HMRL Tailoring Technologies ... Evolving Designs

Original HAMSTER design
(as used by Quit for Keeps)

Generation 1 Print Tailoring
(as used by M-Care, CISRC, and EDBI)

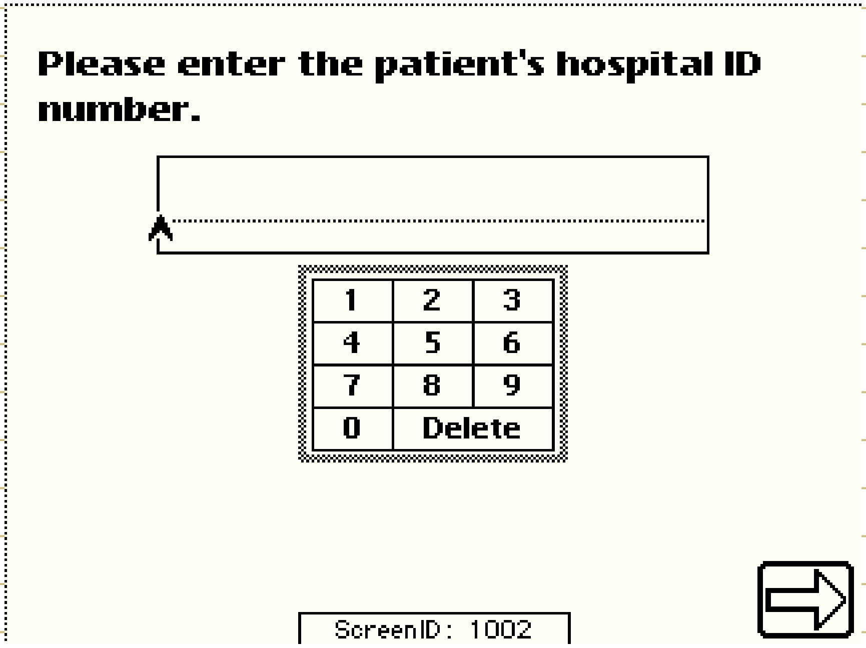
Java HAMSTER Design
(as used Henry Ford Health Systems MRF and Teen Smoking)

HAMSTER

Health Attitude Measurement System, Tiny Electronic Rendition

- Runs on PDAs (Apple Newton)
- Supports elaborate, scripted, and tailored surveys
- Communicates with desktop computer to export the data

Sample Newton User Interface



Please enter the patient's hospital ID number.

- Rated very easy to use and highly accepted by target audience

Embedded Scripts

- Questions have optional scripts.
- Chunks of code run before displaying the question or after it is answered.
- Used for live question tailoring, characterization, and branching.
- Scripting languages make developing these systems much, much easier

Uploading and Printing

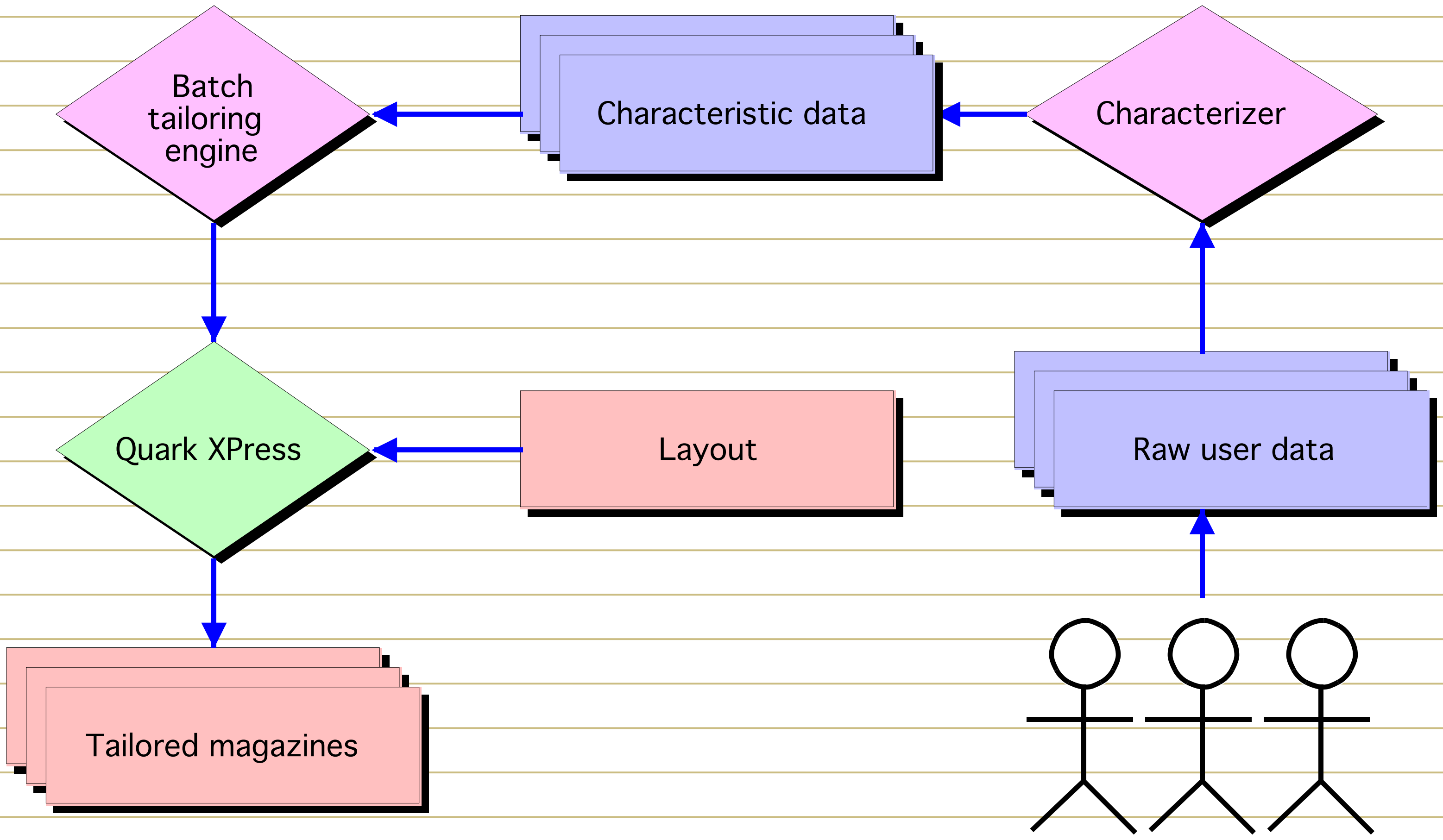
- Each day, all data is uploaded from all PDAs to a desktop computer.
- Updated data is e-mailed to the University of North Carolina.
- UNC then printed and mailed tailored magazines.
- Consolidated data is then re-downloaded to all PDAs.

Re-tailoring

- The next time, a subject could use any PDA and log in – it has their history.
- A new survey would be tailored on previous and current responses ("Last time, you said that you would quit smoking on May 1st. Did you do it?")
- Based on time or change in stage, a new magazine could be generated

Generation 1 Print Tailoring

Raw user data becomes tailored magazines!



Building the Characterizer

- Writers work with "characteristics"
- Taxonomy = rules to turn raw data into characteristics
- Examples of characteristics: body mass index, stage of change
- Keeps message logic simpler, more readable, easier to debug, and closer to the way our writers think

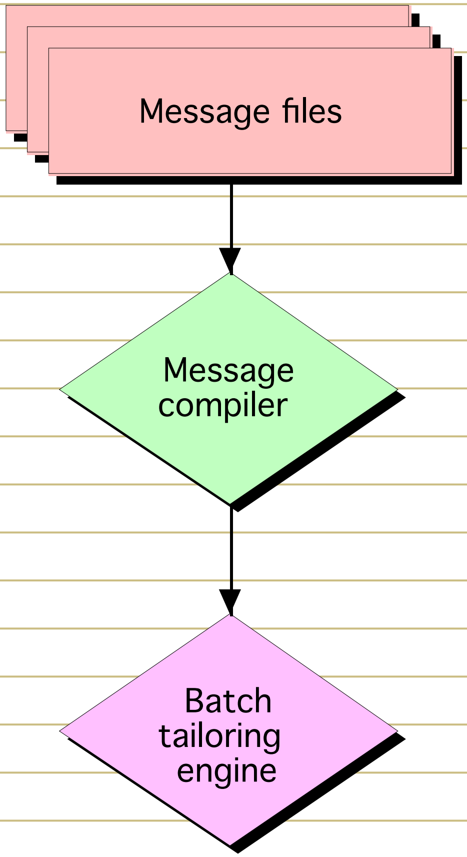
Building Tailoring Engine

- Message files edited in Microsoft Word™

- Message text and selection logic in “GroverTalk”

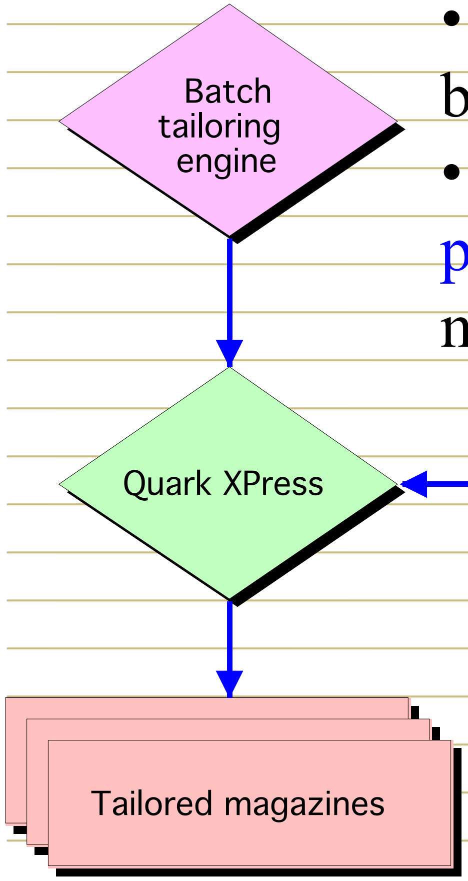
- Perl Message Compiler builds C++ code

- C++ code is built into a customized tailoring engine

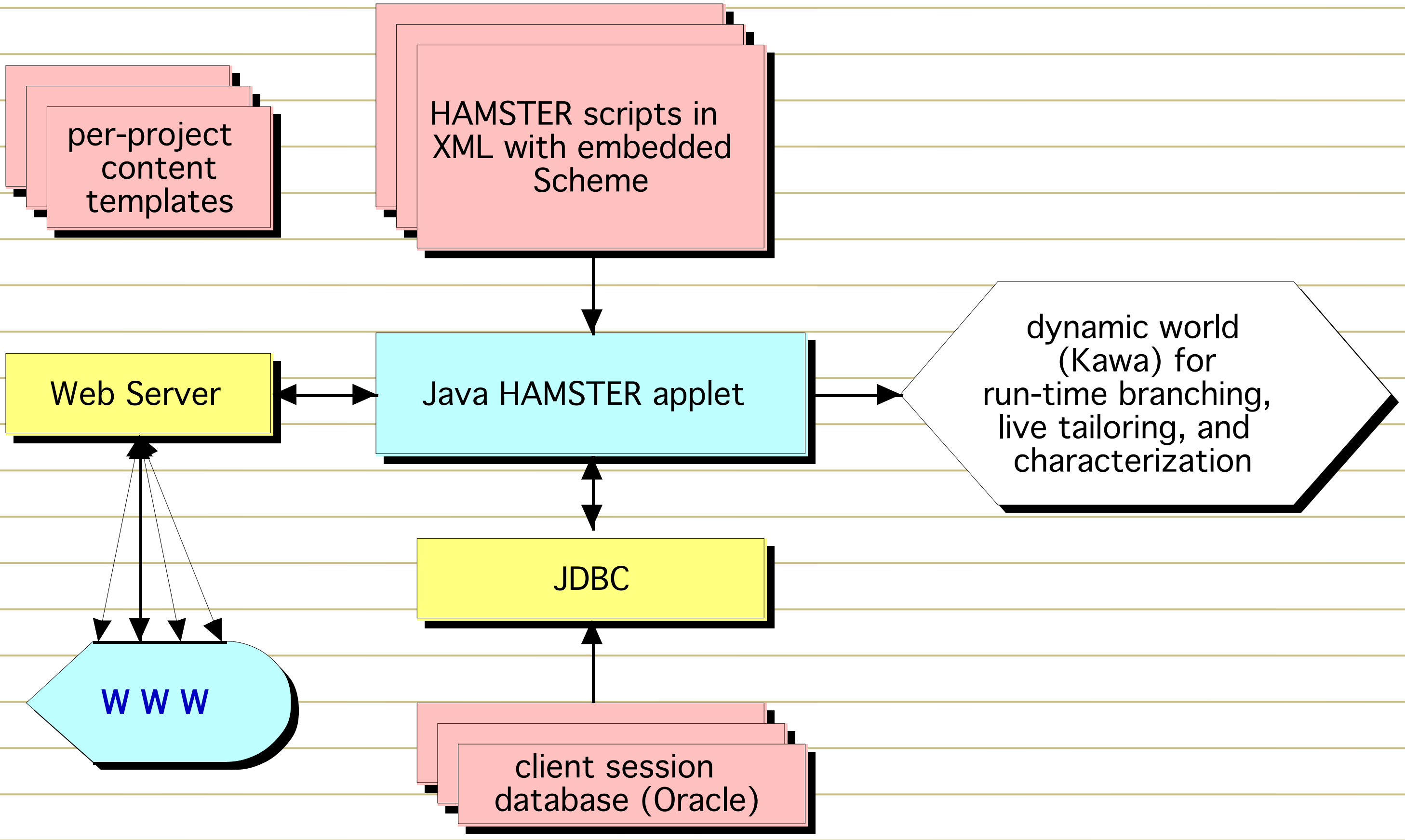


The Layout

- A Graphic Designer builds a layout with Quark XPress™
- The layout is an empty magazine with boxes for tailored text and graphics
- At runtime, the tailoring engine acts as a puppeteer to fill out the boxes and print magazines



Overview of Java HAMSTER


per-project content templates
HAMSTER scripts in XML with embedded Scheme
Web Server
Java HAMSTER applet
dynamic world (Kawa) for run-time branching, live tailoring, and characterization
JDBC
W W W
client session database (Oracle)

Java HAMSTER

Taking HAMSTER and tailored feedback to the World Wide Web

- Runs on any web browser
- Supports elaborate, scripted, and tailored surveys
- Communicates with databases
- Can export data for tailored print

Future Directions

Future research directions enabled by our use of Java, XML and Kawa:

- Dynamic Testing
- Authoring with GUIs such as Xeena

Sample Question

```
{
    surveyTags:  [ 'screening ],
    screenID:  78,
    interactionType:  'chooseMultipleChoice,
    displayText:  "If you tried to quit smoking, how much support or understanding do you
     think you would get from FAMILY?",
    navigationFlags:  [ 'allowGoBack, 'disallowHelp, 'allowSkip ],
    interactionSpec:
    {
            choices: [ "none", "not much", "some", "a lot" ],
            values: [ 1, 2, 3, 4 ],
    },
    storeAnswerIn:  'session . SUPP_FAM,
    nextScreenDefault:  76,
}
```

ID of question

Type of question

Text of question

choices

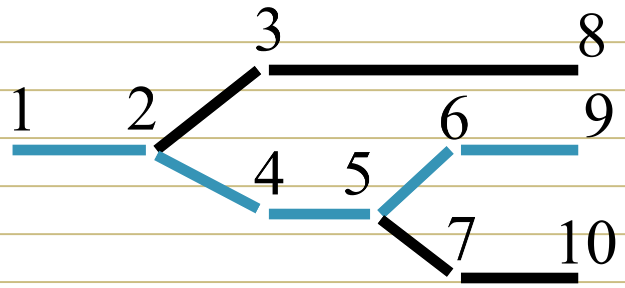
Coding of choices

Store answer in this variable

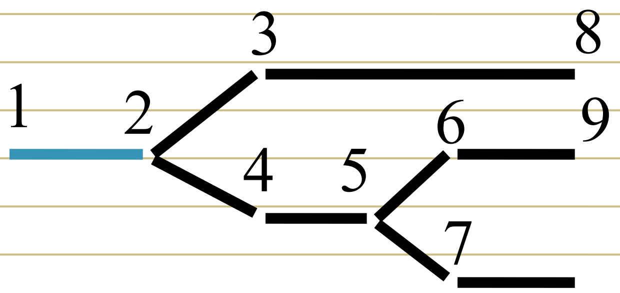
ID of next question

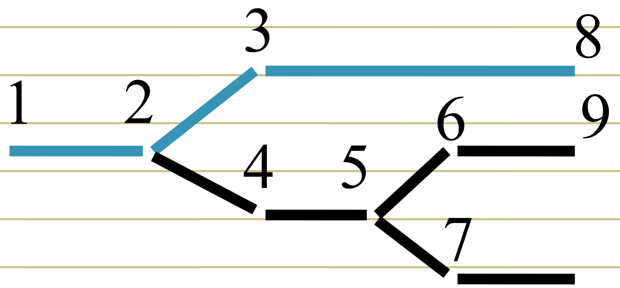
Changing Your Mind

- The visit stack



First pass: 1, 2, 4, 5, 6, 9
Back up four times: 1, 2
Take another branch: 1, 2, 3, 8





Sample of Taxonomy

```
Tobacco:
    Section Tobacco

    Current Smoker=No
    Ever Smoker=No

    if Tobacco1=Y then
         Ever Smoker=Yes

        if Tobacco4=A then amount = less than half pack
        else if Tobacco4=B then amount = half to one pack
        else if Tobacco4=C then amount = one to one point five packs
        else if Tobacco4=D then amount = one point five to two packs
        else if Tobacco4=E then amount = over two packs
        else amount = undefined

        if amount != undefined then
            Current Smoker=Yes

        if Tobacco2=Y then
            Current Smoker=Yes
```

Setting a Variable

Conditional

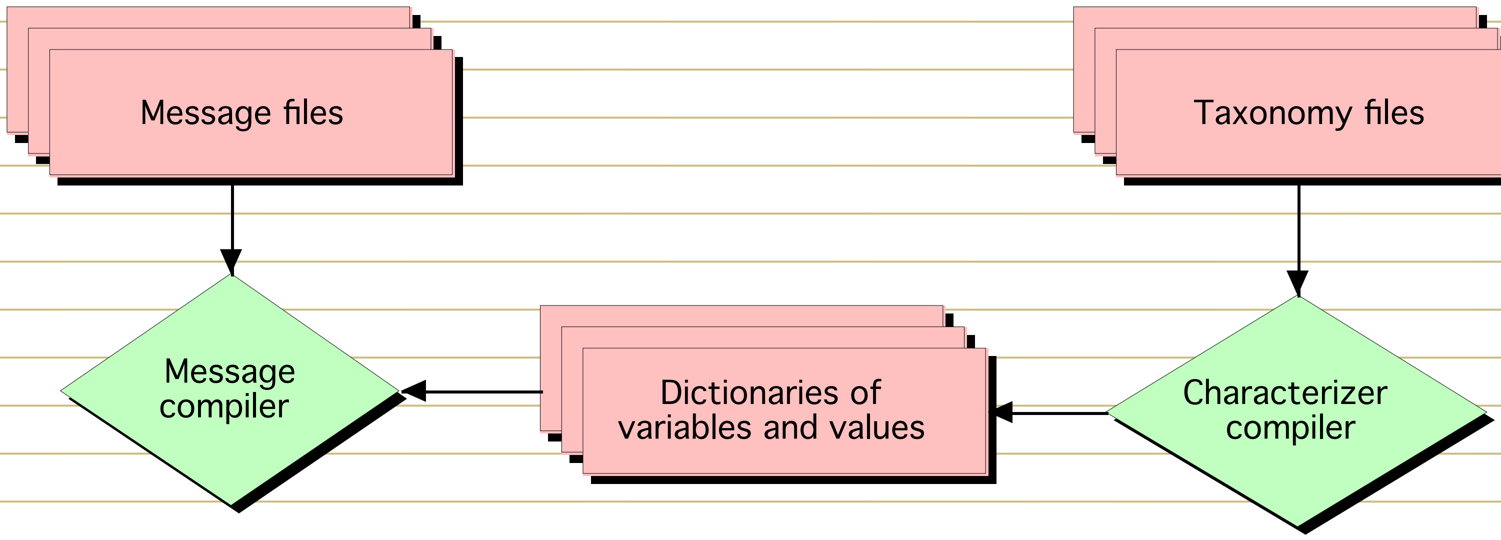
Symbolic Values

Default Value

Dictionaries

- The characterizer compiler generates dictionaries out of the taxonomy files

- Dictionaries are used to verify that all the names and types of variables are consistent



Building the Publisher

- Use Message files + dictionaries
- Compile into a publisher in a process that is similar to building the Characterizer
- The publisher is a C++ program that runs on the Macintosh and uses AppleScript technology to control Quark XPress

Sample of Messages

If the subject cited "I like the taste" as a barrier to quitting

And if stage is not action

"Know" is the name of a box in the layout

```
barrier = Like Taste
    Stage != action
         "@Know:• <B>Did you know<B>
that smoking a brand that you like
less will reduce your urge to
smoke?…"
    Stage = action
         "@Know:• <B>Did you know<B>
that many ex-smokers find it
helpful to keep something in their
mouth to avoid the urge to smoke?
You might try sucking on
toothpicks, cinnamon sticks, ice,
straws…"
```

Technologies Behind Web HAMSTER

- Java - runs in web browsers and has advanced user interface libraries
- XML - the basis of all our future markup languages (messages, scripts, taxonomies)
- Kawa - a simple, but full-featured, dynamic scripting language that runs inside the Java virtual machine

XML Survey for MRF – Sample

```
<QUESTION>
   <PROMPT>
   How long have you been
   smoking cigarettes regularly?
   </PROMPT>
   <ANSWER CHARACTERISTIC=
       "SmokingLength"
       TYPE="single-response">
      <RESPONSE VALUE="Less6Mo">
      Less than 6 months
      </RESPONSE>
      <RESPONSE VALUE="More6Mo">
      More than 6 months
      </RESPONSE>
   </ANSWER>
</QUESTION>
```

XML encoding for a question. It looks like HTML; most users use GUI-based editors to write HTML (and the same thing will happen with XML).

Parts of a multiple-choice question: the PROMPT, the ANSWER, and the individual RESPONSES within the answer

XML Feedback for MRF – sample

```
<SECTION LAYOUTHINT="feedback"
   NAME="Stage feedback"
   <QUESTION GETS="SmokingLength"
   USEIF=
   "(equal? SmokingLength
      “Less6Mo”)">
      <PROMPT>
      Based on your answers, you are
not thinking about quitting smoking.
You told us you have been smoking for
less than 6 months...
      </PROMPT>
```

The script “gets” the stored value of Smoking Length

This is a tiny script written in Kawa

If SmokingLength is less than six months, this prompt is selected and displayed in the layout.

…Any Questions?