Okuka Lussamaki, Operations Research and Development Division (ORDD), Statistics Canada, June 2001.

Introduction

Statistics Canada's Annual Survey of Manufactures (ASM) collects financial and commodities data from 35,000 Canadian manufacturers. ASM survey questionnaires are sent and returned by mail. There are four types of questionnaires: the long form, the short form, a special form for Quebec, and the head office form.

The ASM's most significant feature is the use of personalized questionnaires. The long questionnaires are personalized for each industry group and business. Personalization is based on the North American Industry Code Standard (NAICS), which determines the industry to which the business belongs. There are 23 personalized questionnaires based on the NAICS.

Before personalization, several long questionnaires are sent to businesses in a given industry. Each questionnaire contains a long list of commodities used by the industry. However, only a small subset will be used for each business based on its needs.

ASM personalization began in 1993 in order to reduce the response burden for respondents. This involved using a questionnaire personalized for each individual business in the commodities input and output sections. The long questionnaire has two parts: the financial section and the commodities section. Only the commodities section is personalized. A commodity is in fact a line with several columns where various information is collected. Historical data from the previous year are used to personalize the section. There are also extra lines where respondents can add other commodities, where applicable. About 18,000 personalized ASM questionnaires are sent.

ASM Functionalities

The various ASM functionalities are:

- CATI applications: - financial section
  - commodities section (personalization)

- Quick Data Entry: Blaise application for rapid data entry.

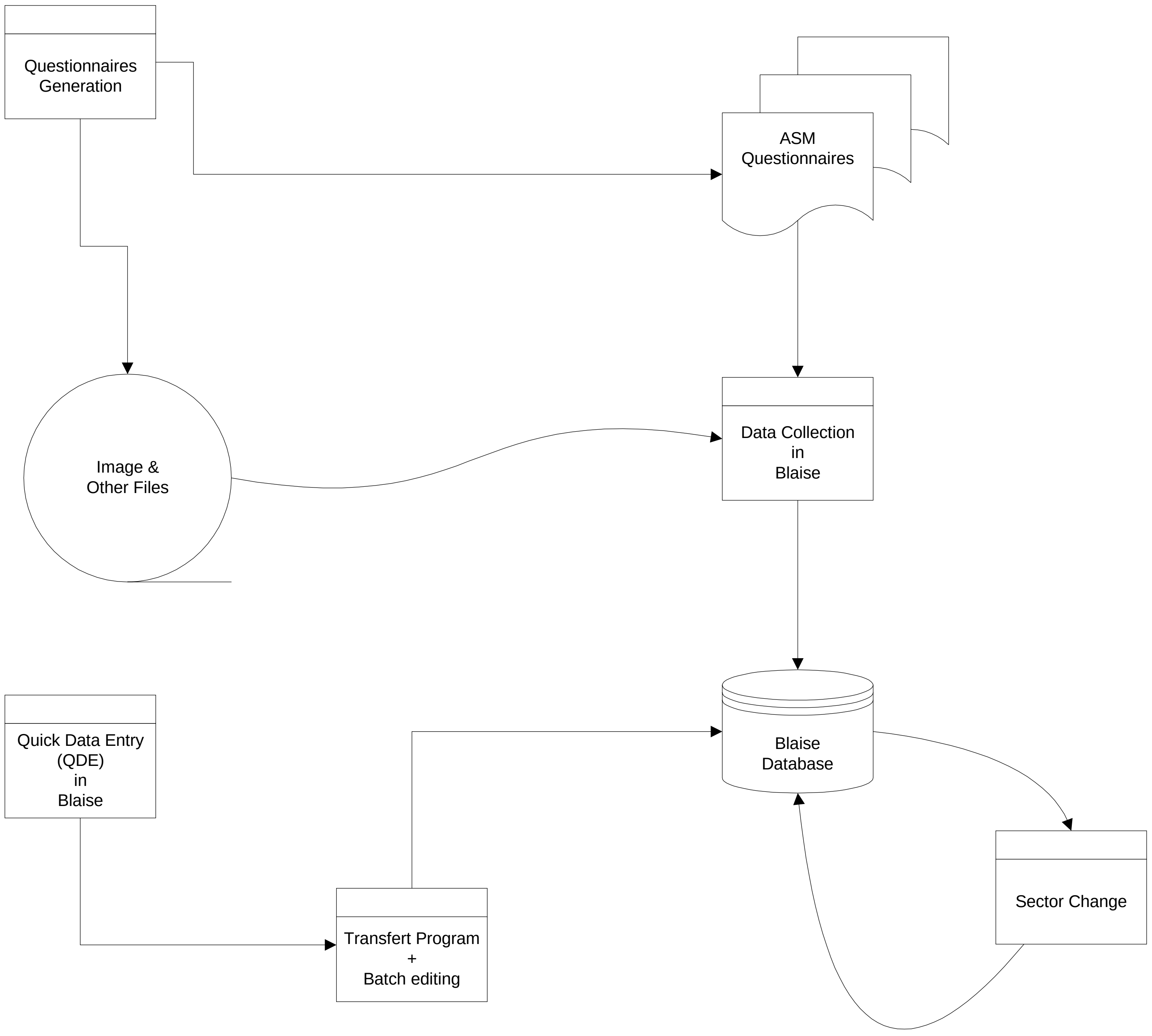
- Automated transfer: Manipula script for data transfer from Quick Data Entry to CATI applications.

- Batch editing: edits are performed in batches at the time of the automated transfer. This gives the respondent a status for follow-up at a later time.

- Change of industry: this occurs when a respondent changes industries from one year to the next.

We will review each functionality as we describe the interactions between the various applications used by the ASM.

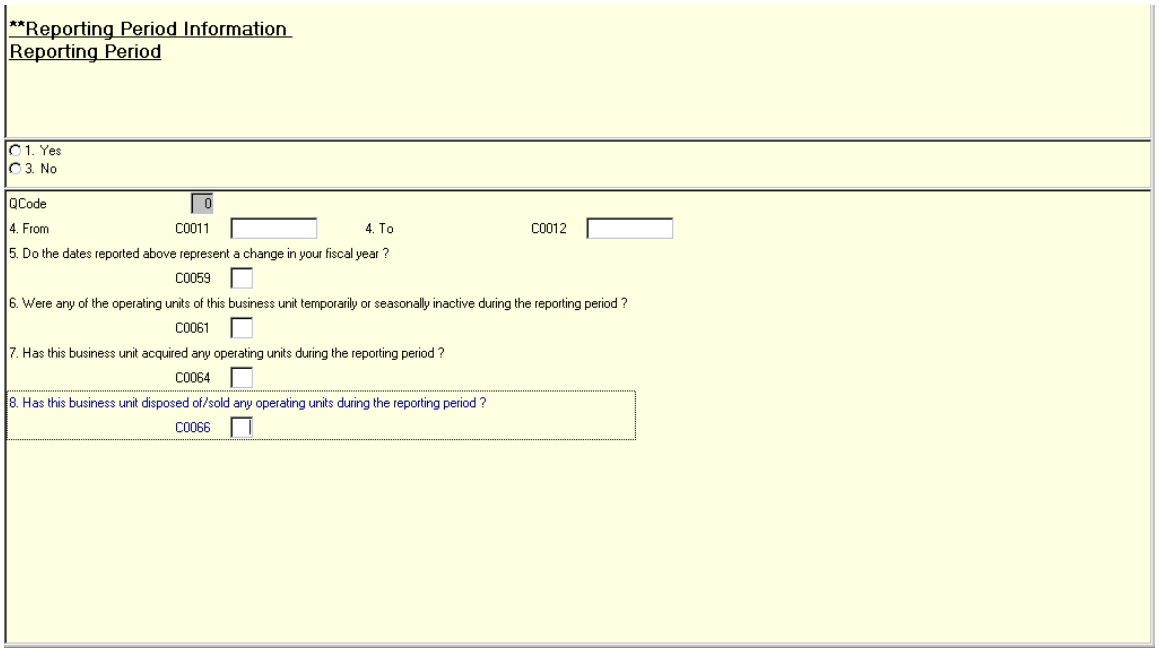
ASM Data Collection


Questionnaires Generation
ASM Questionnaires
Image & Other Files
Data Collection in Blaise
Quick Data Entry (QDE) in Blaise
Blaise Database
Transfert Program + Batch editing
Sector Change

CATI Applications

CATI applications have two major sections: the financial section and the commodities section. The financial section uses the regular Blaise cells.

Here is an example of a financial section:



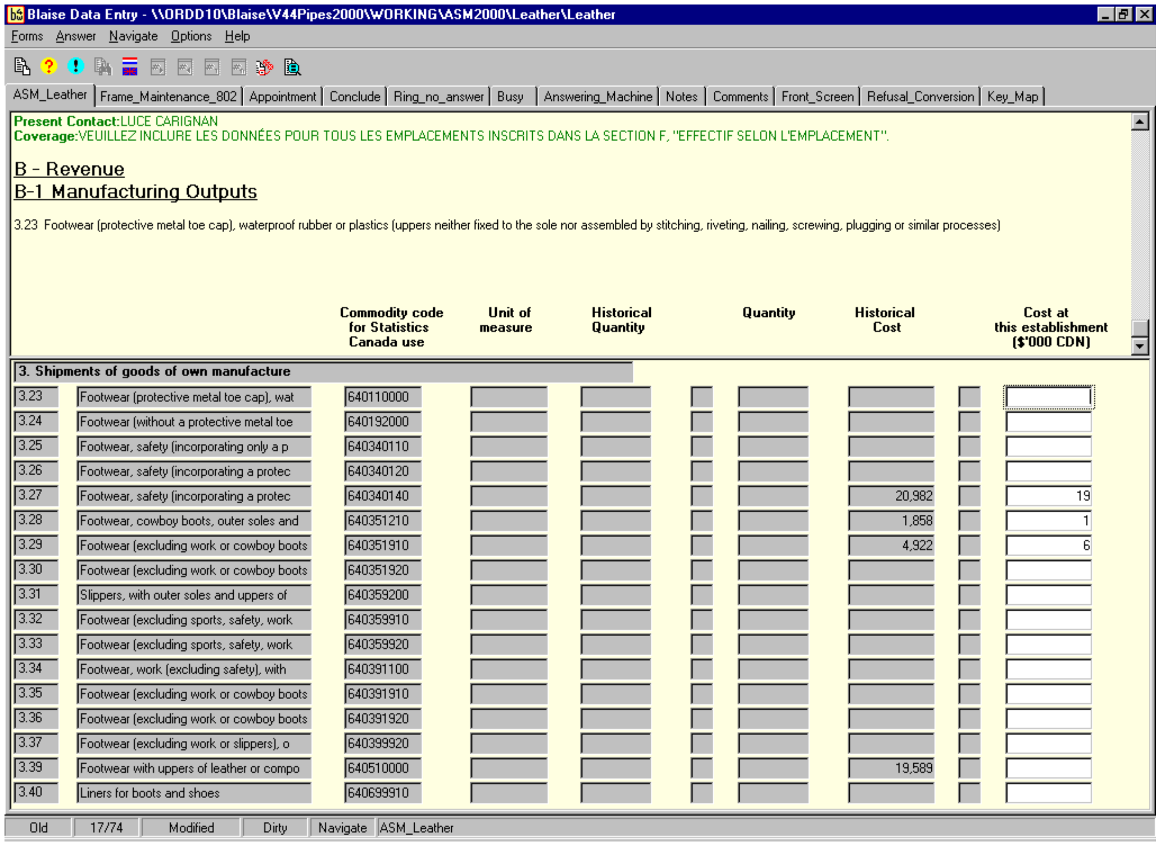
****Reporting Period Information**
Reporting Period

○ 1. Yes
○ 3. No

QCode 0
4. From C0011 4. To C0012
5. Do the dates reported above represent a change in your fiscal year ?
C0059
6. Were any of the operating units of this business unit temporarily or seasonally inactive during the reporting period ?
C0061
7. Has this business unit acquired any operating units during the reporting period ?
C0064
8. Has this business unit disposed of/sold any operating units during the reporting period ?
C0066

Only the commodities section is personalized in CATI applications. Array structures are used to personalize the sections. As mentioned earlier, data from the previous year determine the number of commodities on the respondent's questionnaire and in the Blaise application. Arrays are filled in based on the number of commodities. Each element of the array is a commodity line, and each commodity is a block of several fields. This makes the CATI applications larger and they consequently require considerable resources, particularly in terms of memory.

Here is an example of a commodities section:


Blaise Data Entry - \\ORDD10\Blaise\V44Pipes2000\WORKING\ASM2000\Leather\Leather
Forms Answer Navigate Options Help
ASM_Leather | Frame_Maintenance_802 | Appointment | Conclude | Ring_no_answer | Busy | Answering_Machine | Notes | Comments | Front_Screen | Refusal_Conversion | Key_Map

Present Contact:LUCE CARIGNAN
Coverage:VEUILLEZ INCLURE LES DONNÉES POUR TOUS LES EMPLACEMENTS INSCRITS DANS LA SECTION F, "EFFECTIF SELON L'EMPLACEMENT".

B - Revenue
B-1 Manufacturing Outputs

3.23 Footwear (protective metal toe cap), waterproof rubber or plastics (uppers neither fixed to the sole nor assembled by stitching, riveting, nailing, screwing, plugging or similar processes)

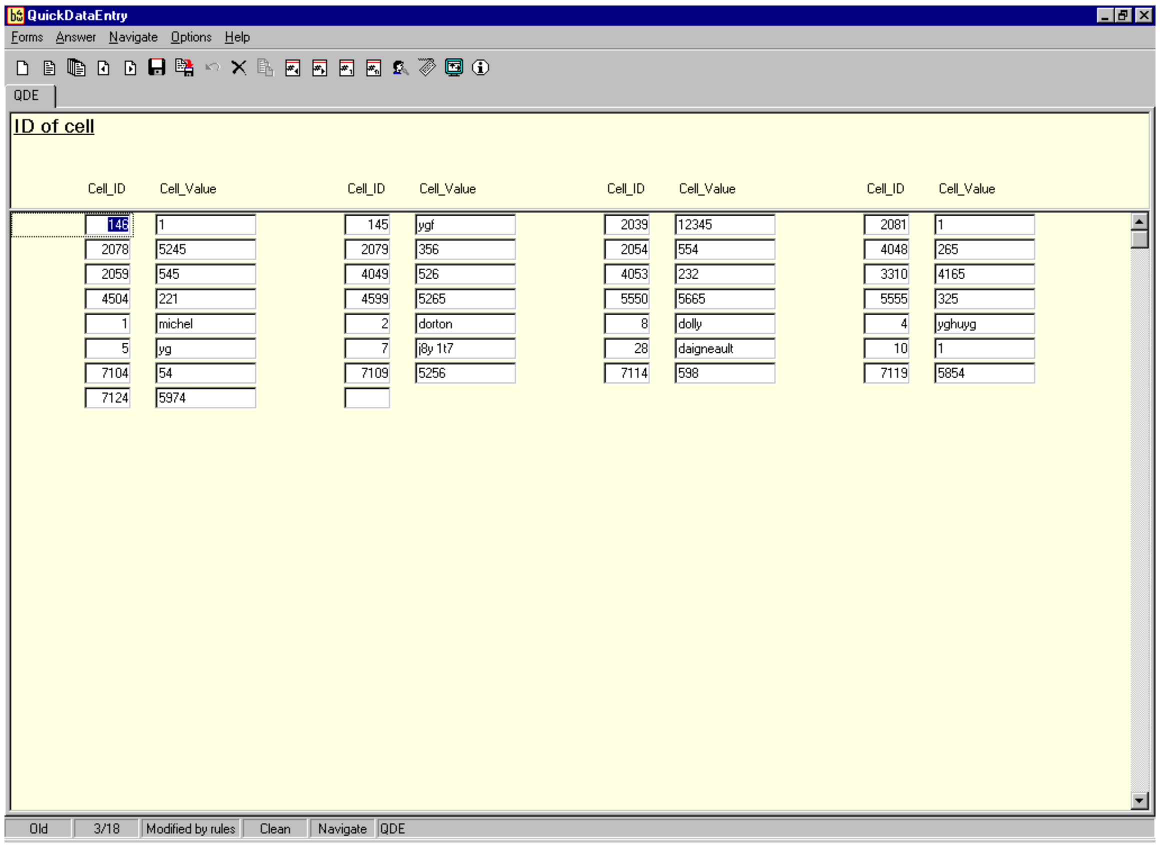
3. Shipments of goods of own manufacture

		Commodity code for Statistics Canada use	Unit of measure	Historical Quantity		Quantity	Historical Cost		Cost at this establishment ($'000 CDN)
3.23	Footwear (protective metal toe cap), wat	640110000							
3.24	Footwear (without a protective metal toe	640192000							
3.25	Footwear, safety (incorporating only a p	640340110							
3.26	Footwear, safety (incorporating a protec	640340120							
3.27	Footwear, safety (incorporating a protec	640340140					20,982		19
3.28	Footwear, cowboy boots, outer soles and	640351210					1,858		1
3.29	Footwear (excluding work or cowboy boots	640351910					4,922		6
3.30	Footwear (excluding work or cowboy boots	640351920							
3.31	Slippers, with outer soles and uppers of	640359200							
3.32	Footwear (excluding sports, safety, work	640359910							
3.33	Footwear (excluding sports, safety, work	640359920							
3.34	Footwear, work (excluding safety), with	640391100							
3.35	Footwear (excluding work or cowboy boots	640391910							
3.36	Footwear (excluding work or cowboy boots	640391920							
3.37	Footwear (excluding work or slippers), o	640399920							
3.39	Footwear with uppers of leather or compo	640510000					19,589		
3.40	Liners for boots and shoes	640699910							

Old | 17/74 | Modified | Dirty | Navigate | ASM_Leather

Quick Data Entry

Given the complexity of the array structures and their size, we thought we could create a small Blaise application (QDE) to quickly enter the data from the paper questionnaire, then transfer the data to the CATI applications.
In this application, only the corresponding cell number in the CATI application and its value are entered. This is an excellent tool for data capture. It accelerates the work of those performing data entry on the computer.
Here is an example of the data entry screen:



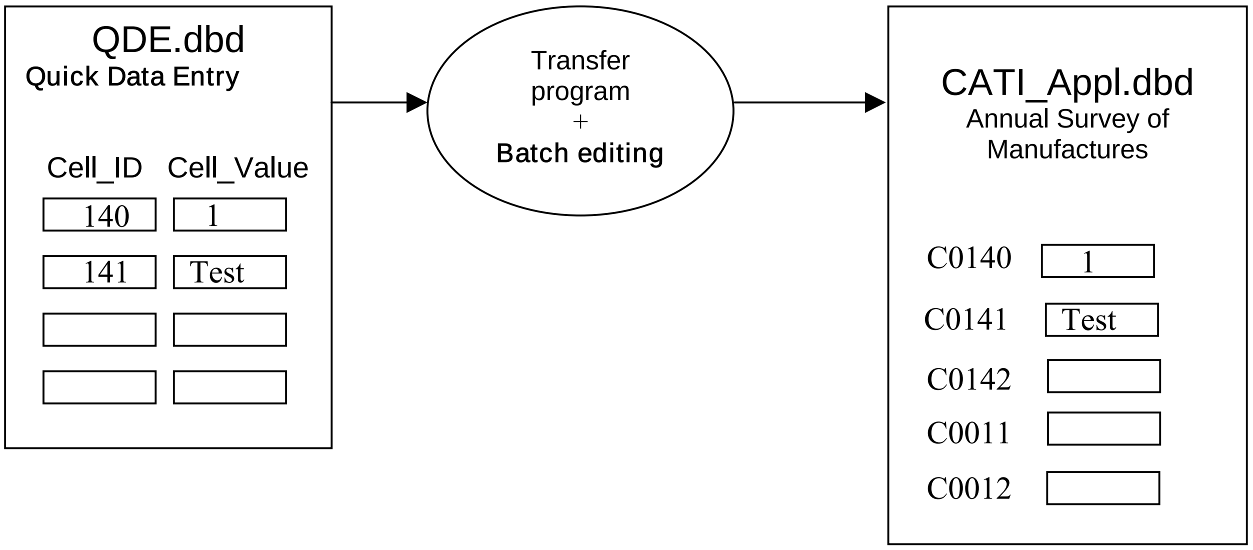
ID of cell

Cell_ID	Cell_Value	Cell_ID	Cell_Value	Cell_ID	Cell_Value	Cell_ID	Cell_Value
146	1	145	ygf	2039	12345	2081	1
2078	5245	2079	356	2054	554	4048	265
2059	545	4049	526	4053	232	3310	4165
4504	221	4599	5265	5550	5665	5555	325
1	michel	2	dorton	8	dolly	4	yghuyg
5	yg	7	j8y 1t7	28	daigneault	10	1
7104	54	7109	5256	7114	598	7119	5854
7124	5974						

Old | 3/18 | Modified by rules | Clean | Navigate | QDE

Automated Transfer

To transfer data from the heads down to a CATI application, a Manipula program is executed through a menu. This transfer allows for a search for each respondent's corresponding cell numbers in the CATI application in order to apply the captured values.
(Show an example of the transfer.)



Batch Editing

Batch edits are executed at the time of the automated transfer to give each respondent a status. The status may be final, partial, etc., depending on the manner in which edits failed. Batch editing reduces follow-up work and prevents other data entry in CATI applications.
Two essential functions are used for batch editing: CHECKRULES and DYNAMICROUTING.

The CHECKRULES function is used in a Manipula program to edit a DataModel, i.e., to scan the fields and assign their respective values without having to physically scan each form for a given survey on-screen.

When the Manipula program is executing, a second function called DYNAMICROUTING declares fields EMPTY if they are "off-route" during execution of the CHECKRULES function.

Example of DYNAMICROUTING and CHECKRULES

DYNAMICROUTING:

```
UPDATEFILE
   UpdateFile2: OutputMeta('Test', BLAISE)
   SETTINGS
       ACCESS = EXCLUSIVE
       ONLOCK = WAIT
       KEY=PRIMARY
       CONNECT = NO
       AUTOCOPY=NO
       DYNAMICROUTING=YES
```

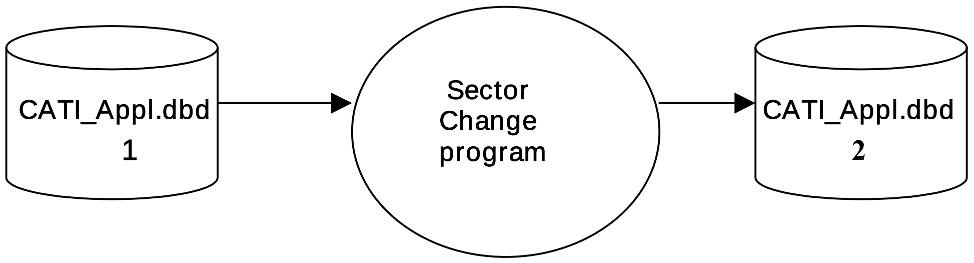
CHECKRULES:

```
MANIPULATE
  UpdateFile1.READNEXT
  WHILE NOT(UpdateFile1.EOF) DO

        IF UpdateFile2.SEARCH(UpdateFile1.QID) THEN
             UpdateFile2.READ
              UpdateFile2.Flag := 1
              UpdateFile2.Qtype:= '0014'
              PROCESS_DATA_FOOD   ---------→   Cxxxx:=
Cell_Value …
                   UpdateFile2.CHECKRULES
                   UpdateFile2.Flag := 0
              UpdateFile2.WRITE
               {UpdateFile1.DELETE}
             UpdateFile2.RESET
        ELSE
           DAYFILE('Errors detected: ' + UpdateFile1.QID + ', ' +  ', does not exist
in the database')
        ENDIF
        UpdateFile1.READNEXT
  ENDWHILE
```

Change of Industry

Another Manipula program is used when a respondent changes its type of activity (for example, when a fabric manufacturer becomes a clothing manufacturer).
This business changes its code under the NAICS and thus requires reclassification in another industry. The change of industry must be indicated in a CATI application in order to change the NAICS and the transfer flag. The business must therefore be transferred from the database of its former industry to that of its new industry under the NAICS.
The Manipula script executes this transfer while respecting all the new changes. It uses a flag to search for the cases to be transferred, then finds the databases for the new NAICS.
Since it is a business in a new industry, the respondent is sent a new questionnaire for that industry. The questionnaire contains the first 10 commodities for each section associated with this industry. This will allow for personalization of the questionnaire and will reduce the response burden for the respondent. Extra lines without commodities are available for the business to make any necessary additions.



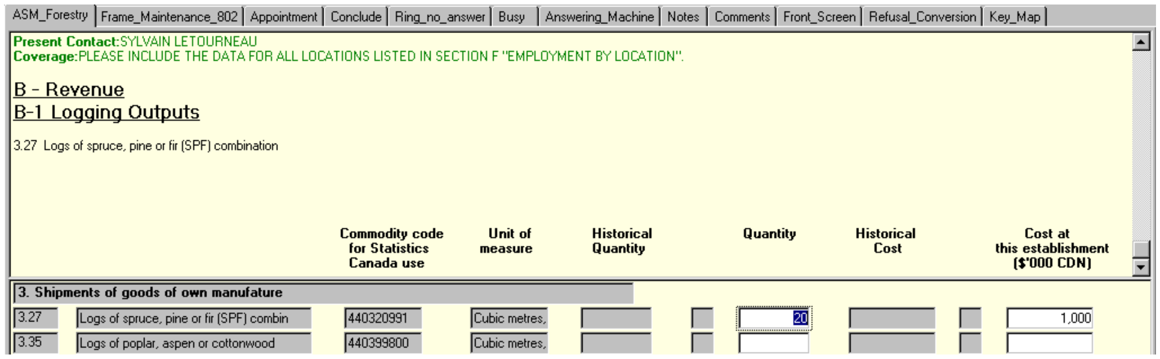
Problems

The following problems were encountered during application development:
1) Screen Personalization
Since we are using array structures for personalization, we had to display only those elements of the array that were filled in. Three problems arose.
- First, Blaise does not allow for the use of dynamic array structures.
- Second, to manipulate the arrays, we had only the loop “FOR” for personalization. Because the array is static, we could not scan it entirely for the on-screen display. Moreover, the loop “FOR” does not allow for an “Exit” when a condition is met. This is possible with Manipula but not with Blaise as yet.
- Third, the commodity descriptions and codes had to be read from an external database. We could have automatically stored them in the applications when the historical data were loaded. But the problem was to display all the fields the client needed to see on-screen. The major difficulty was that these had to be displayed in English or in French, according to the language used by the respondent.

To solve these problems, we began by using one field for each personalized section that would contain the number of commodities recorded the previous year. This field was assigned a value when the historical data were loaded in the database. Thus, using the loop “FOR”, the limit to be verified was the number of commodities, which made it

unnecessary to scan the entire array. At the same time, this allowed us to display only the required number of commodities. Simulating a dynamic array thus solved the first two problems.
For the third problem, we also created a flag to find the commodity description and code in the external database while we scanned the table of commodities. We also put the external database in read-only mode to accelerate our search.

2) Edits for Commodities
The array structure meant that no cell numbers were associated with the commodities. This posed a problem for identifying our edits, i.e., which had failed and which had not. Since these rules were almost the same for all commodities, it occurred to us that we could create generic rules. To identify these from one commodity to the next, we added a field that contains the number associated with each cell entered for commodities. This number is generated only if the cell is entered.
For example, for commodity 10, we entered “20” for the quantity and “1000” for the cost.



Present Contact:SYLVAIN LETOURNEAU
Coverage:PLEASE INCLUDE THE DATA FOR ALL LOCATIONS LISTED IN SECTION F "EMPLOYMENT BY LOCATION".

B - Revenue
B-1 Logging Outputs

3.27 Logs of spruce, pine or fir (SPF) combination

		Commodity code for Statistics Canada use	Unit of measure	Historical Quantity		Quantity	Historical Cost		Cost at this establishment ($'000 CDN)
3. Shipments of goods of own manufature									
3.27	Logs of spruce, pine or fir (SPF) combin	440320991	Cubic metres,			20			1,000
3.35	Logs of poplar, aspen or cottonwood	440399800	Cubic metres,						

The associated numbers will be, for example, “23010” for the quantity and “25030” for the cost. The edits will thus take these numbers into account in order to distinguish them from one commodity to the next.

3) The Length of the Arrays
The length of the arrays presented a problem since it slowed down the applications. Their length varied between 100 and 350 elements from one application to the next. The elements of these arrays are blocks containing several fields.
For example: ArrayCommodity : ARRAY[1..350] OF Bcommodity

It took from one to two minutes to go from one cell to the next. When we asked our clients for the average number of commodities filled in for all applications, the answer was less than 100. We shortened our arrays, which dramatically improved the performance of our applications. We also used a special layout to display less than 20 commodity lines. Changeover time from one cell to the next varied between 4 and 6 seconds.
One fact that should be noted here is that static arrays are a substantial problem with respect to unused memory space. This is particularly so in the case of very large arrays, as with Statistics Canada’s ASM.

Conclusion

All ASM functionalities are managed through a menu. Other changes can be made if Blaise is improved, particularly with respect to arrays, which are currently static and take up a lot of memory space. If dynamic arrays were used, CATI applications would be more powerful and would require less memory. This would also reduce access time to very large databases since we now experience delays when our databases grow larger. Another solution would be relational databases, which would allow for better management of large Blaise databases. We hope to be able to achieve all this with new versions of Blaise.