

Jeanette uses workspaces

Note that this represents a scenario for an envisaged commercial product. It therefore has already made allowance for what I believe is possible using current technology or extensions of this, but in novel ways. However, it is also intended to be a leading edge proposal, so the 'extensions' are quite radical and certainly suggest potential research directions.

Jeanette comes back from lunch to find the memo from the CEO. She creates a new workspace for the report, sets it to be shared with her secretary and starts to gather data.

I envisage workspaces as 'places' where you place data for your self or for sharing.

Although this is an orthogonal concern I have often thought that the use of places as a metaphor for public/private-ness is compelling. If I want to make something public I simply copy or drag it to somewhere public, just as in real life we put something on show, or hide it away. This is of course exactly what happens with a shared folder in a network file system.

Anyway Jeanette's workspace is shared.

The workspace itself, is a 2D space populated by data items. The data items are all assumed to be tables and 'extended tables' - that is tables where some rows may have repeated sub-rows - for example, departments with a list of employee's details listed under each main department row.

```
xTable = seq xRow
xRow = {
  cols: Name -> Value // actual field values
  children: Name -> seq xRow
}
```

The data items may be represented by name alone, name + column headers or name + column headers + (scrollable list of (some)) records

First she goes to the corporate database. The database has already been fronted as a shared workspace by Information Services using the supplied SQL-agents.

This is an easy mapping - database tables or views become data sources ... however ... the workspace needs additional meta-information in order to work - richer type sets (Name, EmployeeName, CustomerName etc.) and an ER model. These may be entered by hand, as may be the case for the corporate database, or may be inferred.

onCue inferred types by a mix of simple pattern matching rules and bits of code. For example, if it saw something matching the pattern "AA9 9AA" it would suggest possible post code. onCue is only working on the clipboard contents, so has only one data item at a time to work on. When one has a whole table it is possible to look at lots of items in a column. If they all (or most of them) match the pattern we can more reliably infer that it is (usually) a postcode column,

For inferring relationships imagine taking a column A of a table T. Now choose a selection of values from the column v1 ... vn. Now some column B of some other table S, such that the column's types agree, we look up each vi in S.B and see what % are found. If they are all found it says that valuesOf(T.A) subset valuesOf(S.B). If none are found it suggests that T.A and S.B do not intersect at all.

The initial typing is important as this reduces the search space for column matches.

As well as inducing an ER structure this process also refines the types of columns. Some tabel columns are taken as defining - for exmaple, the names in the employee database *are* the eomplee names. So if column T.A is foundto consist solely of names that can be found in the employee table, we can infer that its type is EmployeeName.

She selects the required columns from the payroll table and copies them into the report workspace.

I assume here that the default 'copy' is a live link.

workspace now contains:

Database:

payroll: empno*, name, address, salary*, ...
{ * selected columns }

Bill's project spreadsheet is also already on his public workspace so she can include this.

Using the space as a locus for sharing. 'include' here is meant to mean live-copy of selected tables in current workspace.

The spreadsheet of course needs to be interpreted as a table. This sounds trivial, after all what else is a spreadsheet, but is not quite so. Usually spreadsheets include column headers, titles etc. and indeed may well include several tables of data within the same sheet.

workspace now also contains:

'projects' spreadsheet:

project member

Finally she asks her secretary to put the department staff list file onto the report workspace. The word-processor file has staff listed under each department. The agents within the workspace recognise a pattern of headings and names and infer the structure of the file.

This is similar to the job for a spreadsheet, but word processor tabels are lilely to be formatted less clearly, perhaps with multiple tabs or space to give alignment.

I wrote a 'table recogniser' for onCue that did precisely this job for selecxted tables, it was even able to tell that column/row headers could be numbers and distinguish this form numerical table entries. However, this worked when the whole selection was (possibly) a table, and there is another ste of heuristics required to find data tables within a word-processor file. It would be possible to make users select the parts of a document that should be in the workspace.

workspace now also contains:

'department listing' wordprocessor file:
department names

Having collected all the required information, Jeanette starts to create the collated table for the report. She starts with the department listing and then drills on the names column – the workspace recognises employee names in both the payroll data and the projects listing and offers a choice. Jeanette selects both and now has a listing of people within department and for each person the projects they are engaged on and their salary.

Workspace infers the following ER structure:

Entities:

payroll: empname, salary
department listing: department, name [initially grouped by department]
projects: project, member [initially ordered by project]

Relationships:

'department listing'.name $\leftarrow [0,1]-1 \rightarrow$ payroll.empname
// each employee in at most one department
'department listing'.name $\leftarrow n-m \rightarrow$ projects.member
projects.member $\leftarrow n-1 \rightarrow$ payroll.empname
// employee may be in several projects and sometimes none
// every project member is on the payroll file

See the attached powerpoint steps 1 – 6

She notices that some people are involved in several projects, but realises that there is currently no record of proportionate cost per project. She rings Bill to suggest that he should keep an account of this in future. For the present she creates a new field for each employee and uses standard summary information to get the count of projects for each employee and divide the salary by this, giving an estimated cost per project.

See attached powerpoint steps 7 – 10

Finally, she reorders the table so that it is sorted by project within department, ...

... again uses the standard summary information to select the total of the cost/project of the employees involved in each project, ...

See attached powerpoint steps 11 – 12

I had to think hard to work out whether the calculated columns end up depending on the order they were calculated, but in fact this is not the case because the underlying data structure generated by drill down is a tree – in the graph theoretic sense – so a formula such as 'count(project)' means "count of the number of distinct project names from any given department listing record". As it is a tree there is only one path between any two entities and so the meaning of the formula is unique. ... that doesn't sound totally convincing I know, but I'm pretty sure it is OK!

Reordering the columns is not essential to create the second sum, but makes it a lot more clear!!

... then hides the unwanted columns (staff names and salary).

See attached powerpoint steps 13

The final table is copied into a word-processor document so that she can use the standard corporate report templates. It's now 3pm, so she rings home to arrange a night at the theatre and decides it is time for a coffee break.

Jeanette is about to go home when she notices that the workspace icon has highlighted, showing that something has changed. She opens the workspace to investigate and sees that some of the salary figures are highlighted. Of course, she'd forgotten that the technical staff pay rise was to be added today. She needs to get home quickly to change for the theatre, but happily she had included everything in the workspace as live links. She simply reprints the report and has a pleasant evening.

Next morning she delivers the report to the CEO and explains that it includes the pay rise that went through the previous evening. Later that morning he rings Jeanette, "I know it must have taken some work to get together, but I'm finding this a really useful summary of our position. Can you do this as a regular monthly report." Jeanette smiles to herself. Next month, she opens the workspace and selects 'print'.

