The University Database Integration:  
An AutoMed Example  


Peter McBrien  
Dept. of Computing, Imperial College, pjm@doc.ic.ac.uk  

Abstract  
This reports describes a contrived set of database schemas, and how they are integrated using the AutoMed approach. It serves to illustrate several common techniques used in a AutoMed integration of databases.  

1 Introduction  
AutoMed [2] supports many methodologies for performing data integration and hence forming a network of pathways joining schemas together. Here we describe a simple methodology based on forming union-compatible schemas, the general structure of which is illustrated in Figure 1. Each of the $n$ local schemas $LS_i$ is first transformed into a “union” schema $US_i$. These $n$ union schemas are syntactically identical, and this is asserted by creating a sequence of id transformation steps between each pair of union schemas $US_i$ and $US_{i+1}$, of the form id $(US_i;c, US_{i+1};c)$ for each schema construct $c$. These id transformations between pairs of union schemas are generated automatically by the AutoMed software. An arbitrary one of the union schemas can then be selected for further transformation into the global schema $GS$. This two stage process reflects first schema conformance, followed by schema integration and restructuring.  

![Diagram of AutoMed Integration](image.png)  

Figure 1: A general AutoMed Integration
2 The University Global Database Example

Figure 2 gives some specific schemas that illustrate the integration approach of Figure 1. Primary key attributes are underlined, foreign key attributes are in italics and nullable attributes are suffixed by #.

The schema $US_1$ is the simplest to integrate into $US$ since the only constructs it contains are exactly the same in extent as the same constructs in $US$. Hence the pathway $LS_1 \rightarrow US$ in Example 1 involves only extend transformations to add each of the tables student, college and degree, plus the two attributes of dept which are present in $US$ but not $LS_1$, and add transformations to assert the missing primary and foreign key constraints:

Pathway 1 $LS_1 \rightarrow US$

1. extendTable(\{student\})
2. extendField(\{student.id\})
3. extendField(\{student.sex\})
4. extendField(\{student.dname\})
5. addPK(\{\{student, \{id\}\}\})
6. addFK(\{\{student, \{id\}\}, \{\{dept, \{dname\}\}\}\})
7. extendTable(\{college\})
8. extendField(\{college.cname\})
9. addPK(\{\{college, \{cname\}\}\})
10. extendTable(\{degree\})
11. extendField(\{degree.dcode\})
12. extendField(\{degree.title\})
13. extendField(\{degree.dname\})
14. addPK(\{\{degree, \{dcode\}\}\})
15. addFK(\{\{degree, \{dname\}\}, \{\{dept, \{dname\}\}\}\})
16. extendField(\{\{dept, \{street\}\}\})
17. extendField(\{\{dept, \{cname\}\}\})
18. addFK(\{\{dept, \{cname\}, college, \{\{college, \{cname\}\}\}\}\})
In Example 2, transformations 19–50 use extend transformations to state that the tables student, college and degree in US cannot be derived from LS2. Then 33–50 use the dname attribute of person to derive the dept table in US, and use extend transformations for the two attributes street and cname that cannot be derived from LS2. Finally, in 41–49 the male and female relations of LS2 are restructured into the single sex attribute of staff.

**Pathway 2 LS2 → US**

19. extendTable(\(\{\text{student}\}\))
20. extendField(\(\{\text{student.id}\}\))
21. extendField(\(\{\text{student.sex}\}\))
22. extendField(\(\{\text{student.dname}\}\))
23. addPK(\(\{\text{student.(student.id)}\}\))
24. extendTable(\(\{\text{college}\}\))
25. extendField(\(\{\text{college.cname}\}\))
26. addPK(\(\{\text{college.(college.cname)}\}\))
27. extendTable(\(\{\text{degree}\}\))
28. extendField(\(\{\text{degree.dcode}\}\))
29. extendField(\(\{\text{degree.title}\}\))
30. extendField(\(\{\text{degree.dname}\}\))
31. addPK(\(\{\text{degree.(degree.dcode)}\}\))
32. addFK(\(\{\text{degree.(degree.dname), dept.(dept.dname)}\}\))
33. addTable(\(\{\text{dept}\}\), \(\{\text{x} \mid (y, x) \leftarrow (\text{staff.dname})\}\))
34. addField(\(\{\text{dept.dname}\}\), \(\{x, x\} \mid x \leftarrow (\text{dept})\))
35. extendField(\(\{\text{dept.street}\}\))
36. extendField(\(\{\text{dept.cname}\}\))
37. addFK(\(\{\text{degree.(degree.dname), dept.(dept.dname)}\}\))
38. addFK(\(\{\text{staff.(staff.dname), dept.(dept.dname)}\}\))
39. addFK(\(\{\text{student.(student.dname), dept.(dept.dname)}\}\))
40. addFK(\(\{\text{dept.(dept.cname), college.(college.cname)}\}\))
41. addField(\(\{\text{staff.sex}\}\), \(\{x, x' \mid x \leftarrow (\text{male})\} \cup \{x, x' \mid x \leftarrow (\text{female})\}\))
42. deleteFK((\(\{\text{male.(male.id)}\}\), (\(\{\text{person.(person.id)}\}\)))
43. deletePK((\(\{\text{male.(male.id)}\}\)))
44. deleteField((\(\{\text{male.id}\}\), \(\{x, x\} \mid x \leftarrow (\text{male})\)))
45. deleteTable(\(\{\text{male}\}\), \(\{x \mid (x, x') \leftarrow (\text{staff.sex})\}\))
46. deleteFK((\(\{\text{female.(female.id)}\}\), (\(\{\text{person.(person.id)}\}\)))
47. deletePK((\(\{\text{female.(female.id)}\}\)))
48. deleteField((\(\{\text{female.id}\}\), \(\{x, x\} \mid x \leftarrow (\text{female})\)))
49. deleteTable(\(\{\text{female}\}\), \(\{x \mid (x, x') \leftarrow (\text{staff.sex})\}\))

The example uses the **Intermediate Query Language (IQL)**, which is the default query language supported by the AutoMed implementation. In IQL ++ is the bag union operator and the construct \(\{c \mid Q_1, \ldots, Q_n\}\) is a comprehension [1]. The expressions \(Q_1\) to \(Q_n\) are termed **qualifiers**, each qualifier being either a **filter** or a **generator**. A filter is a boolean-valued expression. A generator has syntax \(p \leftarrow c\) where \(p\) is a **pattern** and \(c\) is a bag-valued expression. In IQL, the patterns \(p\) are restricted to be single variables or tuples of variables.

The pathway LS3 → US contains extend steps 60–67 to add the missing student and college tables, which are textually the same as 19–25. It then renaming deptname, adds the missing attributes of dept, renames person to staff, and adds the missing name attribute. Finally, in steps 68–71 it does the same restructuring as steps 41–49 of LS2 → US, converting the male and female relations into the single sex attribute of staff.

**Pathway 3 LS3 → US**
extendTable(\{student\})
extendField(\{student.id\})
extendField(\{student.sex\})
extendField(\{student.dname\})
addPK(\{student, student.id\})
addFK(\{student, student.dname, dept, student.deptname\})
extendTable(\{college\})
extendField(\{college.cname\})
addPK(\{college, college.cname\})
renameField(\{dept.deptname, dept.dname\})
renameField(\{dept.street\})
extendField(\{dept$cname\})
renameTable(\{person, \{staff\}\})
addField(\{staff, sex\}, \{x, \text{'M'} \mid x \leftarrow \text{'male'}\} ++ \{(x, \text{'F'} \mid x \leftarrow \text{'female'}\})
deleteFK(\{male, \{male.id\}\}, \{person, \{person.id\}\})
deletePK(\{male, \{male.id\}\})
deleteField(\{male.id\}, \{(x, x) \mid x \leftarrow \text{'male'}\})
deleteTable(\{male\}, \{x \mid (x, \text{'M'}) \leftarrow \text{'staff, sex'}\})
deleteFK(\{female, \{female.id\}\}, \{person, \{person.id\}\})
deletePK(\{female, \{female.id\}\})
deleteField(\{female.id\}, \{(x, x) \mid x \leftarrow \text{'female'}\})
deleteTable(\{female\}, \{x \mid (x, \text{'F'}) \leftarrow \text{'staff, sex'}\})

The pathway $LS_4 \rightarrow US$ contains a sequence of extend steps for its missing information.

Pathway 4 $LS_4 \rightarrow US$

extendTable(\{\text{staff}\})
extendField(\{\text{staff.id}\})
extendField(\{\text{staff.name}\})
extendField(\{\text{staff, sex}\})
extendField(\{\text{staff, dname}\})
addPK(\{\text{staff, staff.id}\})
addFK(\{\text{staff, staff.id, dname, dept, dept.dname}\})
extendTable(\{\text{college}\})
extendField(\{\text{college, cname}\})
addPK(\{\text{college, college.name}\})
extendField(\{\text{college, college, name}\})
extendField(\{\text{degree, title}\})
extendField(\{\text{dept, street}\})
extendField(\{\text{dept, cname}\})

The pathway $LS_5 \rightarrow US$ contains a sequence of extend steps for its missing information and also three contract steps to remove the university relation and its attributes.

Pathway 5 $LS_5 \rightarrow US$

renameTable(\{\text{person, person, staff}\})
extendTable(\{\text{student}\})
extendField(\{\text{student.id}\})
extendField([\{student, sex\}])
extendField([\{student, dname\}])
addPK([\{student, [student, id]\}])
addFK([\{student, [student, dname], dept, [dept, dname]\}])
extendField([\{degree\}])
extendField([\{degree, dcode\}])
extendField([\{degree, title\}])
extendField([\{degree, dname\}])
addPK([\{degree, [degree, dcode]\}])
addFK([\{degree, [degree, dname], dept, [dept, dname]\}])
deleteFK([\{college, [college, uname], university, [university, uname]\}])
contractField([\{college, uname\}])
deletePK([university, [university, uname]])
contractField([university, uname])
contractTable(university)

Finally, we list below the pathway from the union schema US to the global schema GS:

Pathway 6 US → GS

addTable([person, [staff] ++ [student]])
addField([person, id], [staff, id] ++ [student, id])
addField([person, name], [staff, name])
addField([person, sex], [staff, sex] ++ [student, sex])
addField([person, dname], [staff, dname] ++ [student, dname])
addPK([person, [person, id]])
addFK([person, [person, dname], dept, [dept, dname]])
deleteFK([student, [student, dname], dept, [dept, dname]])
deletePK([student, [student, id]])
deleteField([student, [x, y] | x ← [student]; (x, y) ← [person, id]])
deleteField([student, [sex, [x, y] | x ← [student]; (x, y) ← [person, sex]])
deleteField([student, [dname, [x, y] | x ← [student]; (x, y) ← [person, dname]])
deleteTable([student], [x | x ← [person]; not (member [staff] x)])
deleteFK([staff, [staff, dname], dept, [dept, dname]])
deletePK([staff, [staff, id]])
deleteField([staff, [x, y] | x ← [staff]; (x, y) ← [person, id]])
deleteField([staff, [name, [x, y] | x ← [staff]; (x, y) ← [person, name]])
deleteField([staff, [sex, [x, y] | x ← [staff]; (x, y) ← [person, sex]])
deleteField([staff, [dname, [x, y] | x ← [staff]; (x, y) ← [person, dname]])
deleteTable([staff], [x | x ← [person]; (x, y) ← [person, name]])

3 Conclusions

See http://www.doc.ic.ac.uk/automed/ for more details of AutoMed.

References