



The iotaMed™ Design

Martin Wehlou MD, CISSP, CSDP
MITM AB, Sweden, 2010
martin@mitm.se

V3 2010-10-19

Table of Contents

1	Introduction.....	6
2	License.....	7
2.1	Documentation.....	7
3	What the user sees.....	8
3.1	Initial overview of patient.....	8
3.1.1	Issue list (A).....	8
3.1.2	Common notes area (B and C).....	8
History area.....		9
Observation union.....		9
4	Issue Template.....	10
4.1	The issue template document structure DSD2.....	10
4.2	The issue template document structure DTD.....	10
4.3	The issue template document structure as pseudo BNF.....	11
4.3.1	The issueTemplate element.....	11
Template inheritance.....		12
4.3.2	Action element.....	12
Action type 'referral'.....		12
Action type 'form'.....		12
Action type 'prescription'.....		12
Action type 'xray'.....		13
Action type 'mail'.....		13
4.3.3	Observation element.....	13
Observation types.....		13
4.4	Observation value linking.....	14
4.4.1	Inheritance and observation linking.....	14
4.5	Observation export.....	15
5	Issue worksheets.....	16
5.1	Issue worksheet fields.....	16
Normal empty field.....		16
During recording.....		16
After recording.....		17
During playback.....		17
After transcription.....		17
After signing.....		17
5.2	Issue worksheet value pool and ID.....	17
5.2.1	Exporting issue worksheets.....	18
5.2.2	Value histories.....	18
6	Template editor.....	19
7	Issue Template Distribution System (ITDS).....	20
7.1	Distributed index.....	20
8	iotaMed Instant Messaging (IMIM).....	21
9	iotaMed Access Rights Protocol (IMARP).....	22
10	iotaPad: iotaMed for Apple iPad.....	23
11	iotaMed Business Layer (iotaMedBL).....	24
12	iotaMed Data Access Layer (iotaMedDAL).....	25
13	Database engine.....	26
13.1	Neo4j.....	26
14	Interconnections.....	27
14.1	iotaPad to iotaMedBL.....	27
14.2	iotaMedBL to iotaMedDAL.....	27

14.3	iotaMedDAL to database engine.....	27
14.4	iotaMedBL to legacy EHR system.....	27
14.5	iotaMedBL to quality registers.....	27
15	Ideas for studies.....	28
15.1	Early studies.....	28
15.2	Late studies.....	28
15.2.1	Comparison of clinical effect.....	28

Version history

<i>Version</i>	<i>Date</i>	<i>What changed?</i>
1	20101016	Creation
2	20101017	Added description of voice recording and issue worksheet fields.
3	20101019	Updates in DTD. Updates in BNF description. Added value pools, value histories, and a discussion on exporting issue worksheets.

Terms and definitions

<i>Term</i>	<i>Description</i>
<i>action</i>	An <i>action</i> is an element in the issue template and issue worksheet that defines a possible executable function such as creating a referral, lab order, or other form. The <i>action</i> contains information needed to prefill as many fields as possible in that form.
<i>BNF</i>	Backus-Naur Form, a way to succinctly describe a syntax.
<i>common notes</i>	The first default window shown for a patient on <i>iotaPad</i> which shows quite a few similarities with the notes in a classic <i>EHR</i> .
<i>DSD2</i>	The definition we use to describe XML document structures. It's a project you can find out about at: http://www.brics.dk/DSD
<i>EHR</i>	Electronic Healthcare Record (same as EMR).
<i>EMR</i>	Electronic Medical Record (same as EHR).
<i>encounter</i>	Meeting between doctor and patient. Can be inpatient or outpatient. Includes history taking, examination, discussion, conclusion, etc.
<i>IMARP</i>	<i>iotaMed</i> access rights protocol
<i>IMIM</i>	<i>iotaMed</i> instant messaging
<i>iotaMed™</i>	The whole of the design based on interactive guidelines, but also including the interfaces, file formats, and other characteristics developed for the purpose.
<i>iotaMedBL</i>	<i>iotaMed Business Layer</i> : the part of <i>iotaMed</i> that runs under OSX and takes care of most processing that isn't related to a user interface or the direct storage of data.
<i>iotaMedDAL</i>	<i>iotaMed Data Access Layer</i> : the part of <i>iotaMed</i> that talks to the <i>iotaMedBL</i> on the one hand and to the actual database engine on the other. <i>iotaMedDAL</i> runs under OSX.
<i>iotaPad</i>	The <i>iotaMed</i> client implemented on the Apple iPad.
<i>issue</i>	The medical problem, symptom, affliction, or disease. May also include therapies, such as antico treatment or a suite of radiotherapy treatments.
<i>issue template</i>	The model guideline containing the text and instructions for where <i>observations</i> and <i>actions</i> should take place.
<i>ITDS</i>	The <i>issue template distribution system</i> . Takes care of adding or updating <i>issue templates</i> in <i>iotaMed</i> systems, after creation or modification of these <i>issue templates</i> by the medical authorities. Version control is part of the ITDS.
<i>issue worksheet</i>	An instance of an <i>issue template</i> being used in a patient record. Contains actual observation data. May also have both added and removed content as compared to the <i>issue template</i> , but always remains backlinked to the original <i>issue template</i> .

<i>Term</i>	<i>Description</i>
<i>item</i>	An <i>item</i> is a part of an <i>issue template</i> or <i>issue worksheet</i> that contains some text, optionally one or more <i>actions</i> , and optionally one or more <i>observations</i> .
<i>link file</i>	A file containing <i>linking statements</i> , which connect the <i>observation</i> identifiers in an <i>issue worksheet</i> with data representations in the environment, including lab systems and legacy <i>EHR</i> .
<i>Neo4j</i>	An open source graph database which we use as persistence layer in <i>iotaMed</i> .
<i>observation</i>	In an <i>issue template</i> and <i>issue worksheet</i> , the <i>observation</i> is a placeholder for actual data, such as blood pressure or lab values. Individual values are used in <i>issue worksheets</i> and are referred to as <i>observation values</i> .
<i>observation union</i>	A list of all the different <i>observations</i> found in all <i>issue worksheets</i> and presented in the <i>common notes</i> part of the <i>iotaPad</i> window.
<i>template editor</i>	An editor that can be used by medical endusers, such as head of departments, to author <i>issue templates</i> .
<i>template format</i>	<i>Issue templates</i> are written in XML with a set of defined tags and attributes. The set of these tags and attributes, plus the rules that define the relationship between tags and attributes, is the <i>template format</i> . The <i>template format</i> is defined in a DSD2 document. See: http://www.brics.dk/DSD/

Locations and references

<i>URL (http://)</i>	<i>Description</i>
iota.pro	Wiki considered home page for the project
vard-it.se	Swedish/English blog about IT in healthcare and <i>iotaMed</i>
vard-it.se/vi	Swedish language phpBB forum for same
brics.dk/DSD	Source for definition and examples of the DSD2 schema
neo4j.org	Home page for the Neo4j graph database project
github.com/mwehlou/iotaPad	Source repository for the <i>iotaPad</i> prototype

1 Introduction

iotaMed[™] is a new electronic healthcare record (*EHR*) overlay intended to take a new grip on the problems related to record keeping and management of patients. It does not replace any preexisting *EHR* system, but enriches the information already present in that system, while at the same time adding in the missing entities. In some cases, *iotaMed* will have its own database where these additional entities are kept. If an *EHR* vendor implements *iotaMed* functionality into his existing system, the additional entities will probably be added to the existing database. *iotaMed* does not implement the traditional chronological annotation record, orders, referrals, lab, or radiology, but relies on existing systems for that functionality. It will, however, be an all inclusive frontend for these systems for the physician or nurse using *iotaMed*.

iotaMed takes ideas found in many diverse areas, including specialised medicine, and generalizes them in the medical record. It fundamentally changes the way records are kept, moving from a logbook format of notes to a guideline based interactive form, which allows the physician to apply evidence based medicine in a flexible format allowing individual adaptations and deviations. The control of the process remains firmly in the hands of the physician, but the information he needs is right in front of him at all times. One can say that *iotaMed* implements a "heads-up display for doctors".

The basic idea with *iotaMed* is that the doctor is presented with one or more "*issue worksheets*" for the patient. Each "*issue worksheet*" relates to one "*issue*" the patient has. An "*issue*" can be a disease, a symptom, or a major therapy regimen, such as a sequence of radiotherapy sessions.

The difference between an "*issue worksheet*" and a regular clinical guideline is that the *issue worksheet* is interactive. If the *issue worksheet* calls for the measurement of blood pressure, for instance, you can enter the measured value right there in the worksheet. If the worksheet says that you should check for a certain lab parameter, the most recent value of that parameter for that patient can be seen right there in the *issue worksheet* so you don't have to go look for it. Or if it hasn't been measured yet, you touch that part of the *issue worksheet* and a lab order form pops up with all the relevant information, including the desired lab test, already filled in, ready to send.

The *iotaMed* design doesn't end there. It includes methods and formats for interchange between systems, and a publish-subscribe protocol which would allow systems to keep a tab on how other systems use and access data, or even control how those systems can use data. One system could in fact represent a patient and in this way give the patient a "control panel" to set access masks and check on how care givers have used his records.

iotaMed goes on into the murky waters of databases and is designed to work with both singleton databases and distributed databases for the entities that belongs to it.

No doubt, the *iotaMed* design will invade surrounding technical areas like a fungus, and with a bit of luck, will prove to be the penicillin for medical information technology we so sorely need.

2 License

(This section is just a draft, it may change considerably, disappear entirely, or at a moment's notice metamorphose into Batman and fly off into outer space. Stand clear.)

The *iotaMed*TM product and project is open source based. The use of both the code and the documentation is encouraged and no license fees apply for that. You are not allowed to sell the original *iotaMed* code unchanged for a profit and there are conditions to take into account if you include our code in commercial projects. There are also limitations and conditions that apply if you modify the code, but as you will see when reading the licenses¹, it is nothing that will stop you from doing almost as you please when it comes to introducing useful products working in conjunction with our code.

The use of the *iotaMed*TM trade mark, however, is restricted. We intend to license the use of this trademark to products that we feel live up to the intentions of the *iotaMed* project and which show the requisite quality of design and implementation. We do not necessarily charge monetarily for the licensing of this use by others, unless considerable work is needed for us in connection with making sure that it is appropriate to license the mark.

Let's put it this way: if you've done a good job, created a nice, beautiful, and solid program or standard that will do the *iotaMed* project good and make us proud to have you, and even contributed something back to the *iotaMed* project, then don't worry, you won't have to pay for a license, you'll get that nonexclusive right of use for free and in writing. We may even throw in a bottle of champagne if we've got the money to spare for that. Or a can of mineral water, maybe.

On the other hand, if you're just out to get the trademark on an otherwise crappy product, you won't get it, regardless of what you're prepared to pay for that. If we're selling our souls to the devil, we prefer to choose which devil ourselves. But if you're prepared to improve your product up to required standards², well, we can help for a price, if you need help.

2.1 Documentation

The license for the documentation, such as this document, is this:

- You can redistribute it for free if complete and unchanged, including front page, table of contents, and all the text, including this section.
- You can refer to, and reuse parts of the documentation if you include a mention of the source. Don't forget to include the version number and date of this document or any other document you refer to, and where the user can get a more recent version. This (or *these*, as may be) documents will certainly change a lot and old versions can quickly become counter-productive and confusing.

¹ Which you won't find just yet. I need to select and define the type of license we're going to use.

² *What "required standards" you may well ask?* Well, those we will impose as we go along, just to make sure the product is what we want it to be. That's the beauty with owning a trade mark; you get to decide what it means.

3 What the user sees

The rest of this document goes into the details of how *iotaMed* works internally, but we need to know what the user will see and why this is what he should see.

3.1 Initial overview of patient

Once a patient is chosen and the medical record is opened for the user, this is the layout he will see:

Hypertension Diabetes II NDR Form A	20101006 History: Dr Wehlou Feels just fine. Some aches in the legs, but nothing he can't stand. B
	BP: 120/80 Pulm: VES bilat. C Abdomen: Normal HbA1c: 7.3
	20101004 History: Dr Wehlou Been nauseous a few days, but hasn't lost weight. B
	BP: 135/85 Pulm: VES bilat. C Abdomen: Slight tenderness, no defence HbA1c: 6.8

The areas are marked as follows:

A: the list of *issues* this patient has. Each entry corresponds to one *issue worksheet*.

B: history

C: observation union

3.1.1 Issue list (A)

The left part of the screen shows the *issue list*. If the *iPad* is held vertically, this part is shown as a popover if the "issues" button is touched.

The actual *issue worksheets* are shown in normal size font and any attached quality register forms are shown indented and in smaller font, as is the "NDR" (National Diabetes Register) form in this example.

3.1.2 Common notes area (B and C)

The entire right pane in the illustration above is the *common notes* area. It intentionally closely resembles the notes in a classic *EHR*. It is a scrolling area where each encounter shows a *history field* and a set of *observations* where we call that set the *observation union*.

Both the *history* area and the *common notes* area are fully interactive at any time, as long as the notes for an *encounter* haven't been signed and locked.

History area

Each encounter has its individual history data element, which is not specific for any one *issue worksheet*. When the patient tells the doctor how he's been feeling before coming back to visit, it is usually at least partially relevant to all the *issues* the patient has and there would be little point in having several different history fields.

Observation union

As long as no *issue worksheet* has been selected in the left pane, the scrolling right side area shows a union of all *observations* of all the individual *issue worksheets* in a list below the history field. The result closely resembles a classic encounter entry in regular *EHR* systems and is also the form that is exported to the legacy *EHR* system connected to *iotaMed*.

In the absence of *issue worksheets* for a patient, a predetermined set of *observations* may be present in the list below the history field, such that it is easier for the physician to follow a standard clinical examination procedure before deciding on which *issue worksheets* to create. Since the *observations* in this list are the same type as those in the *issue worksheets*, no unnecessary duplication occurs.

4 Issue Template

The *issue template* contains a clinical guideline for a particular *issue*. The contents of an *issue template* are determined by a medical authority, such as a head of a department or a regional healthcare authority, typically a doctor. He or she uses a *template editor* to create or modify an *issue template*. The distribution of that template to the *iotaMed* user systems are handled by the *issue template distribution system*, described elsewhere in this document.

4.1 The issue template document structure DSD2

(Note: I love it when there is a nifty and clear standard notation, but I'm beginning to doubt this is one. It's certainly great for automatic verification of a document, but what I really need is a clear notation for consumption by human eyes, and this only makes my eyes water. Good ol'fashioned DTDs are a lot clearer as to intention and content, and my TextMate verifies them just fine. So this section will probably go in the dumpster.)

The *issue template* has a standard document structure and is expressed formally in the DSD2 syntax. For more on the DSD2 syntax, see: <http://www.brics.dk/DSD/>

The issue template definition looks like this in DSD2:

```
<dsd xmlns="http://www.brics.dk/DSD/2.0"
      xmlns:m="http://www.brics.dk/DSD/2.0/meta"
      xmlns:im="http://www.mitm.se/iotaMed/issueTemplate"
      root="im:issueTemplate">

<m:doc> A DSD2 schema for iotaMed issue templates. </m:doc>

<if><element name="im:issueTemplate"/>
  <declare>
    <contents>
      <repeat><element name="im:section"/></repeat>
    </contents>
  </declare>
</if>

<if><element name="im:section"/>
  <declare>
    <attribute name="title">
      <string/>
      <normalize whitespace="trim"/>
    </attribute>
  </declare>
</if>
```

4.2 The issue template document structure DTD

Ah, oldfashioned and all, but at least a DTD is readable. So here we go.

```
<?xml version="1.0"?>
<!DOCTYPE issueTemplate [
<!ELEMENT issueTemplate (section)>
<!ATTLIST issueTemplate title CDATA #REQUIRED>
<!ATTLIST issueTemplate fromICD10 CDATA "">
<!ATTLIST issueTemplate toICD10 CDATA "">
<!ATTLIST issueTemplate lang CDATA "US-EN">
<!ATTLIST issueTemplate region CDATA "">
<!ATTLIST issueTemplate serial CDATA #REQUIRED>

<!ELEMENT section (#PCDATA|subtitle|item|info)*>
<!ATTLIST section title CDATA #REQUIRED>

<!ELEMENT subtitle (#PCDATA|subtitle|item|info)*>
<!ATTLIST subtitle title CDATA #REQUIRED>
```

```

<!ELEMENT item (#PCDATA|ul)*>

<!ELEMENT info EMPTY>

<!ELEMENT ul (li)*>
]>

```

Still incomplete, but stay tuned...

4.3 The issue template document structure as pseudo BNF

It may be easier to describe and understand the *issue template* structure if described in a pseudo BNF form. Attributes are in braces, elements without.

<i>element</i>	<i>consists of</i>	<i>tag, if any</i>
issueTemplate	:= {fromICD10, toICD10, title, lang, region, serial} parent*, section*	issueTemplate
parent	:= {serial, chksum, sequence}	parent
section	:= {title} flow?	section
subtitle	:= {title} flow?	subtitle
flow	:= (subtitle item info subcon)+	
item	:= (subcon action observation)*	item
subcon	:= (list paragraph)+	
list	:= listitem*	ul
listitem	:= subcon?	li
info	:= subcon?	info
action	:= {type, title, cause, addressee}	action
observation	:= {name, type} input*, display*	observation
paragraph	:= text	
input	:= {form, mask}	input
display	:= {from, count, period}	display
issueTemplate attributes		
{fromICD10}	string: ICD10 code	
{toICD10}	string: ICD10 code	
{title}	string, will be displayed	
{lang}	ISO language code: language used in template text	
{region}	string: a comma separated list of medical authorities responsible for the template	
{serial}	numeric: a 128-bit random number, expressed as a GUID in string form	
section attributes		
{title}	displayed title	
subtitle attributes		
{title}	displayed title	
action attributes		
{type}	'referral' 'laborder' 'sickleave' 'form' 'prescribe'	
{title}	string: the text to display on the popup button field	
{cause}	string: the text to fill the "reason for referral" field with	
{addressee}	string: destination address for referrals, newlines as ';'	

4.3.1 The issueTemplate element

The *issueTemplate* element defines the template, the pathology range it covers, its language, and its inheritance.

Template inheritance

When templates are modified, the parent lineage from which it is descended is preserved in the template. That way, the lineage can be traced both ways, which is useful for:

- going up the line: checking if changes have been done in a parent that we need to incorporate or warn the user about
- going down the line: find out how many changes have been done, how popular they are (so they can be worked into the main stream), and how many cases of a particular disease are out there and how they are managed

Example:

```
<issueTemplate fromICD10='I10' toICD10='I10' title='Hypertension' lang='US-EN'  
    serial='EC770D43-EFF9-4D0F-B280-C01784074A02'  
    chksum='uj523m;lf34m;ldfh45b'  
    sequence='2'>  
  <parent serial='468F76A6-CCA4-4A21-96A4-B5ADCA68D3F3' chksum='vnio..asj' sequence='1' />  
  <parent serial='E7E630C7-2CD1-44F2-AB77-64EAC9B82103' chksum='un7...87j' sequence='0' />  
  ...  
</issueTemplate>
```

In this code, the last *serial* number indicates the original template. Some changes were made and the second from bottom *issue template* was created. The current *issue template* is the third version and has *sequence* number = 2.

Each template also has a 160 bit checksum expressed in alphanumeric. One would be excused for assuming that the *chksum* alone would suffice, but it isn't unlikely that two *issue templates* in the same inheritance tree would be identical, but with a different *issue template* connecting them. For example, somebody adds a line and somebody else removes it again and there you are. Having only *chksum* identifying the *issue templates* would then royally screw up the inheritance tree.

A remark on the *sequence* attribute: it may seem redundant, but I'm sure having it will stop most bugs caused by confusion as to the actual order of the *parent* elements. This way, the order of the *parent* elements in the XML is undefined, since the *sequence* attribute rulez.

4.3.2 Action element

The *action* element can be a bit complicated, so let's describe that in more detail here. Depending on the *type* of the action element, other attributes are needed to specify the *action* more in detail.

Action type 'referral'

Most referrals will be done using predefined forms.

Action type 'form'

Forms that do not fit any other *action type* can be filled in using a generalized '*form*' type.

Action type 'prescription'

This *action type* activates the pharmacological prescription window to either initiate, modify, or extend an ordination. The attributes we'll use in this case are:

Action type 'xray'

This *action type* is used to order an xray examination.

Action type 'mail'

This *action type* is used to write a longer message that can, depending on the addressee, result in an email or other electronic message, or a paper document to be sent through regular mail.

4.3.3 Observation element

The observation element isn't trivial either, so it also needs its own treatment. The *observation* element contains two other elements, the *input* and the *display* element. The idea here is that a typical observation element allows the user to input new values, while showing old values next to it for reference.

The *observation* has several attributes and contains optional *input* and *retrieve* elements, at least one of which must be present in each *observation*. Example:

```
<observation name='bodyweight' type='numeric'><input mask='%f.1' />
  <retrieve from='last' count='2' period='any' /></observation>
```

In this example we provide for a numeric field for body weight, while also retrieving the last two registered values over any timeframe and showing those next to the input field. Note that the order of the elements does not matter; which field is presented first or last is decided by the presentation software.

Observation types

The *type* attribute for the *observation* element can have the following values:

<i>numeric</i>	The value is a floating point numeric value.
<i>string</i>	The value is a string
<i>select</i>	Select one of several choices
<i>multi</i>	Select zero or more of several choices

In the case of *select* or *multi*, the available choices are given by *choice* elements as in this example:

```
<observation name='pain' type='select'><choice title='None' id='1' />
  <choice title='Some' id='2' /><choice title='Severe' id='3' />
  <input /><retrieve from='max' /><retrieve from='min' /></observation>
```

In this example, we let the user select one of three options, while at the same time displaying the highest and lowest value ever seen together with the date when they were seen. The *id* value in the *choice* elements is for mapping through a link file³. Example:

```
'77AA...66FF'::'pain' = ::clinterm::'ArthritisPainLevel'
```

This requires the right hand side term to have exactly the same number of choices as the left hand side, matching them on *id* number in sequence. If you need to transform values, you have to pipe them through a process using the *pipe* token:

³ For more on *link files* see "observation linking".

```
'77AA...66FF'::'pain' = ::pipe('/usr/sbin/iotaconv')::clinterm::'ArthritisPainLevel'
```

The *pipe()* command takes care of supplying the full path of both source and destination term to the process in parenthesis. You can use a macro replacement here if you want to make your link files more flexible.

Using the *pipe()* command, it's fully possible to map *m* number of choices to *n* number of choices and back⁴. It is also possible to map numeric or string data to choices and vice versa.

4.4 Observation value linking

The observation values presented to the user and entered by the user into the system needs to be linked to values available in the local environment. For example, a lab value in *iotaMed* must link to the right value of the right type in the lab system available at that location.

A particular *observation* name is fully qualified by a structure holding the *name* of the observation in the *issue template* together with the *serial* of the *issue template*. This has a number of advantages, as follows.

4.4.1 Inheritance and observation linking

If an *observation* has been linked to a local lab value as follows:

```
'468F76A6-CCA4-4A21-96A4-B5ADCA68D3F3'::'s-hb' = 'safir3'::'hemato'::'coulter1'::'hbs'
```

...where ":" is a path component separator and everything after the equals sign is dependent on how to locate terms in the local system term catalog. In this example, I use the same path separator, but that is not necessary.

The left part consists of two components:

1. the *serial* of the *issue template*
2. the *name* of the *observation* in the given *issue template*

These link statements are kept in one or more external files that can be reached by whatever engine is used to retrieve values for *iotaMed*.

A very interesting property of the above method is that if an *issue template* is modified and thus gets a new *serial*, it is trivial to apply the most recent *observation links* and to use inherited links for those *observations* where the links have not changed.

Within a system, the right part is the identifier that must be used to link *observation* values between different *issue worksheets*. Example:

Issue template 1 looks like this:

```
<issueTemplate title='Hypertension' ... serial='E7E6...4A02'>
...
  <observation issueTemplate='EC770D43-EFF9-4D0F-B280-C01784074A02' name='Hb-S'.../>
```

Issue template 2 looks like this:

```
<issueTemplate title='Colitis ulcerosa' ... serial='67B2...DD01'>
...
  <observation issueTemplate='468F76A6-CCA4-4A21-96A4-B5ADCA68D3F3' name='S-HB'.../>
```

Note that in both these cases, the "issueTemplate" value in the attribute does not match the actual

⁴ With some loss of information just like any m-to-n mapping. Just try Google translate forward and back...

issueTemplate it is in, and that is because these values have been inherited from an earlier *issueTemplate* and not updated since. The "issueTemplate" value points to the *issueTemplate* where the observation definition was last added or modified.

The link file or files contain the following statements:

```
'E7E6...4A02'::'Hb-S'@'%'d02.1 mmol/l' = 'safir3'::'hemato'::'coulter1'::'hbs'  
...  
'67B2...DD01'::'S-HB'@'%'d2.1 mmol/l' = 'safir3'::'hemato'::'coulter1'::'hbs'
```

The above link statements allow *iotaMed* to conclude that 'Hb-S' in one *issue template* or *issue worksheet* refers to the same thing as 'S-HB' in the other, in this particular installation. The mask part (following the "@@") will be used in the presentation of the value, which implies that the dimension of the value is installation dependent, not dependent on the *issue template*. The mask is optional, and if absent, the mask must be supplied by the source system. If present, it overrides the mask supplied by the source.

If you wish, you could have the right sides of these statements refer to a standard term from a standard term catalogue, and then in a subsequent step refer from the standard term to the actual local term or path for the lab system, but that is an entirely optional step.

For clinical values that are not sourced from a lab system or others, the syntax looks like this:

```
'77A0...FEED'::'BP'@'%'mmHg' = ::'clinterm'::'BP'  
'12AB...0DEA'::'RR'@'%'mmHg' = ::'clinterm'::'BP'
```

"*clinterm*" in the example can be anything you choose. It would not be too far fetched to reserve some terms for standards use, for instance 'SNOMED-CT'. The number of path members is in principle unbounded, so any kind of terms should work here. *iotaMed* only looks for matches and does not really care what is contained there. Since the first patch component is absent, *iotaMed* does not start to search for externally connected systems for this value and uses it solely for linking with observations in other *issue worksheets*.

4.5 Observation export

You can send *observation* values from one system to another using an *observation transfer file*. Such a file has the following structure:

- *Issue worksheet/template* inheritance
- *Observation* values

This, in code, could look like this:

```
<obsFile>  
  <observation issueTemplate='E7E630C7-2CD1-44F2-AB77-64EAC9B82103'  
    name='Hb-S' datetime='20101005-2305'>13.6</observation>  
  <observation issueTemplate='468F76A6-CCA4-4A21-96A4-B5ADCA68D3F3'  
    name='Hct' datetime='20101005-2305'>47</observation>  
  ...  
</obsFile>
```

One particular remark is worth giving here: the actual *issue template* from which these *observation values* are taken does not need to be referred to, since each *observation value* actually contains a reference to an *issue template*.

5 Issue worksheets

Issue worksheets are nothing but instantiated *issue templates*. They do instantiate in two varieties, however:

- *issue worksheet* proper
- *quality register form* or *read-only issue worksheet*

There is nothing essentially different technically between these two. The difference is that the *quality register form* only contains exactly those *observations* that are needed for a quality register. Most, if not all, of the *observations* in such a form is automatically filled in since the same observation will be found in the parent *issue worksheet*, such as "Diabetes II" in the example given earlier in this document. Additionally, a *quality register form* cannot be modified or written to by the user since all the information therein should be derived from information either in a regular *issue worksheet* or from values from the underlying legacy EHR or lab system.

Keeping the quality register form separate this way allows a number of nifty features:

- It is easy for the doctor to see exactly which data is sent to the register.
- The doctor can decide when to send the data, but it can also be done automatically at predetermined intervals.
- Through the mechanism of the *link files* and the *pipe()* command, the data may be processed and modified into another scale or form for the quality register form.
- The quality register forms can be maintained externally without impacting the *issue worksheets*.
- It is very hard to send the wrong data, since both the quality register form and the clinical *issue worksheet* work from the same data.
- It can be used to communicate desired actions to nursing organisations.
- It can be used to provide feedback from nursing organisations to doctors on an ongoing basis.
- It can be used to transfer a subset of data from one system to another in regions, or nationwide.

5.1 Issue worksheet fields

The fields in the worksheet need to be easy to understand and command. Since most fields allow either typing or voice recording, the following user interface methods will be used.

Normal empty field



Tapping the gray ball starts a voice recording. Tapping anywhere in the field to the right of the gray ball activates the built in keyboard for manual typing.

During recording



During the recording, a red pause symbol is shown. In the text field, a slowly increasing row of dots indicate progress. In the navbar, a VU meter is shown. If the pause symbol is tapped, recording

pauses. If the field to the right is tapped or any other place on the screen, the recording is finalized and sent for transcription.

After recording



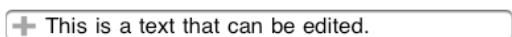
A text field with a row of dots is shown as long as the recording is not transcribed. The user can tap the play symbol to listen to the recording.

During playback



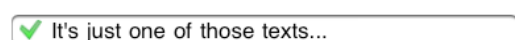
The user can tap the gray stop symbol to stop playing or anywhere outside the field to the same effect.

After transcription



Once the recording has been transcribed by a secretary, or immediately after the user has entered the text himself through a keyboard, the text is displayed with a plus sign as symbol. By tapping the symbol, the user can edit the text using the keyboard. He can return to voice recording by deleting all text first, then the gray "record" symbol will reappear.

After signing



The symbol changes to a green "verified" symbol.

5.2 Issue worksheet value pool and ID

An *issue worksheet* has a number of value placeholders under the *observation* tag. The value *input* tag allows for editing, while the *retrieve* tag only displays values. All of these values come from the *observation value pool*. Each placeholder, be it *input* or *retrieve*, refers to a particular *observation*, which is, if you think of it, only a definition of a value type. Well, it is even if you don't think of it, but still...

In an *issue worksheet* at any particular time, only one particular *observation value* is shown, even though there may be a whole series of them for a particular *observation*. Example: we have an *observation* tag referring to "Hb". If we have more than one "Hb" in the *observation value pool*, which one do we show? Or in the case of *input* tags, do we show any at all, or do we assume a new one should be entered?

For display fields as given by the *retrieve* tag, we can indicate with attributes if we want to see the most recent values, or some other kind of value, so that seems easy enough. But there's a potential problem even here, namely if we have an *item* that says something like "If two most recent values are above 7.1, the patient has diabetes", and we then confirm that the patient has diabetes. If later values are lower, we don't want the values in the *issue worksheet* to change at that particular point, since the text and further actions of the doctor would make no sense anymore, then.

A solution would be for any item that implies a decision to be made by the user depending on *observation* values, retrieved or input values, to allow a confirmatory action, a checkbox or multiple choice, which would then freeze the displayed values as well. This should be indicated by color or font somehow. It is clear that these items need to be formulated with care to make the intention

clear.

All this implies that items in an *issue worksheet* need to have a state that indicates if they are "open" or "confirmed". If "open", then they expect some kind of action from the user. If "confirmed" they do not, but that should not stop the user from intentionally changing their state later, if need be.

5.2.1 Exporting issue worksheets

When an *issue worksheet* is exported, we actually export a snapshot of the issue worksheet at a particular moment in time. To allow the user to select the right moment in time, if the current moment is not suitable, the selection should be coupled to a selection of the chronological record that parallels the *issue worksheet*. This modus operandi also easily lets us export all the *issue worksheets* with their state synchronized to the same moment in time.

5.2.2 Value histories

Every *observation* in an *issue worksheet* must then refer to a series of values; its *value history*. Each individual value must have a datetime and a unique identifier.

6 Template editor

To create or modify an *issue template*, a *template editor* is needed. This editor allows easy structuring of an *issue template* and the decoration of items with *actions* and *observations*. The editor also helps with the creation of the right kind of *serial* and *sequence* attributes for the *issueTemplate* element.

Theoretically, you can edit *issue templates* using just a plain text editor and working with raw XML, but since we want to put the editing and publishing of *issue templates* into the hands of medical authorities, raw XML could give them blisters, and we don't want that.

7 Issue Template Distribution System (ITDS)

When an *issue template* has been created or modified, it needs to be made available to *iotaMed* systems used by physicians in the field. Also, if a preexisting *issue template* is modified, a note or warning may be needed when an *issue worksheet* based on this *issue template* is opened by a physician. For example, if the first choice of medication for a particular *issue* is modified, a note to this effect should be made visible to any physician that needs to know about the change, at the moment and place where this information is important; usually when the patient is seen by a physician and his record is opened.

These functions are part of the responsibility of the ITDS.

7.1 Distributed index

The *ITDS* has, among other things, a distributed index allowing peer systems to locate *issue templates* and to find conversion files for *observations*. It also allows quality registers and others to track usage of *issue worksheets*, with certain boundaries necessary to protect the data from compromise.

The *template index* has as key value the *serial* of the *issue template* and as value the destination system URI. A large number of URIs may be present for each *serial* key.

Keys are entered into the *template index* on request of the *iotaMed* systems. The keys are removed whenever it is found that a key/value pair points to a system that denies having that *issue template*. This way, the participating *iotaMed* systems are not forced to maintain the index and lost messages do not matter much.

At least one, preferably several, *iotaMed* like systems should be tasked with holding on to at least one copy of each *issue template* ever made, in case a history path will need to be built at a later date. These systems will periodically update their storage by scanning the indices for new, unknown *issue templates* and retrieve them from the URI given in the value of the index entry. If the database knows about these repositories, it is relatively easy to have the database actively update the repositories instead, saving on a lot of database accesses and at the same time having a much earlier update of the repositories.

The index is also able to retrieve on the other attributes, namely: *fromICD10*, *toICD10*, *lang*, and *region*, making template searches according to theme, language and locality possible.

Each index instance talks to other indices to spread the contents peer-to-peer.

8 iotaMed Instant Messaging (IMIM)

iotaMed makes use of a short message format that can best be described as "instant messaging". For instance, every sound file that contains the dictation for a single field in an *issue worksheet* is sent immediately after dictation for transcription and return of the transcription to the originating *issue worksheet* for insertion. This entire round trip, including transcription, should have a length measured in minutes.

The same system can also be used within departments to locate and instruct a nurse, call up janitorial assistance, or pose a question to a supervisor. These queries and responses can be a mix of voice, text, and transcribed voice messages. We envisage messages of the size of a sentence or two for most use.

9 iotaMed Access Rights Protocol (IMARP)

There is a desire for patients to have insight and control of the access to medical records, and there is a need for an interchange protocol that can allow communications about this. Not only patients may need to control and monitor EHR access, so the protocol must allow for third parties to do that securely.

The protocol must have the following abilities:

- Set attributes for a particular patient
- Set attributes for that patient and particular healthcare providers (people or organizations)
- Set attributes for particular *issue templates* or *ICD10* ranges
- Set subscribers for these attributes
- Determine if the subscribers need to approve before access or just be notified

The "other half" of the protocol provides the notifications or requests for permission that were specified in the above "first half" of the protocol.

10 iotPad: iotMed for Apple iPad

iotPad is the project working name for the *iotMed* client running on the Apple iPad. It consists mainly of user interface elements, while leaving as much processing as possible to the business layer running on an OSX machine it connects to.

The screenshot shows the iotMed application interface on an iPad. The top status bar displays 'Carrier', signal strength, '2:15 PM', and '100%' battery. The patient information bar at the top shows '121212-1324 Erika Sjöfararen' and the diagnosis 'Hypertoni'. A left sidebar contains a list of conditions: 'Hypertoni' (highlighted in blue) and 'Fotledsfraktur'. The main content area is titled 'Utredning' and lists 'Tidigare/ nuvarande sjukdomar:' with a bulleted list of conditions: Stroke, TIA; Angina pectoris; Hjärtinfarkt; Hjärtsvikt; Diabetes mellitus eller annan endokrin sjukdom; Claudicatio intermittens; Astma; Gikt; Njursjukdom; and Lipidrubbing. It also includes 'Aktuella läkemedel och överkänslighet', 'Hypertonisjukdomen: Debutår. Tidigare blodtrycksvärden. Utredningsresultat.', 'Blodtryck under eventuell graviditet', 'Effekt och biverkningar av tidigare blodtrycksbehandling', and 'Förekomst av blodtryckshöjande läkemedel/ substanser (laktrits, NSAID, centralstimulantia, P-piller)'. Below this is the 'Status' section with a table of clinical data:

Parameter	Value	Date
Längd.	169.5 cm	2010/05/15
Vikt.	86.3 kg	2010/05/15
Midjemått.		
Tecken på hjärtsvikt.		
Hjärta.		
Lungor.		
Buk - Leverförstoring.	ua	2010/09/12
Auskultation över njurartär	inga biljud	2010/09/12
Pulspalpation.		
Ögonbottenundersökning.	grad 2	2008/03/13
Blodtrycksmätning	165/105	2010/07/28

EKG - Arytmier. Tecken till ischemi eller genomgången hjärtinfarkt.
Vänsterkammahypertrofi (Sokolow kriterium: SV1 (mm) + R V5/V6 (mm) >35 mm eller R V5/V6 >27 mm alt Cornells Index: RaVL (mm) + S V3 (mm) >35 mm).

11 iotaMed Business Layer (iotaMedBL)

12 iotaMed Data Access Layer (iotaMedDAL)

13 Database engine

The actual database engine used is not part of the *iotaMed* project as such. We use the Neo4j graph database, an open source project with its homepage at: <http://neo4j.org/>

The access method used between *iotaMedDAL* and the database engine is Neo4j's REST interface, described at: <http://components.neo4j.org/neo4j-rest/>

For temporary local storage, we use sqlLite, the built-in engine in iOS and OSX, and talk to it through Cocoa Core Data APIs.

13.1 Neo4j

Neo4j is a graph database running under a Java VM, and has a REST interface, which allows us to run the actual database on the same OSX machine the *iotaMedBL* runs on or on any other machine on the local net, irrespective of operating system.

14 Interconnections

Since most parts of the *iotaMed* system are distributed, several interconnection links need to be defined and maintained.

14.1 *iotaPad* to *iotaMedBL*

The *iotaPad* user interface module on the iPad needs to talk to the business layer under OSX, which we call *iotaMedBL*. The link must be secure, but preferably as stateless as possible. The idea is that it should be possible to swap out an iPad for another and keep as much of an ongoing session valid as possible.

This implies that the connection between the iPad and the OSX machine should be protected only by SSL.

14.2 *iotaMedBL* to *iotaMedDAL*

...

14.3 *iotaMedDAL* to database engine

...

14.4 *iotaMedBL* to legacy EHR system

...

14.5 *iotaMedBL* to quality registers

15 Ideas for studies

In this chapter, I jot down ideas for scientific and clinical studies related to the *iotaMed* system.

15.1 Early studies

...

15.2 Late studies

15.2.1 Comparison of clinical effect

When introducing *iotaMed*, the prime effect we expect is an improvement of clinical care. There is no way to do a double blind or even single blind study, but it would be informative to introduce *iotaMed* in one department if another comparable department can be located not too far away where *iotaMed* is not introduced. A number of parameters could then be compared, such as:

- conformance to established clinical guidelines
- number of patient visits needed or length of time to achieve a workable diagnosis
- the cost per patient for each defined diagnosis
- outcome

Alphabetical Index

chksum.....	12
clinterm.....	15
Inheritance.....	14
instant messaging.....	21
iotaMedBL.....	26
iotaMedDAL.....	26
iotaPad.....	23, 27
iPad.....	23
Issue worksheets.....	16
ITDS.....	20
link files.....	16
mask.....	15
Neo4j.....	26
observation linking.....	14
observation links.....	14
pipe.....	13
pipe().....	16
quality register form.....	16
read-only issue worksheet.....	16
REST.....	26
retrieve.....	13
sequence.....	12
serial.....	12, 14
SNOMED-CT.....	15
sqlite.....	26
Template editor.....	19
Template inheritance.....	12
transcription.....	21
voice messages.....	21