases. Depending on the activity, any of the above mentioned document types may be used in the application. In addition, multimedia documents stored in the server side should also be processed and annotated with lexical, syntactic, and semantic information (e.g. information about where to hyphenate or level of difficulty based on vocabulary and syntactic complexity of the text); this process can be performed offline (when documents are uploaded) and should be repeated whenever additional tools are available. The iRead infrastructure should also contain tools to provide this sort of processing on-the-fly, for texts retrieved by students and teachers/experts, at least for hyphenation and smart highlighting in the reader applications, as well as for quick characterisation of the level of difficulty of the document with respect to the active user profile.

Personalisation - Adaptive Features

Personalisation in iRead takes place via the provision of a dynamic profile for each user, which is used by the server and applications (whenever possible) to identify and suggest the next learning activity to be used by the student, the content that that

activity should utilise, with respect to the chosen learning objective and the severity of the problem, and also the next learning objective which should be tackled by the student.

In many cases, experts prefer to perform one or more of the above mentioned tasks manually, especially when it comes to choosing the next problem/learning objective. This is the main reason why we use the word 'suggest' instead of 'choose' on behalf of the expert. Experts should also be presented with the ability to alter the suggestions made by the system, add or remove specific words or texts to be used by the students in the iRead applications, and design a complete learning schedule for the following sessions of the student (which learning activities and apps to use and the content associated with those activities).

Text Classification

Text classification in iRead involves processing of documents (texts and web pages) with respect to the user's profile (or, possibly, pre-defined profiles corresponding to prototypical users or stages of learning development) and the numerical estimation of their relevance to that profile. The main idea here is that the same text may be suitable for one particular user, given their profile and usage history, but too difficult or too easy for another user.

This component relies on a set of low-level processing components to function (hyphenation and phonetics analysis, syntactic analysis), expert knowledge of what should be expected from the part of the students at each school class, and the categories of different problems in the user profile for each language.

Recording usage history

Recording and processing usage analytics is an important part of iRead, since it allows for the estimation of the temporal evolution of the user profile (whether the abilities of the user with respect to particular attributes of their profile have evolved over time), correlation of progress with learning activities, and better adaptation of the apps and games and better content selection. Usage history recording is also important in the case of offline operation, i.e. whenever the apps function without a continuous and consistent link to the iRead server, resulting in incremental and offline updates of the user profile.

Another use case where data recording is important is functionality of third-party applications. In the case of apps designed by the consortium and wherever access to their internal structure is available, data logging is straightforward and integrated in the app itself; however, since we want the iRead infrastructure to be open to apps built outside the consortium or project, we need to identify a minimal set of logging requirements for the user profile history and adaptation to function properly, and leave any additional or proprietary attributes to be logged by the apps in an internal

fashion. Data privacy is an important factor here, since some of the attributes of the user profile and usage history must remain internal to the iRead infrastructure and not be shared with any third-party apps.

The actual attributes to be logged will be decided with reference to the apps built in iRead and the contents of the user profile. In the case of games, it is also important to log, besides the content presented to the user, the result of each learning activity (success/failure), and any other choices made by the expert of user (i.e. which game the student chose to play, indicating their preference), any game behaviour information, such as time to complete, in-game assessment and evaluation questionnaires, and game-specific information.

Offline functionality

In iRead we make the assumption that access to the central server will not always be available, either due to lack of infrastructure (e.g. slow or unreliable internet connection) or due to school policy; however, this connection must be available at specific times, to allow for content delivery to the tablets and for uploading usage history and analytics to the server, possibly triggering user profile updates. Some of the required functionality for offline use includes:

- Enough content must be downloaded to the tablet for student to use with games for at least fifty sessions. This can be achieved, e.g. by scheduling unattended updates on Friday afternoons, when the school network typically encounters limited traffic.
- Applications other than games should manage content in such a manner so that offline usage is possible (e.g. by installing or downloading content whenever an internet connection is available); relevant requirements are up to the designers of those applications
- If every student has access to a particular device, then the student's profile can be downloaded to that device and updated accordingly; local server functionality can be used to access the relevant content for that profile
- If that's not possible, we will investigate the possibility to setup a local server at school level; this is also convenient for keeping sensitive data within the school premises or network
- The minimum working requirement is for a number of predetermined profiles to exist locally, so that teachers can assign the best fitting profile to each student; no profile adaptation is possible in this case, but the students statistics will be uploaded to the central server whenever possible, catering for updates at that time.

3. CATEGORIES OF DATA

User Related Data

- **User Profiles:** The basic information that the iRead system needs to store in order to work properly is a *profile* for each registered user and language. By having access to individual profiles the iRead system can provide personalised and adaptive learning sessions to each of its users.
- **User Preferences:** some the iRead apps will support customisation based on user preferences: e.g. of the look-and-feel of the app (style of highlighting for the reader app, background color, font type and size, etc.).
- **Usage history:** When users interact with the system and its components, a log of their actions will be stored, enabling the system to make suggestions and maintain or adapt the learning strategies according to the users' analytics.
- **User authentication data:** The iRead system interacts with multiple users (children, teachers/experts, parents). In some use cases, it is desirable to allow multiple users to gain access to the system from the same device (tablet, laptop, etc.). Allowing access to the system to multiple users from the same device requires an authentication mechanism. Such a mechanism is required in order to load the appropriate user profile, load learning material for the specific user and update the history of the appropriate user. In addition, access to user data must be controlled in terms of data security. For example, a teacher may be allowed to gain access only related to his/her students. In order to accomplish the above, a user authentication system and an access control system is required.
- **Data analytics:** Temporal statistical data related to the usage of the system from each user will be stored or, alternatively, inferred by the historical data related to a user's learning session.

Resource Data Bank

- **Dictionaries:** Since the iRead system may provide word based activities (e.g., a game that displays 3-syllable words or an activity that asks from the child to spell a word), iRead will include annotated dictionaries to support these activities. The core of a dictionary is a database with the words of a language, plus additional data such as hyphenation rules, rules for stemming or information related to the type of each word (e.g., verb, noun, adjective, etc.).
- **Texts:** This category contains text documents from school books and other media that students of the targeted age consume; information here is mostly text-based (as opposed to e-books), but there may be descriptive images in the text to make it more appealing for the younger audiences. As mentioned before, texts should be categorised either offline or online with respect to how suitable they are for actual or prototypical users of the iRead infrastructure and apps.

- **eBooks:** eBooks constitute a new use case for the iRead system and expand its user base significantly, given the wide adoption of e-readers and tablets among the targeted audience. The difference between text documents/PDFs and eBooks is that eBook reading software may offer additional capabilities which alter text appearance (usually font face and size) and may also utilise text-to-speech capabilities. In our case, this essentially means that these documents are usually not reach in other media besides text, and that some basic annotation should be there to cater for these advanced capabilities, taking also into account the active user profile.

Learning Material

- **Text:** iRead is a learning environment for children and adults, thus it utilises a variety of learning resources, mainly in the form of text, stored locally or accessed online. Depending on the use case, these texts may also come with DRM (digital rights management) restrictions, which should be adhered to.
- **Learning activities:** These activities can be games, supervised and unsupervised exercises, tests/quizzes, etc. In a sense, each activity is a separate program that may require specific data in order to be carried out. In most cases, the content of each activity will be user dependent (based on the user's profile and history), but will also depend on the mechanics of the application or game (some apps will utilise discrete words, while others will need sentences or small passages of text).

Expert Data

- **Learning Strategies:** The experts' knowledge on how to structure a strategy to facilitate learning reading must be encoded within the iRead system. This can be stored as a set of rules, mapping user profile states to choices of content and application or game to be presented to the user. Our initial strategy will be to encode this knowledge using a rules evaluation component, such as DROOLS (an open source, Java-based rules management system), mapping student competencies, emerging results, and selection of activity to particular content (texts or words) and activities suggested by the system to either experts or users. These rules can then be refined based on actual usage when the iRead system is deployed to a larger audience.

Data Sensitivity

We attempt to classify the data types described above into two groups according to their sensitivity with respect to security.

- **Low Sensitivity:**
  - Language Resources
  - Learning Material
  - Expert Data

- **High Sensitivity:**
  - User Profile
  - Usage history
  - User preferences and selection of content

Note that language resources, learning material and expert data are user independent. It suffices to store them in a server or in every device that runs the iRead software. So, no need for extra data protection is required.

On the other hand, all types of data that are related to specific users are classified as critical. Profiles must be protected since they are personal sensitive data for each user. The same holds for the usage history and the preferences of each user. Also, the importance of protecting authentication date is straightforward.

4. USE CASE SCENARIOS

The objective of this section is to determine the functional and non-functional requirements of the iRead system. The primary tool we use are use-cases, a commonly used technique for software and systems engineering. By considering different scenarios of use of the system, it is possible to determine what each component of the system should be or do, in other words its functionality.

We consider the following categories of iRead use:

- iRead used at school.
- Independent use of iRead.
- External applications using existing user-models.
- Extending the iRead system.

Category A: iRead used at school

In this section we focus on the use of iRead at school. The school has decided to use iRead system as part of their teaching plan. Students in a classroom are registered in the system by their teacher, who is responsible for this class. Once a student enrolls the system, a user-model is initialized and the student can start playing the literacy games, using the Reader-app, and more generally interacting with the iRead components. Furthermore, the teacher is able to monitor the progress of her classroom or of a particular student and possibly adjust her teaching plan. The above scenarios are examined one by one in the following.

A-01: “School registration in the system”

For a school that decides to use iRead system, a mechanism for registering the school in the system is required.

1. School-administrator logs in the user-management web-app
2. School-administrator selects the “add school” function
3. School-administrator enters details of school and classrooms
4. School-administrator selects the “add teacher” function
5. School-administrator enters details of teacher
6. School-administrator logs out

Notes

- School has an administrator or IT personnel.
- School-administrator is assumed to be registered in the system.

Issues/Questions

- **Accepting a school .**

A formal procedure for allowing a school to use the iRead system is required. The school director could fill in an application form asking to use the iRead system. Once the request is approved, a "school-administrator" user will be created with appropriate permissions to register the school/classrooms and enroll teachers for the school.

- **Who is responsible for registering a school?**

In the above use-case a school can be registered through a user-management app. The person responsible for entering the details of the school and classrooms, and for registering teachers, could be an administrator of the school or the school director. Another option could be that schools are registered by the administrator of iRead system. In general, if a school, an expert or a student, decides to user iRead, possible ways of registration should be considered.

- **School/classroom/teacher details.**

The information required for a school/classroom/teacher should be determined.

- **Update of teacher's details / change of password.**

Suppose that a teacher wants to change her details. We should determine whether this is allowed and how. We also need a "forgot password" and "reset password" procedure for the teacher's account. This could be done either by the school-administrator or by an email stored along with the teacher's details.

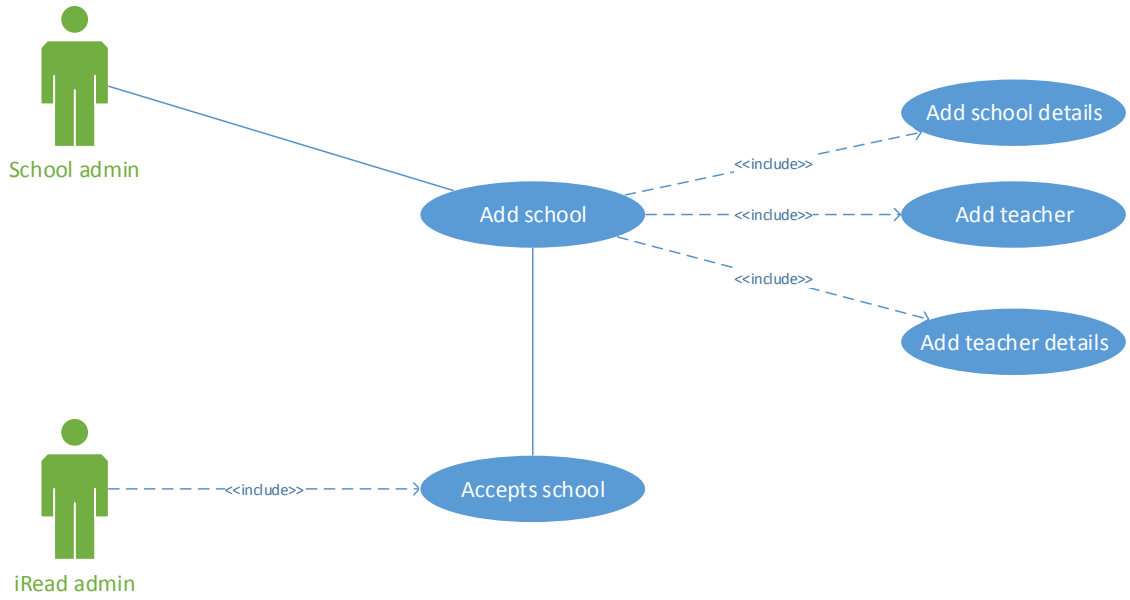

Figure 1: UML use case for A-01

A-02: "Teacher creates student account"

In school environment, a teacher is responsible for a class of students. In the following scenario, the teacher enrolls a student in the system.

1. Teacher logs in the user-management web-app
2. Teacher selects the “add student” function
3. Teachers enters details of student
4. Personal details: first name, last name, gender, age, mother language
5. Login details: username, password
6. Language model
7. School and classroom details
8. Teacher initializes the user-model for the student with one of the following alternatives:
9. Teachers selects “initialize Profile” function and enters data into student’s initial user-model (data is assumed to be generated by a manual evaluation test)
10. Teachers selects “Auto Initialize profile” function that initiates an interactive evaluation activity for the student that “somehow” initialized the user-model
11. A default user-model is used
12. Teacher logs out

Notes

- Teacher is assumed to be registered in the system.
- Language domain-model is assumed to exist in the infrastructure.
- School is assumed to be registered in the system.
- Class is assumed to be registered in the system.
- Evaluation test and interactive evaluation activity are assumed to exist.
- Default user-models are assumed to exist.

Issues/Questions

- **More than one user-model per user.**

It may be useful to allow the same user to have more than one user-models, in the case where a student learns to read in two different languages, or in the case where we cover domains other than “learning to read” (for example, one model for EFL and one model for “learning to fly”)

  - How do we support many user-models to one student?
  - Will the login credentials (username-password) be the same?
  - How will the system know which user-model to load when a student logs in?

- **Student details.**

Are there other details we want to store for a student? Possibly we need the email of the student’s guardian. This could be useful if access to student’s user-model is granted through an email request-link.

- **Initialization of user-model.**

In the above user-case, the user-model is initialized with three different options:

  - manually entering data,

- through an interactive activity,
- using a default user-model.

In the first case, the teacher may enter the evaluation test's result at a time after enrolling the student. This implies that initialization of the student's user-model may take place after its creation. If we allow immediate usage by the student, some default values could be used for the user-model until the teacher initializes the user-model. This implies that, invocation of "(auto) initialize user-model" by teacher (i.e. manually entering data) will override the default user-model values (and any changes made during play).

In the second case, the student's user-model is initialized through an interactive activity. The duration and the content of the activity could depend on several parameters, such as age, language etc., that need to be determined.
In the last case, several default user-models should be defined. We should determine how many default user-models will be available and which parameters should be taken under consideration (e.g. age, language). If a default user-model suggests for example too difficult activities for a student, the child could be easily discouraged.

Are there other alternatives for initializing a user's user-model?

- **Manual change of a user-model.**

Suppose that the teacher creates new evaluation tests. Will the teacher be able to "update" a user's user-model by these tests? A more general question, is whether the teacher will be able to change directly the values of a student's user-model.

- **Update a student's details / change of password.**

Suppose that a student wants to change her details. This issue is also considered in Use-Case A-01. The "forgot password" and "reset password" procedure is needed, especially for the internet trial. This implies that "contact" (guardian's) email is essential for the student's details.

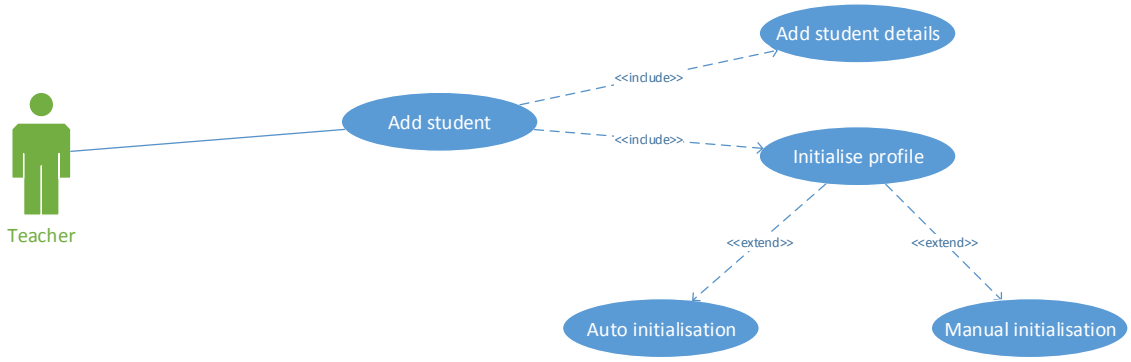

Figure 2: UML use case diagram for A-02

A-03: "Student plays with iRead Games"

This scenario describes the most common use of iRead, where a student interacts with the literacy games.

1. Student logs in the literacy games application
2. Game "gains access" to user-model
3. Game "gains access" to usage-data for user
4. A decision is made (based on user-model and usage-data) regarding
5. the next model-property to be addressed
6. which activity will be used
7. Which literacy material will be used
8. The student starts playing the activity
   Her actions are being recorded
   Static or dynamic versions of the activity may be possible
9. When student finishes with the activity,
10. the game "logs" the usage-data
11. the game calls for a possible re-evaluation of the user-model
12. any change is recorded into the user's user-model
13. Go to step 4, unless the student wants to quit playing
14. Student logs out

Notes

- Student is assumed to be registered in the system.
- User model is assumed to be initialized (or active).
- User model is assumed to be stored in the "infrastructure".
- Usage-data are assumed to be stored in the "infrastructure".
- The literacy games can be also played outside the school environment.

Issues/Questions

- **What constitutes the infrastructure of iRead?**

This is the big question for the design of the system. At the proposal we have that a Profile Management module and a Data-Usage Management module are part of iRead's infrastructure. Possibly we also want a User Management module which handles access control. What is not clear is where the logic (select next model-property, select next activity, select literacy material, re-evaluate the user-model) is located. In the iLearnRW[1] system, the logic and the data were located at the same server. At iRead, we may want to separate them.

- **Static or dynamic versions of the activity.**

Based on the performance of the user during the activity, some material may be skipped (if use is successful) so that the game is not boring (other "dynamic"

[1] EU FP7 ICT project iLearnRW - Integrated Intelligent Learning Environment for Reading and Writing (project number: 318803)

variations may be possible). In iLearnRW[1] we were using a "static" version where each student would play all the literacy material. The experts (linguists) will have to provide the exact details of dynamic activity adaptation, if such an option is finally adopted.

- **Are the language resources part of the infrastructure?**

The language resources (dictionaries, tables, sentences, services such as a syntax analyzer) seem to be supporting material to the applications that are using a particular language domain-model. They are related to the domain-model and may contain information about the model. However, different applications which use the same domain-model may build their own language resources.

- **An offline version of the system.**

A functional requirement of the iRead system is that it should be able to operate in an offline fashion. We should figure out how to implement this in a way that it is transparent to the user. It may be achieved by having the logic of the application at the client (game) side, together with enough data to feed the game for a long period. The following should also be taken into consideration

  - More than one students could use the same device.
  - Is the user-model stored locally in the application?
  - Are the usage-data stored locally? Could this be a limitation if many students use the same device?
  - A synchronization function is required.

- **Data logging.**

We should implement at least a couple of data "logging modes" for our usage-data, for example "fail"/"success"/"replay"/"nothing" etc. Proper data logging should be the responsibility of the clients (games). The logged usage-data in the system should allow us to replay the game as the actual user did. The amount of required data to achieve this should be taken into consideration.

- **Access to usage-data.**

At step 3 the application gains access to usage-data. A question that should be answered is how the logic of selecting next activities etc. makes use of usage-data. If the "logic" is part of the game, the amount of required usage-data should be considered. This is also important in the case where several users use the same device, and usage-data for all of them is required.

- **Quitting the application.**

A student could quit the application without logging out. Do we discard any recorded usage that is not sent to the system, or should the application send all remaining data and log out automatically?

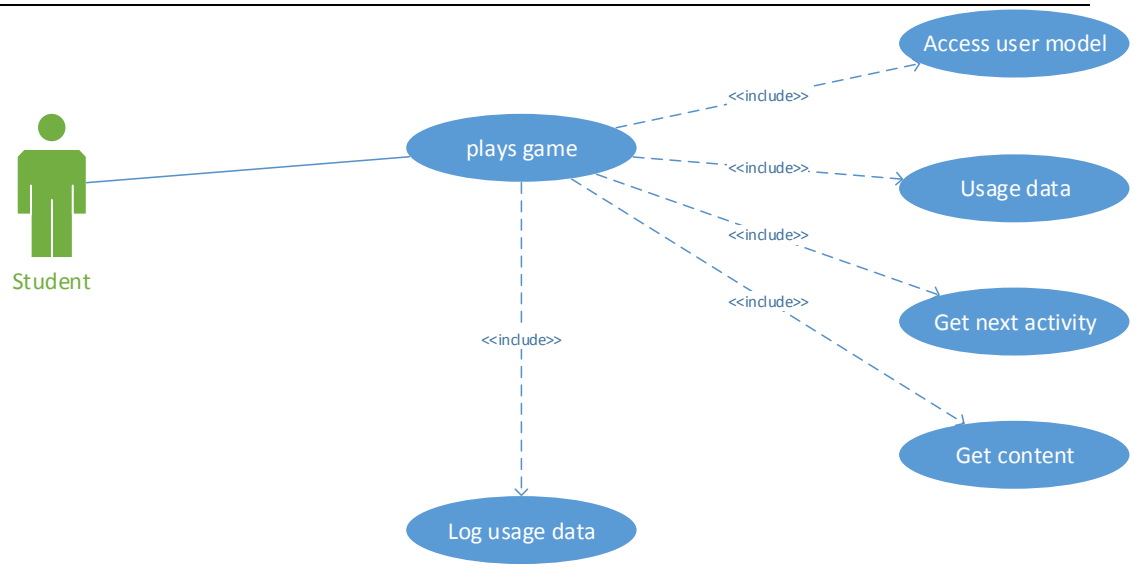

Figure 3: UML diagram for playing the literacy games

A-04: "Student uses iRead Reader-app"

This is a very vague description as the Reader-app's functionality will be defined in deliverable **D7.1: iRead Reader app Interaction and Visual Design (M12)**.

1. Student logs in the Reader-app
2. Students either loads book from a public book library or loads a document created by her teacher
3. The Reader-app displays the book/document in a user-specific manner (based on her user-model or preferences)
4. The student "reads" the document and "interacts" with it
5. On exit, the Reader-app may
6. log data related the user's reading
7. log data related to the user's interactions
8. Attempt to re-evaluate the user's user-model
9. any change is recorded into the user's user-model
10. Student logs out

Notes

- User model is assumed to be initialized (or active).
- User model is assumed to be stored in the "infrastructure".
- Libraries of books are assumed to exist.
- A teacher is assumed to be able to upload documents in a depository.
- Personalized text formatting is assumed to be supported by the Reader-app.
- Usage-data are assumed to be stored in the "infrastructure".
- The Reader-app can be used outside the school environment.

Issues/Questions

- **Functionality of the Reader-app**.

The functionality of the Reader-app will be determined during its design. The full specification of the Reader-app will be available in deliverable **D7.1: iRead Reader app Interaction and Visual Design (M12)**. As part of the design, several of its aspects should be specified including its text reformatting properties, how a user interacts with it, how the user's user-model and user preferences are taken into account, what usage-data are logged, which document formats are supported etc.

- **Documents created by a teacher.**

In the case where a teacher creates documents, she should be able to upload them to some repository (see Use-Case A-07). An interesting question is whether the teacher can upload personalized documents for a particular student. In this case we need a notification mechanism that will inform the student about the new document.

- **An offline version of the Reader-app.**

Whether an offline version of the Reader-app will be available and its exact functionality will be described in deliverable **D7.1: iRead Reader app Interaction and Visual Design (M12)**. Same issues with Use-Case A-03 regarding an offline version would apply here.

- **Quitting the application.**

The case where a student quits the application without logging out is the same as in Use-Case A-03.

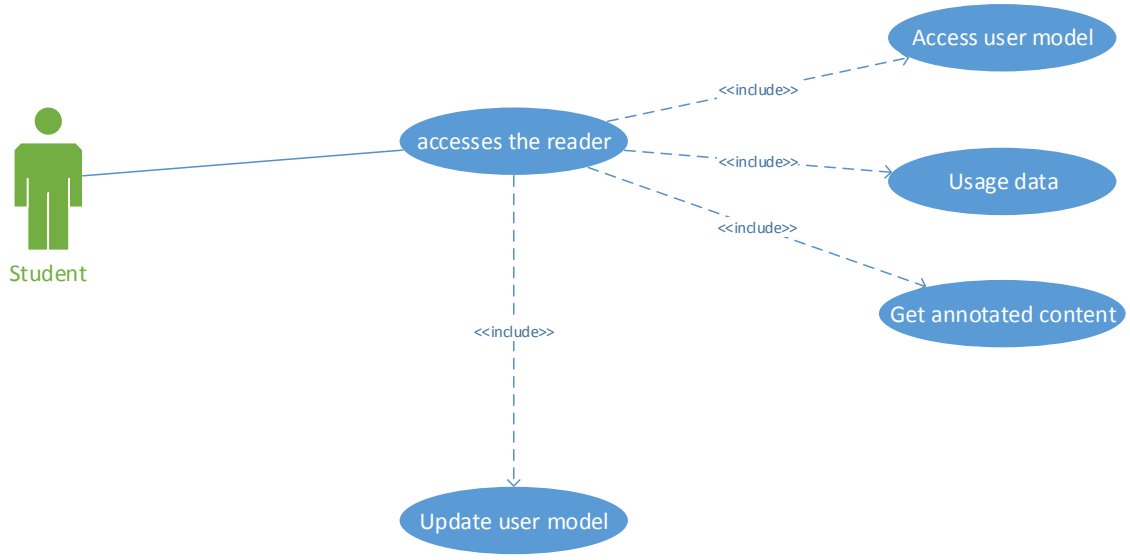

Figure 4: UML diagram for the reader application

A-05: "Student overviews her progress"

This is a very vague description as the functionality of the monitoring-app and of the analytics module is to be defined.

1. Student logs in a web-app that allows for an overview of her progress
2. Student gains access to her user model and usage-data
3. Student views a heat map (or another visualization) of her user model and model-properties she has mastered or needs to work with
4. Student logs out the web-app

Notes

- User model is assumed to be initialized (or active).
- User model is assumed to be stored in the "infrastructure".
- Usage-data is assumed to be stored in the "infrastructure".
- Each student can view only her profile.

Issues/Questions

- **Age-sensitive web-app.**

The web-app should be age-sensitive. This is extremely important for the representation of the user-model and the available analytics.

- **Graphical presentation.**

Various graphical representations of a user-model should be examined.

- **Usage-data.**

Are there other information, related to the student's usage-data, that the student will be able to view (statistical, analytics, etc.)?

- **Parent's usage.**

A student's progress could be available to her parents (guardian) as well or only. In the latter case, the parent could login the web-app with the student's credentials. Another option would be to define a type of "parent" user, having access to the student's user-model and usage-data.

A-06(a): "Teacher monitors a student's progress"

This is a very vague description as the functionality of the monitoring-app and of the analytics module is to be defined. This use-case is very similar to Use-Case A-06(b) that follows, and the same notes and issues apply.

1. Teacher logs in a web-app that allows for the monitoring of students (in her class)
2. Teacher sees a list of students and selects the student of interest
3. Teacher gains access to the student's user model and to usage-data of the student. Other data may be also available (statistical, analytics, advice, etc.)
4. Based on the available data, the teacher plans future activities for the individual student
5. Teacher logs out the web-app

A-06(b): "Teacher monitors the progress of her class/school"

This is also a very vague description as the functionality of the monitoring-app and of the analytics module is to be defined.

1. Teacher logs in a web-app that allows for the monitoring of her class/school
2. Teacher sees a list classes/groups she is responsible for and has access to
3. Teacher gains access to usage-data, statistical data, analytics, etc. for the groups in question
4. Based on the available/presented data, the teacher plans future activities/actions for the class
5. Teacher logs out the web-app

Notes

- Teacher is assumed to be registered in the system.
- Usage-data is assumed to be stored in the "infrastructure".
- Each teacher has access privileges and can monitor specific students.

Issues/Questions

- **Access control and user privileges.**
  Each teacher has access to specific students (in her class). A special educator-expert may have access to children she works with (not necessary in a class). A module that gives access privileges to the user-models of children is required. Read-only access may be appropriate in some cases. Access lists may be a simple way to implement. It seems that this should be integrated in the infrastructure.
- **Graphical presentation of analytics.**
  The user-model, usage-data, statistical data, etc. should be presented in a user friendly and easy to understand manner. A graphical representation should be used whenever possible.
- **User details.**
  Will the teacher be able to change the student's details or reset her password?
- **Analytics: Infrastructure or client?**
  The analytics seem to be application specific. Are they part of the infrastructure or of a client application? We should decide whether the visualizations/graphical presentations are part of a library which is made available to client-apps, or whether we leave it to the client-apps to deal with data presentation (without any support). We will have to develop at least some client apps.

A-07: "Teacher selects learning material for a specific student"

In the following scenario, a teacher browses through a repository of books and selects learning material for a student based on her user-model.

1. Teacher logs in a web-app that allows for the monitoring of her class/school
2. Teacher sees the list of students she is responsible for and has access to

3. Teacher selects a student
4. Teacher selects a book repository containing several books/texts/docs (she has placed earlier)
5. The documents are ranked based on their difficulty for the specific student
6. The teacher selects an "easy" book for the student (to be read later by the student using the Reader-app)
7. The book is made available for the student when she logs in the Reader-app
8. Teacher logs out

Notes

- Teacher/student are assumed to be registered in the system.
- A book/text/doc repository is assumed to exist.
- A text classification module is assumed to exist and used for document ranking.

Issues/Questions

- **Integration of repositories.**

The scenario assumes an existing repository of books/docs or a repository containing material selected by a teacher. A mechanism for integrating repositories should be examined. How do we deal with large collections? It seems too hard to process them on-the-fly. Some preprocessing may be necessary, while small collections may be processed on-the-fly. An option of preprocessing could be to "locate" the occurrences of model-properties in each text, and evaluate the metric formula given a user-model on-the-fly.

- **Book is made available at the Reader-app.**

In steps 6 and 7 the teacher selects a document for a particular student to be read with the Reader-app. For the document to become available on the Reader-app when the student logs in, a notification mechanism is required.

- **Supported format of documents** has to be specified.
- **Personalized file-explorer app.**

This use-case suggests a personalized "file-explorer" app. It can be similar to windows explorer with an extra column, i.e., "difficulty". Files of "document-type" can be sorted based on "difficulty".

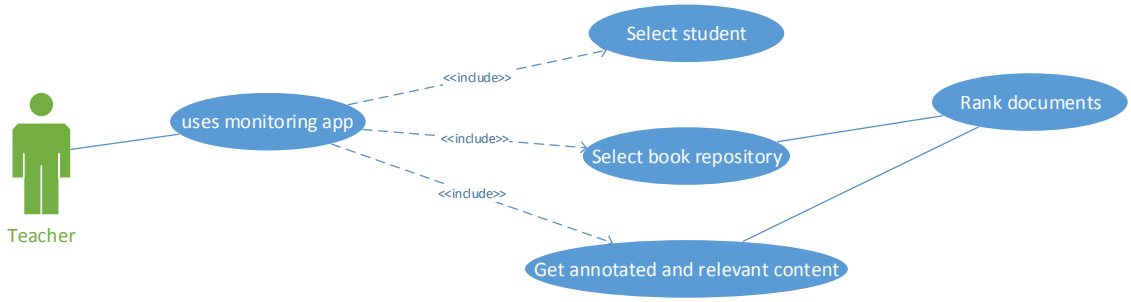

Figure 5: UML diagram for assisted selection of reading material

Category B: Independent use of iRead

The last pilot for the iRead evaluation will allow for unattended use of iRead through the internet. Cases A-01 to A-05 apply for a user who enrolls the pilot.

Issues/Questions

- **User registration.**
A platform for registration of "independent" users should be available.
- **Types of users, access control and user privileges.**
If we allow "independent" use for a teacher or expert that can register and monitor students, questions and issues similar to Use-Cases A-06(a) and A-06(b) apply.
- Another general question is who decides about **deleting a user from the system** and how this could be accomplished.

Category C: External apps using existing user-models

Common to all use-cases of this category, is the fact that an external-application must get access to the user's user-model. For example, a "personalized file-explorer" could sort documents based on their difficulty. The ranking of the documents depends on the values of a user's user-model, therefore access to the user-model is required. Similarly, an eBook store or a publisher could suggest appropriate reading material for a student based on her user-model.

C-01: "Personalized File-explorer"

1. User opens personalized File-explorer app and logs in
2. The Personalized File-explorer gets access to the user's-model
3. The user browses a "file system". In each folder, each document has a "difficulty score" based on the user's user-model. Documents can be ranked based on the difficulty score
4. The user selects a document for further analysis
5. An interactive difficulty heat map may appear that contains entries for each user-model entry (weighted by the corresponding severity in the user's-model). Clicking on an entry, highlights the relevant parts of the text (other functionality may be possible)
6. The user makes the document available for viewing with the Reader-app
7. User exits (logs out)

Notes

- The details of the app (interactivity/visualization/functionality) are to be specified.
- The app can support an eBook store that supports personalized search functionality.

Issues/Questions

- **How does the app register to use the iRead infrastructure?**

We should have a way to control what apps access the infrastructure. A registration mechanism may be required. Are there other solutions? Security and privacy should be addressed accordingly.

- **Access to the user's user-model.**

This is extremely important. Possible solutions we should investigate include:

  - The user provides her "user-model" to the app. The user-model may be supplied in the form of a file/cookie/token that has been exported from the iRead infrastructure or from the application which generated it.
  - The app contacts the iRead infrastructure to request the user's user-model. The user is redirected to the iRead infrastructure and is required to be authenticated. After successful authentication, the user-model is sent to the app. This solution is similar to "federated login".

- **Which content classification?**

We should decide if the application makes use of the iRead content classification (by calling services) or if it should implement its own.

- **How is the content classification applied?**

This may be a hard task to be done on-the-fly for all documents. Severity scores may be computed for a set of "representative" user-models. Or, the app may "index" files and preprocess them in a similar way as proposed in user-case A-07. We may also have a specific folder where a user can drop documents to be ranked.

- **Compatibility with the Reader-app.**

Will the documents be available for viewing with the Reader-app (step 6)? This would imply that a user can manage his own repository. Similar questions with user-case A-07.

- **Support for teachers.**

The File-explorer could also support teachers (or anyone registered in the iRead system and authorized to monitor students) who want to browse documents for a particular student and then upload them to their repository or suggest them to the student.

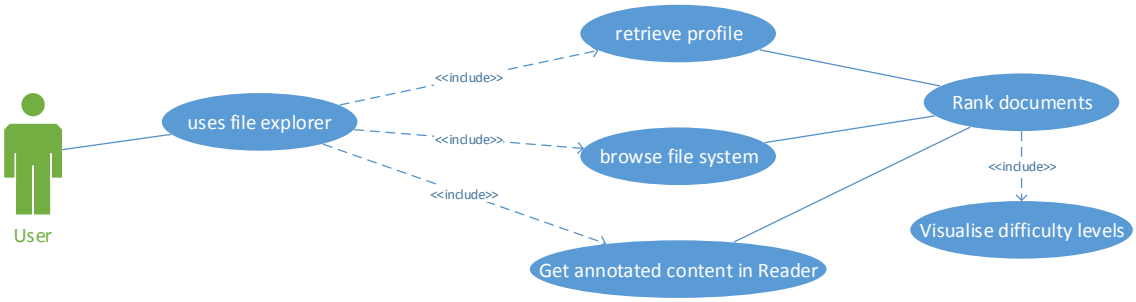

Figure 6: UML diagram for independent use of a file explorer app

C-02: "eBook store with personalized search functionality"

This use-case is an extension of C-01 and the same questions and issues apply.

1. User logs in the eBook store app
2. The eBook store gets access to the user's user-model
3. The user issues a search query to the bookstore collection
4. The query results are ranked and presented in "easiest-to-read first" order based on the user's user-model (the ranking used the "difficulty score" of the books)
5. The user selects an eBook from the query results and details on the "difficulty/easiness of reading" are displayed (in some interactive graphical form)
6. User logs out

Notes

- The details of the app (interactivity/visualization/functionality) are to be specified.

Issues/Questions

- **How does the app get access to the user's user-model?**

The same questions as in the Use-Case C-01.

- **Privacy issue: Further use of user-models**

The user-model provided by the user to the eBook store should not be further used. Ownership is with the user, not with the eBook store. Further use by the eBook store (to make reading suggestion for the future) should be authorized by the user. Note that the user- model may not be accurate after some months, since it is evolving with the reading skills. Will the eBook store keep the user's user-model? Another alternative would allow the eBook store to access each time the user's user-model.

- **Content classification**

Similar to Use-Case C-01.

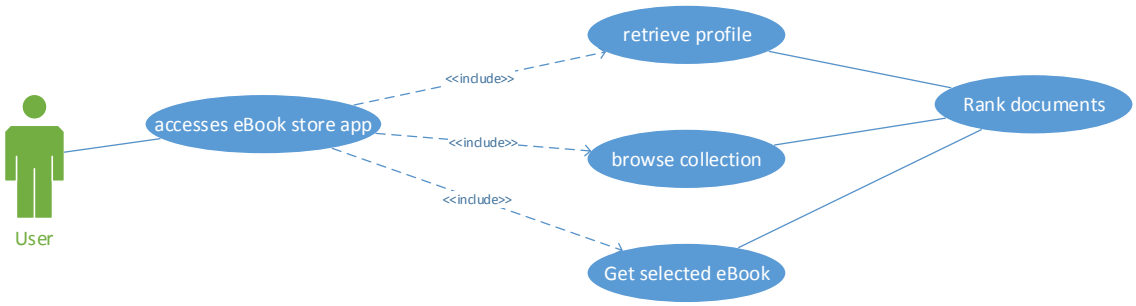

Figure 7: UML diagram for assisted eBook selection

C-03: "Personalized meta-search Engine"

A Personalized meta-search Engine could process the result of queries over a search engine, by filtering and reordering the result. Search queries could be related to reading material that can be ranked based on the user's user-model. For example, the user could search for pdf files or eBooks over the internet. As another example, the user could search for news and view the results ordered by reading–difficulty, or get suggestions based on her user-model. The functionality of a Personalized meta-search Engine is similar to Use-Case C-02. The privacy issue seems to not apply.

C-04: "Collaboration of School with eBook publisher"

1. A school adopts iRead and enrolls several of its students
2. The students start using iRead and their user-models get updated
3. The school comes to an agreement with an eBook publisher to provide personalized reading suggestions to the students (in the form of email)
4. The school provides the eBook publisher with access to the students' user models and provides email addresses for their parents
5. The eBook publisher starts emailing to the students' parents reading suggestions for their children. This is repeated at regular time intervals or when the user-model presents some significant change (improvement)

Notes

- We assume that the school has consent to provide the user models to the publisher.
- We assume the school has consent to provide the emails and the publisher to send book advertisements (i.e. not spam).

Issues/Questions

- **Access to the user's user-model.**

The same questions as in the Use-Cases C01-03 regarding how the app get access to the user-models.

- **Privacy issue: Further use of user models.**

The ownership problem is evident here. Does the user-model belong to the user or to the school who created (and who is probably paying subscription to iRead)? Can the user access it? What if it contains proprietary info? Proper consent (from parent or guardian) must be given. Privacy law rights (denial/erasure/etc.) should be respected.

- **Content classification.**

Similar to Use-Cases C01-03.

- **Is the eBook publisher "registered" as "approved" by the system?**

We should examine whether the publisher will be somehow registered in the system, and how this could be done. Possible solutions could include approval by

the consortium or automatic approval in the case a user allows access to her user-model.

Category D: Extending the iRead system

The purpose of iRead is to support the reading process for seven languages (including English as a foreign language). Domain-models, user-models and language resources will be developed for these languages, along with "learning components" (literacy games, Reader-app). The iRead system's design could allow several extensions, such as: definition and support of new language domain-models or other domain-models (not necessarily related to learning or reading), enrichment of existing language resources, incorporation of new activities and literacy games, etc. Use-cases of this category are extremely important since they will define the limits of the iRead infrastructure.

D-01: "iRead supporting new domain models"

Consider the following scenario. A company which is involved in "personalization" develops a new domain-model and want to use iReads infrastructure to support its operations (storing/accessing/updating/etc. the user-models). The new domain-model may be relevant to learning (e.g., preparing for the LOWER/GMAT exams, or learning how to fly a plane) or to other activities (e.g., tracking performance levels and heart-beat measurements during exercise).

This use-case is extremely important for the extensibility of the system. We should decide whether this is within the scope of the project. Here we set the limits of the iRead project.

1. A company develops a new domain model, say X-domain-model
2. The company decided to use iRead to make its services available to users. It installs the "dedicated server version" or gets access to the "shared" installation
3. The company enters the X-domain-model into iRead
4. The company creates user-models based on the X-domain-model
5. The X-domain-model goes into operational mode. Use-cases analogous to the ones described in the previous sections apply.

Notes

- We assume that a way to define a new domain-model is available and supported by iRead.
- We assume that a "dedicated server version" of iRead exists.
- We assume that the "infrastructure" of iRead supports the addition of new domain-models.

Issues/Questions

- **Domain-model description.**

How do we describe/define a new model? Do we use an XML-style description? We should determine the rules that the description/definition of a new domain-model needs to follow. We could also offer an editor for creating domain-models that are "compatible" with iRead system.

- **Mechanisms for introducing new domain-models into iRead.**

We should define the mechanisms with which new domain-models are introduced into iRead and, then, how they are made available for instantiation when creating user-models.

- **Availability of new domain-models.**

If an existing user want to register for the new domain-model as well, how is login achieved? This issue is also present in Use-Case A-01.

- **Update of new user-model.**

For a general domain-model a mechanism for updating the values of the corresponding user-models must somehow be defined and integrated within the model. Where should this "logic" reside? If not in iRead system, updating a user-model should be possible by appropriate services.

- **Resources.**

The new domain-model could require a set of resources. Does the iRead system support addition of these resources and are they accessible for free?

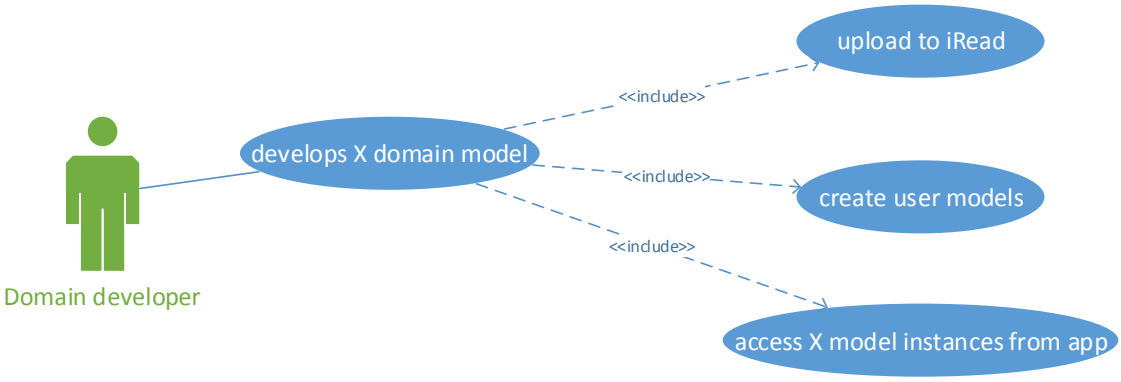

Figure 8: UML use case diagram for adding a new domain model

D-02: "Add a domain model for a new language"

The iRead system may be extended by adding, for example, language domain-models for French, Italian, and other languages. Other possible extensions may include adding dyslexia domain-models for languages other than English/Greek, and adding "Learning Foreign/Second Language" domain-models (other than EFL).

1. Linguists develop a new domain-model, for some language X, say X-model
2. Linguist develop language resources for to support the domain for language X
3. The X-model is incorporated to iRead

4. The language resources are incorporated to iRead (or made available to the user-apps of iRead)
5. X-domain model becomes operational. All previous use-cases may be applied.

Notes

- We assume that the X-model is defined and introduced in the iRead infrastructure by following Use-Case D-01. Maybe for the case of new languages, a more "restricted" plan could be applied.

Issues/Questions

- **Language resources.**

Are language resources part of the iRead infrastructure? Different language resources may be required by each language domain-model and they may be supported by different services. It may make sense to have language resource not as part of the iRead infrastructure but as part of the infrastructure that supports and external app. On the other side, since iRead is focused on language learning, we should support at the project all existing language domain-models, but not as part of the "core" infrastructure.

- **Model description.**

How do we define a new language model? Do we offer an editor for creating language model?

- **Update of new user-model.**

For the new language domain-model, the mechanism for updating a user-model should be determined.

- **Availability of new language model.**

How is the new language domain-model made available for existing or new users?

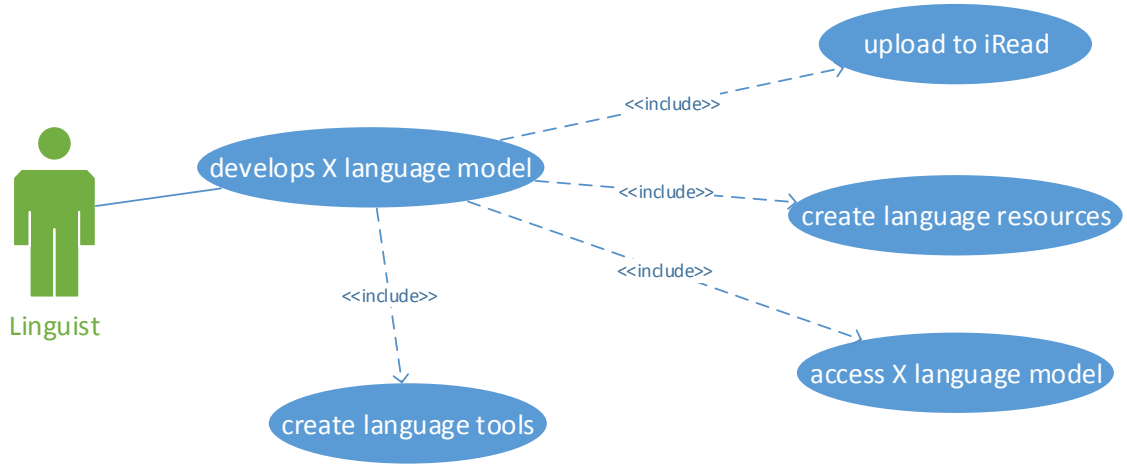

Figure 9: Use case diagram for adding new language models

D-03: "Add or update language resources for existing domain models"

This case is similar/related to D-01 and 02. Updating the available language resources is important for updating the system. Care should be taken so that there are no compatibility problems between version of iRead or iRead apps. The exact mechanism for updating resources has to be specified.

D-04: "Update the iRead Games by incorporating new activities"

This case is also similar/related to D-01 and 02 and important for updating the system. Compatibility issues between version of iRead or iRead apps should be avoided. A possible solution could be to make an "update" available for the games-app so that a user could download the new version.

D-05: "New Game that wants to uses existing user models"

The new game is developed by new vendors who want to gain access to existing user-models and usage-data. We should study what kind of access will be provided to them (Read/Write/Update/append) on a user-model, a user's usage-data and storing usage-data in the system. Care should be taken so that there are no compatibility issues between version of iRead or iRead apps. Will the new game be somehow registered in the iRead system (similar with issues of C-01)?

D-06: "New type of user is inserted"

In iLearnRW[2] we had three roles: "student", "teacher" and "expert" forming a hierarchy. There may be need for new roles (e.g., parent, child, school director, data analyst) each requiring access to different set of data (user-models and /or usage-data). How do we insert them in the user hierarchy? It is really a hierarchy?

D-07(a): "A New metric is needed (for analytics)"

In the case where a new metric for analytics has been defined and needs to be incorporated in the system. How do we introduce it? Do we just rewrite the analytics app?

D-07(b): "A New metric is needed (for content classification)"

Similar to the previous case, we should investigate how a new metric for content classification could be made available to all components and apps that use iRead "infrastructure's" content classification.

[2] EU FP7 ICT project iLearnRW - Integrated Intelligent Learning Environment for Reading and Writing (project number: 318803)

D-08: "Provide access to language resources/services (without authentication)"

This can be similar to the iLearnRW Online Resource Bank. The resources could be simply made available for non-authenticated users/apps. For example, someone may want to make use of the syntax analysis services. Other solutions to consider may include to make them available

- For free.
- For free for "authorized" or registered apps and third parties.
- As a limited service, e.g. limited number of calls per minute.
- As a pay-for service.

Category E: Automatic update of iRead system

In this section we consider the mechanisms for updating the values of a user-model and the domain-model. As a student makes use of the literacy games and the Reader-app, progress is made and various reading skills are mastered while others become available. The user-model should reflect this progress by appropriately changing its values. On the other hand, usage-data stored in the iRead system, along with teachers' selections, will be used to update the selection-mechanism of next activities for a domain-model.

E-01: "Update user-model"

We should determine the mechanism for updating a user-model. The representation of the domain-model and the corresponding user-models are key-elements for this process.

Issues/Questions

- **Nature of process.**
  Will the teacher be authorized to manually change the user-model values?
  How will the process be informed by teacher-expert's findings?
- **Default user-model.**
  In the case of using a set of predefined default user-models for initializing a user-model, the initial values of the user-model may not reflect correctly the reading abilities of the student. Hence, a "fast update" would be required so that the user is not discouraged (see also user-case A-02).

E-02: "Domain-model update"

Apart from representing the learning process, the domain-model includes the "logic" for selecting appropriate activities and learning material for a user based on her user-model. The selection of next-activity in the literacy games is essential for the games to be interesting.

Issues/Questions

- **Domain-model representation.**
  The representation of a domain-model is crucial for defining its update mechanism.
- **Adaptation of domain-model.**
  Domain-models will be developed for all seven languages supported by the iRead system. The details, criteria and exact mechanisms for "updating" domain models will be specified in deliverable **D4.4: User Adaptation Component (M24)**.

E-03: "Processing for analytics"

Analytics will in general process the usage-data stored in iRead system. All apps (literacy games, Reader-app, external apps, etc.) will continuously add usage-data to the system. For experts that will use analytics in order to evaluate the system and make decisions, a preprocessing of the usage-data is required. The question that arises is whether this is a continuous process or a scheduled process, e.g. once per six months.

Category F: Other uses of iRead and its infrastructure

F-01: "Dedicated server iRead installation"

A school wants to install its own dedicated server installation. In this installation, all data and services are stored/run at the school's server. This may be required for privacy/security reasons. This use-case is extremely important, especially for the iRead incubation hub.

The school administrator downloads the "server version" of iRead
iRead is installed at the school's server and is started
Local-iRead becomes operational. All previous use-cases may be applied

Notes

- We assume that a server version of iRead is available for download.
- We assume school has server facilities and IT personnel.

Issues/Questions

- **Development of dedicated server versions of iRead.**
  A "downloadable" should be made available with appropriate instructions. The installation should be similar, for example, to that of installing and operating a file server or a web-server.
- **Update.**
  When new resources are available, or new metrics are added to the system, or when the updating mechanisms of Category E are changed, an update could be made available.


iREAD

5. CLIENT-SERVER ARCHITECTURE

The convenience of using a server has already been indicated in the previous sections. Here we describe and briefly analyse the usefulness of implementing a client-server architecture to support various aspects of the iRead system, even in the case of offline operation.

Advantages of a server-based architecture

A central server can provide to iRead the following functionalities:

Ability to store a large data collection (Resource Data Bank). If a server is included in the system, we can store a variety of multimedia data, metadata, and user-related information in it. The stored material can be used by the applications that run on the client side. This design choice also provides us the ability to make scheduled or incremental updates to the user profile, which are then synchronised with all instances.

Centralised control of the users' data and progress. All user analytics and usage history can be stored (most likely in encrypted form) in the server in order to be accessible from authorised users of the system. A GUI will be developed in order to give access to the teacher/expert/parent responsible for each child, helping them to monitor each child's progress and the activities that it engages in. The teacher/expert can make adjustments to the child's learning strategy in order to provide him/her with activities that are more relevant to his/her profile.

Single user – multiple devices. In those use cases where a particular user may access different devices to interact with iRead, their data must be saved on a remote server and synchronised with the device at start-up.

Regular profile updates. By saving the user profiles and all historical data on a remote server, a teacher/expert is able to manually update a profile when they see that a learner's performance (based on their usage history) has changed. The updated profile will become available to all devices that utilise the particular profile.

Perform computationally expensive tasks on the server rather than the client side. A server that runs on a powerful machine can be helpful and spare the client side (i.e., a tablet) form executing difficult and resource demanding tasks. An example of this usage is the dictionary queries that will take place throughout a client session. If the dictionary is stored on the server, we can develop special software (including a web based API) that can provide to a client fast access to it.

Space, Time Trade-off. By including a powerful machine as a server in our project we have the ability to reduce the cost of some demanding tasks that take place throughout a client session. This can be achieved by pre-computing and storing (maybe large) data in our server that can help it perform fast computations on specific problems that arise at the client side; dictionary processing and annotations are an example of this trade-off.

User Authentication. At its simplest form, user authentication is required in order to load the correct profile for each child using the iRead system through a device. It can be also used as an access control mechanism giving access to each user only to the data the user is entitled to. Data necessary for user authentication are sensitive and require appropriate data protection measures. For the passwords, only their cryptographic hashes will be stored.

System architecture

An outline of the system architecture in iRead is presented in the following figure.

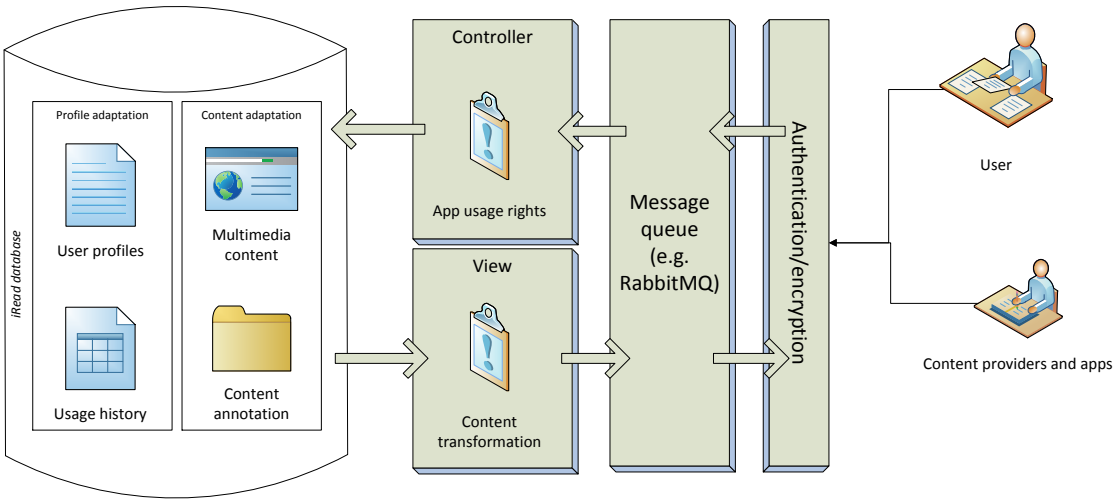

Figure 10: Abstract system architecure

In this figure, data flow is initiated from iRead users (students/experts and content providers), who issue a request for content to be used in the apps or interact with content already downloaded from the server. This request goes through two validation phases, one related to user or app authentication and one to usage rights, in order to prevent data leaks and unauthorised access to the sensitive data in the iRead database. Since we expect the system to receive hundreds of simultaneous requests during school hours and in order to balance the relevant server load, every request by the users is added to a message queue and, in turn, filtered via the controller of the system. If the request is valid, the requested data is retrieved from or added to the iRead database (e.g. in the case of uploading success or failure during an activity) and transformed to comply with the characteristics of the particular user profile (e.g. adding mark-ups needed for highlights in a reader application) and the application. Finally, the annotated data is returned to the requesting application for consumption.

Two (possibly) offline processes are documented for the iRead database. The first (profile adaptation) may be scheduled in regular intervals (e.g. every Friday afternoon for school users), whenever cumulative successful or unsuccessful interactions with the activities pass a predetermined threshold, triggering a profile update in a positive or negative manner, or when experts who view the user analytics decide that a manual change in the user profile needs to be introduced. The second offline process, content

adaptation, has to do with processing content uploaded to the server, which will be used in the iRead games and applications, such as the reader; these applications, in order to fulfil their learning objectives, need to receive textual information with associated metadata which correspond to its syntactic and semantic content. As mentioned earlier, this process takes place only once for any passage, web page or dictionary uploaded to the server, and the produced metadata are associated with the content and used in conjunction with it; in iRead, this will take place during the setup of the server, for existing annotated dictionaries and texts, while newly created dictionaries will be annotated as soon as the relevant user profile and syntactic analysis tools become available.

Other processes that run on the server side include content selection, based on user profile and possibly expert or user choices, and preparation of user analytics for the expert, parent, and user. Piwik (https://piwik.org/) is a candidate for the latter component, while Apereo (https://www.apereo.org/projects/openlrs) is a product closely related with learning management systems.

6. SYSTEM SPECIFICATIONS

From the previous sections we have indicated that the iRead system may be considered as a client/server application. Data will be stored in a central "project server", or servers maintained by project partners, in the form of user profiles (and data used to construct them). These profiles will be password and/or IP address protected. Thus, no unauthorized person in the scope of the project will have access to these data, while parents or guardians may opt out of the system and its databases at any time. A child should be able to use any compatible device (at school or at home) that has the iRead software installed (client).

Basic indicative information exchanges between client and server include:

a) User logon on server (Client → Server).
b) Server acceptance or rejection of the user connection (depends on login/password and IP address) (Server → Client).
c) Corresponding profile is loaded (Server → Client).
d) Information flow from a computer to server (updating a user profile) (Client → Server).

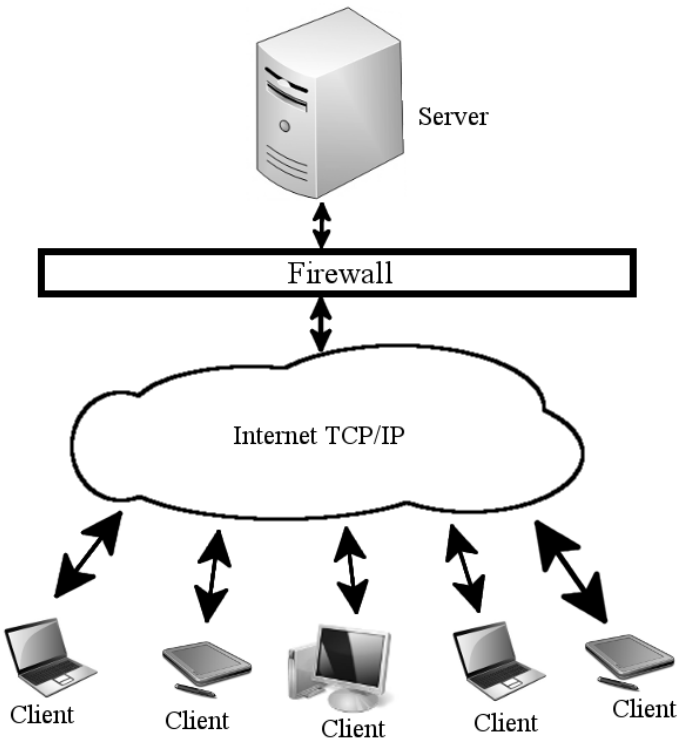

Figure 11: Server/Client Architecture

Hardware – Software Specifications

Server

As mentioned in the proposal, the consortium enjoys the opportunity to benefit from the cloud infrastructure of Okeanos, offered by the Greek Research and Technology Network (GRNET). Okeanos is dubbed as "Infrastracture as a Service", meaning that subscribers may build their own virtual computer, which is always connected to the Internet, without worrying about hardware failures, connectivity issues, and software problems. This includes the ability to create Virtual Machines and Virtual Networks (useful for secure interchange of data), store files online, share them with users and colleagues and access them anytime.

Since the server side of the iRead infrastructure will have to deal with incoming requests for data (user profiles and content), as well as perform additional tasks in real time and offline (content filtering, content processing and classification, profile updates, etc.), we will reserve the following resources in Okeanos (adding more resources is possible via a point-and-click interface, but we want to test a realistic system to cater for better exploitation after the project ends):

- Virtual machines: 12
- Hard Disk Storage: 4TB
- RAM: 128GB
- CPUs: 32

Client (Tablet/desktops/laptops)

Based on the requirements of existing client-side applications similar to the ones that will utilise the iRead infrastructure, and on the amount on processing that needs to be performed at the client side, the minimum requirements for tablet clients are as follows:

- Screen size: 9.7 inches
- Screen resolution: at least 1280x800
- Operating system: Android 5.0
- RAM: 2GB
- Storage space: 16GB expandable with SD cards
- WiFi/LAN connectivity

The recommended specifications for tablets are

- Screen size: 9.7 inches
- Screen resolution: at least 2048x1536
- Operating system: Android 6.0+
- RAM: 4GB
- **Storage space: 32GB** expandable with SD cards

- WiFi/LAN/**3G/4G** connectivity

Items in **boldface** are those different than the minimum requirements. For Apple devices, the minimum requirements are those of Apple iPad 32GB Wifi, while the recommended are those of Apple iPad Air 2 32GB 9.7" 4G.

In the case of desktop computers, the minimum requirements are:

- CPU: Intel Core i3 or equivalent
- Cores: 2
- CPU clock: 3.90 GHz
- RAM: 4GB DDR
- Hard disk storage: 1TB
- Graphics card: Intel HD 630
- Screen: 24 inch
- Screen resolution: 1920 x 1080

while in the case of laptops:

- CPU: AMD E2 7110 or equivalent
- Hard disk storage: 500GB
- Graphics card: onboard
- Screen: 15 inches
- Screen resolution: 1920 x 1080

Interoperability

Interoperability is the ability of a system to provide services to and accept services from other systems and to use the exchanged services so that the systems effectively work together, e.g. a research oriented application use system data for specialized analysis. In order to achieve high degree of system interoperability, the use of de facto protocols and models of storage is required. The last is also important for treatment and transmission of information. More precisely, the following are required:

a) A pre-determined form for information storage (standards of data and meta-data formalization).
b) A pre-determined way for information exchange (communications and protocols technologies where information is transmitted according international standards (a text file with standard format, etc)).
c) A pre-determined way for data access.
d) A pre-determined way for data organization (technologies of metadata etc.).
e) A pre-determined way for secure data access for third-party applications.

System Security

The data security issues in the final system are those of a typical client/server application. The issues concern the unauthorized access of the server and the

transmission of application data. Both of these issues will be addressed applying standard available software solutions (use of encryption/decryption techniques, allowing access only to specific client server applications, firewalls, etc). The exact mechanisms to be employed will be decided during implementation time (cf. Deliverable D3.2 for more information) and will depend on design decisions concerning the actual components of the software system.