Contents

I User Requirements Document 6
1 Kinds of ISAC users 8
2 General requirements 9
3 Requirements of the visitor 11
4 Requirements of the learner 12
5 Requirements of the math author 16
6 Requirements of the dialog author 18
7 Requirements of the course admin 20
8 Requirements of the course designer 23
9 Requirements of the administrator 26

II Software Requirements Document 27
10 General requirements 29
11 The worksheet 30
12 The browser generators 31
   12.1 The theory browser generator .............................. 31
   12.2 The problem and method browser generators .............. 32
   12.3 The example browser generator ........................... 32
13 The knowledge browsers 33
   13.1 The theory browser .......................................... 34
   13.2 The problem and method browsers ........................ 34
   13.3 The example browser ....................................... 35
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>The dialog guide</td>
<td>37</td>
</tr>
<tr>
<td>15</td>
<td>The mathematics engine</td>
<td>39</td>
</tr>
<tr>
<td>III</td>
<td>Architektural Design Document</td>
<td>40</td>
</tr>
<tr>
<td>16</td>
<td>The Core System</td>
<td>41</td>
</tr>
<tr>
<td>17</td>
<td>The Knowledge Browser</td>
<td>43</td>
</tr>
<tr>
<td>17.1</td>
<td>Structure</td>
<td>44</td>
</tr>
<tr>
<td>17.2</td>
<td>BrowserFrontend</td>
<td>45</td>
</tr>
<tr>
<td>17.3</td>
<td>information-processor</td>
<td>46</td>
</tr>
<tr>
<td>17.4</td>
<td>browser-dialog</td>
<td>46</td>
</tr>
<tr>
<td>18</td>
<td>KE-Store</td>
<td>48</td>
</tr>
<tr>
<td>18.1</td>
<td>XML-Import/Export</td>
<td>49</td>
</tr>
<tr>
<td>18.2</td>
<td>Communication with the <em>dialog</em></td>
<td>50</td>
</tr>
<tr>
<td>18.3</td>
<td>Relations</td>
<td>51</td>
</tr>
<tr>
<td>18.4</td>
<td>Presentation</td>
<td>51</td>
</tr>
<tr>
<td>18.5</td>
<td>Example Collections and composite examples</td>
<td>52</td>
</tr>
<tr>
<td>18.6</td>
<td>object structure</td>
<td>53</td>
</tr>
<tr>
<td>18.6.1</td>
<td>hierarchy</td>
<td>56</td>
</tr>
<tr>
<td>18.7</td>
<td>metadata</td>
<td>56</td>
</tr>
<tr>
<td>19</td>
<td>dialog</td>
<td>58</td>
</tr>
<tr>
<td>19.1</td>
<td>Session-Dialog</td>
<td>60</td>
</tr>
<tr>
<td>19.2</td>
<td>Browser-Dialog</td>
<td>60</td>
</tr>
<tr>
<td>19.3</td>
<td>WorkSheet Dialog</td>
<td>60</td>
</tr>
<tr>
<td>19.4</td>
<td>User-Administration</td>
<td>61</td>
</tr>
<tr>
<td>19.4.1</td>
<td>user</td>
<td>61</td>
</tr>
<tr>
<td>19.4.2</td>
<td>course</td>
<td>62</td>
</tr>
<tr>
<td>19.5</td>
<td>permissions-module</td>
<td>62</td>
</tr>
<tr>
<td>19.6</td>
<td>KE-Objects and external Informations</td>
<td>64</td>
</tr>
<tr>
<td>IV</td>
<td>Software Design Document</td>
<td>65</td>
</tr>
<tr>
<td>20</td>
<td>The Worksheet</td>
<td>66</td>
</tr>
<tr>
<td>20.1</td>
<td>Elements of user-interaction</td>
<td>66</td>
</tr>
<tr>
<td>20.2</td>
<td>Data stored and manipulated by the WDG</td>
<td>67</td>
</tr>
<tr>
<td>20.2.1</td>
<td>Draft for an user-model</td>
<td>67</td>
</tr>
<tr>
<td>20.2.2</td>
<td>Dialog state</td>
<td>68</td>
</tr>
<tr>
<td>20.3</td>
<td>The object world of the worksheet component</td>
<td>68</td>
</tr>
<tr>
<td>20.3.1</td>
<td>The dialog guide object</td>
<td>68</td>
</tr>
<tr>
<td>20.3.2</td>
<td>The user model object</td>
<td>69</td>
</tr>
<tr>
<td>20.3.3</td>
<td>The presentation layer object</td>
<td>70</td>
</tr>
</tbody>
</table>
20.3.4 The proof tree object ......................................... 70
20.3.5 The proof element selector object ............................. 72
20.3.6 The formula object ............................................. 72
20.3.7 The subterm selector object ................................... 73
20.3.8 The tactic object ................................................ 73
20.3.9 The assumption object ......................................... 73
20.3.10 The error class object ......................................... 73
20.3.11 The problem specification object ............................ 73
20.3.12 The menu object ............................................... 74
20.3.13 The worksheet highlight object ............................. 74
20.4 Interactions of the worksheet component with the outside world
as used in the ISAC system ........................................... 74
20.4.1 Communication between WPL and WDG .................... 74
20.4.2 Communication with the Mathematics Engine ................ 75
20.4.3 Communication with the Knowledge Browser ............... 76
20.4.4 Draft for an interaction diagram ............................... 76

21 The knowledge browsers 79
21.1 Browser Servlet .................................................. 79
21.1.1 request a problem-description ................................. 80
21.1.2 needed methods ............................................... 81
21.2 Information encoder ............................................. 81
21.2.1 needed methods ............................................... 82
21.3 Information processor ........................................... 83
21.4 browser-dialog ................................................... 84

22 description-storage 85
22.1 description-object ............................................... 86

23 The Dialog 87
23.1 create a new session .............................................. 87
23.1.1 needed methods ............................................... 87
23.2 open a new browser (–window) ................................. 88
23.2.1 needed methods ............................................... 88
23.3 open a new worksheet ........................................... 89
23.3.1 needed methods ............................................... 89

24 The browser generators ............................................. 90
24.1 ................................................................. 90
24.2 ................................................................. 90
24.3 ................................................................. 90
24.4 ................................................................. 90
24.5 ................................................................. 90

25 The example browser and generator ................................ 91
V  Usecases  

26 Enter an ISAC site  

27 Use the mathematics knowledge base  
   27.1 Browse the knowledge  
   27.2 Access other components of ISAC  

28 Do math on the worksheet  
   28.1 The student starts a calculation  
   28.2 The student does a step in a calculation  
   28.3 Subproblem with maximal user-guidance  
   28.4 The student calls KB-items from the worksheet  

29 Authoring  
   29.1 Author the knowledge base  
   29.2 Author the example collection  

30 The example for reference  
   30.1 Description, formalization and modeling phase  
   30.2 Knowledgebase and specification phase  
   30.3 Interaction on the worksheet and the browsers  
   30.4 The solving phase and subproblems  

VI  Subprojects  

31 SML-XML encoders  

32 XML-design for math web-data  

33 HTML-encoders for math documents  

34 Integrated editor for math web-data  

35 A selection-tool on hierarchies  
   35.0.1 navigation-tree  

36 XML-design for math examples  
   36.0.2 layout-hints  

37 MathML en/decoder  

38 Integrate a formula editor into SWING  

39 Co-operate in the development of a formula editor  

40 Dinopolis-wrapper for ISAC's math engine
41 Implementation of the basic classes 131
42 Worksheet 132
43 Dialog 133
44 User model 134

VII Appendices 135
   A List of terms used in the ISAC-project 136
   B ISAC’s tactics 143
   C Abbreviations 145
   D Development environment 146
Part II

Software Requirements Document
This Software Requirements Document is structured along the modules abstracted from the functionality defined in the User Requirements Document. The User Requirements Document, however, is structured along the different kinds of users envisaged. In order to establish comfortable tracing, the $m : n$ relation between user requirements and software requirements will be accurately recorded in a double-linked way.
Chapter 10

General requirements

Development environment, standards and components, documentation and revision control are described in appendix D.

The call hierarchy between the components underlies a severe restriction due to UR 2.0.9 and UR 2.0.8: Because a browser can only deliver a HTTP-request, the worksheet must start browsers, too (and not only the dialog, as it would be desirable w.r.t. the system architecture).

SR 10.0.1 Browsers are called by a 'master-browser' and by the worksheet.

The connection between Java and SML has to be 'hand-made'. During the next few years there will be several changes at the interfaces between the ISAC-components belonging to these two language environments.

SR 10.0.2 The SML-kernel is started by a java thread which controls the time-out eventuall resulting from non-terminating loops in the knowledge interpreter.

SR 10.0.3 The SML standard-output channel is read by parsers, one for the SML-structures, and one for the mathematics formulas embedded in the SML-structures.

System requirements for the users: Users (with exception of the math authors, which work directly on the SML-kernel) are expected to work on standard-browsers (UR D.0.9) and some additional software components.

SR 10.0.4 On the client-side a Java virtual machine version .... must reside on the local computer of the learners and authors.

SR 10.0.5 On the server there is a Linux or Unix operating system, Linux version ...., Unix ...
Chapter 11

The worksheet

A worksheet is the protocol of a more or less interactive calculation of an example, and it offers access to all services necessary to calculate the result of the example.

A calculation mirrors the structure of the prooftree.

**SR 11.0.6 The structure of a calculation is given by the ME.** Consequently any editing in a calculation affects the parts depending on the edited formula or tactic. Edit the worksheet w.r.t. UR 4.0.33 for publication etc. is done outside ISAC after having exported the calculation.

**SR 11.0.7 Export a calculation to a standard format** preferably XML plus MathML.
Chapter 12

The browser generators

\isolde{\textls{-40} ZSAC\textls{-40}’s mathematics knowledge (on (1) theories (2) problems (3) methods) is held in SML datastructures. Here we describe all requirements concerning the generation of the browsers written in Java, whereas the generation of the math knowledge itself is described in the ‘interfaces for authors of math knowledge’.}

The browser generators provide the contents for the browsers. This means that a browser queries its respective browser generator about subnodes etc. whereas the further description is held outside the SML-engine. To maintain this connection, unique identifiers are needed.

\textbf{SR 12.0.8} \textit{browser-generators use locally unique identifiers} (locally unique within SML) for their informations (for each node of their structure).

\textbf{Data are structured hierarchically,} i.e. the problems, methods and examples. (The theories for a directed graph without cycles; thus it also can be presented by a hierarchy.)

\textbf{SR 12.0.9} \textit{The hierarchy displayed in a frame} in order to have it visible all the time.

\textbf{SR 12.0.10} \textit{The hierarchy has arbitrary levels.} For the headlines for each level see SR \ldots SR 12.3.1.

\textbf{SR 12.0.11} \textit{The hierarchy shows the position} of the related element displayed in the browser-window; for this purpose an additional button is required, because the ‘back’ and ‘forward’-button on the browser may disconnect the relation.

12.1 The theory browser generator

The generation of the theory browser is already implemented by Isabelle. Within phase 1 of development, ZSAC will take this component without any change.

\textbf{SR 12.1.1} \textit{The theory browser generator is given by Isabelle.}
12.2 The problem and method browser generators

12.3 The example browser generator

This authoring component comprises all tools necessary to generate the content displayed by the example browser.

**SR 12.3.1 The headlines of the example-hierarchy:** The hierarchy comprises the labels of the chapters, sections, subsections etc. plus the respective head line, and the blocks of examples with the respective labels – all defined by the user (see UR 8).

**SR 12.3.2 Integrated editors for text, formulas and figures.** This integration should be as smooth as possible; it includes an MathML-based formula-editor for the formalization.
Chapter 13

The knowledge browsers

All browsers (Example-Browser and Knowledge-Browsers) present their output in a similar way. Textual descriptions have to be combined with images, formulas, formalisations, problems, ... and links to further informations (UR5.0.36). The structure of this informations has to be taken from the respective browser-generators. All kinds of information might be interlinked among each other, but not all parts of the information might be already present in the system when a new item is inserted. (e.g. a example might use the keyword is_rooteq_in in its where clause before the keyword is described in the system. If the example is viewed after a description became available, a link to the explanation should be provided automatically.) To support this feature, the browsers need a fast way to look up a special description.

SR 13.0.3 fast look up for description (includes resolve of the identifier and query for a description)

Groups of users are usually 1-to-1 related to courses; members of courses get specific examples and specific advice by explanations.

SR 13.0.4 Each user is assigned exactly one group during a session. The user, however, may start another session as a member of another group.

SR 13.0.5 There is a default group for visitors.

Additional data and visibility properties: A considerable part of the data are additional to the data generated from the SML structures. The visibility concerns enabled or disabled links from problem and methods, and the access to examples from the hierarchy frame. There are respective time constraints on the visibility.

SR 13.0.6 SR 13.0.7 Additional data and visibility are course specific.
SR 13.0.8 *Time constraints are given by intervals:* there are specific values for the limits of the interval indicanting infinity (constraint is given from the very beginning, or lasts forever).

13.1 The theory browser

The theory browser is already implemented by Isabelle. Within phase 1 of development, ISAC will take this component without any change.

SR 13.1.1 *the theory browser is given by Isabelle.*

13.2 The problem and method browsers

**Additional data** are explanations and typical examples which may be provided for each problem and each method. Both of these kinds of additional data are specific for courses and may underly time constraints (UR 7.0.45, UR 7.0.46, UR 7.0.47).

The primary contents are the respective problem and the respective method; all other data are reachable by links. This holds for special mathematical data (on rulesets etc.) from SML, too.

SR 13.2.1 *A problem-page consists of* the

- name of the problem
- the fields 'given', 'where', 'find' and 'relate'
- a link to the special math data
- a list of subordinated methods (only displayed in the hierarchy-frame)
- an arbitrary list of links to additional data

The next lower and next higher level in the hierarchy can be found in the hierarchy-frame.

SR 13.2.2 *A method-page consists of*

- name of the method
- the script
- a link to the special math data
- a list of subordinated methods (only displayed in the hierarchy-frame)
- an arbitrary list of links to additional data

SR 13.2.3 *The links to additional data* have the following attributes:

- text on the link
- the reference (to an explanation or to an example starting a worksheet)
- groups
  - ID of first group
  - locked: list of intervals
  - used by userID —> userModel
  - ID of second group . . .
13.3 The example browser

In contrary to the problem and method browsers, the presentation of the contents of a browser-window is not generated automatically. UR 8.0.59 requests for a layout 'handmade' by the course designer; there are, however, a lot of attributes invisible for the learner to be added by the course designer, too.

Attributes of examples and collections are numerous, and thus defaults help to save space. There is only one collection, partitioned hierarchically into subcollections.

SR 13.3.1 Default data are filled in bottom up: A default is filled by the first non-default parent.

SR 13.3.2 A subcollection has the attributes

- headline of the collection (displayed also in the hierarchy-frame)
- description of the collection
- list of subcollections
  OR
- layout of examples belonging to this subcollection
- author
- copyright owner
- groups of users, for each group:
  - ID of the group
- schemas
  - error
  - fill-in
- dialog
  - blackbox
  - detail
  - activity
  - stepwidth
- invisible: list of intervals OR duration (? TODO !)
- locked: list of intervals OR duration (? TODO !)
- evaluation
  - TODO: difficulty, length, ...
  - finished-by:
  - elements: list of mandatory examples (or groups) to be done
  - number: number of (arbitrary) examples (or groups) to be done
  - points: calculated from TODO: difficulty, length, ...

Pay attention to the entry 'list of subcollections OR layout of examples belonging to this subcollection': This means that the subcollection exactly one
hierarchy-level above the bottom of individual examples has a specific content, the graphical layout of the examples. This is in order to meet UR 8.0.59.

**SR 13.3.3 An example has the attributes**

- label
- reference ??? to the description of the example (in the layout ?)
- list of formalizations and specifications
- author
- copyright owner
- time stamp
- groups of users, for each group:
  - ID of the group
  - schemas
    - error
    - fill-in
  - dialog
    - blackbox
    - detail
    - activity
    - stepwidt
    - invisible: list of intervals OR duration (? TODO !)
    - locked: list of intervals OR duration (? TODO !)
    - evaluation
      - TODO: difficulty, length, . . .

???done by userID —> usermodel???

**SR 13.3.4 Hitting the label of the example starts the calculation.**

Visibility of the examples is defined in two levels: (1) 'locked' displays the text of the example, but doesn’t allow to calculate it; and (2) 'invisible' doesn’t display the example at all.

**SR 13.3.5 Only visible examples are checked for being locked.**
Chapter 14

The dialog guide

As already mentioned, the dialog guide will be fleshed out in a later phase of development involving didactics and learning theory. Now we try to establish a framework open for later completion.

The dialogstate is read and updated during one session. A dialog resumes the dialogstate from the previous session done as a member of the same student-group.

SR 14.0.6 The last dialog is stored for each group the student is a member of. When storing and replacing the previous dialogstate, this dialogstate is transferred to the history of the usermodel (eventually after compression).

SR 14.0.7 The dialogstate has the attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>begin</td>
<td>timestamp of begin of session</td>
</tr>
<tr>
<td>provided-end</td>
<td>e.g. for examinations</td>
</tr>
<tr>
<td>actual-end</td>
<td>empty, or timestamp of end of session</td>
</tr>
<tr>
<td>group</td>
<td>the user has started the session with</td>
</tr>
<tr>
<td>interactions, for each:</td>
<td></td>
</tr>
<tr>
<td>timestamp</td>
<td>empty if ISAc entered via KB</td>
</tr>
<tr>
<td>label of example</td>
<td>tactic, formula, command or label in KB</td>
</tr>
<tr>
<td>input</td>
<td>??? of which part of system ???</td>
</tr>
<tr>
<td>response</td>
<td>of dialog</td>
</tr>
<tr>
<td>pattern</td>
<td></td>
</tr>
<tr>
<td>activity</td>
<td></td>
</tr>
<tr>
<td>stepwidt</td>
<td></td>
</tr>
<tr>
<td>...TODO ...</td>
<td></td>
</tr>
</tbody>
</table>

The use of these fields is shown by use-case UC TODO.

The usermodel consists of two parts: the settings of the personal preferences and the history of (condensed) dialogstates. The history is constructed
from the dialogstates: before a dialogstate is being replaced at the start of a new session, its data are restructured and appended to the history.

**SR 14.0.8 The usermodel has the attributes**

- **settings**
  - patterns, for each: of dialog
  - activity
  - stepwidth

- **history**, for each session:
  - `begin_end` 2 timestamps
  - `group`
  - `kb_touches`, for each:
    - `label of KB item`
    - `timestamps`
  - `examples`, for each:
    - `label of example`
    - `begin_end` 2 timestamps
    - `finished` yes/no
    - `performance` from example.evaluation.TODO and from dialog.interactions

The use of these fields is shown by use-case UC TODO.
Chapter 15

The mathematics engine

Here only these requirements are recorded which have been newly raised when specifying the interfaces to the ME.

SR 15.0.9 *The structure of a calculation is given by ??? for each formula*
Bibliography