TALK2C: A Tool For Automatic Migration

Dhinakar Jacob S Dr. Madhuchhanda Das
Applied Technology Group, Tata Infotech Limited
Bangalore - 560 038
INDIA
{selwyn.dhinakar, madhuchanda.aundhe}@tatainfotech.com

Abstract

The IT infrastructure of today’s organizations is often populated with yesterday’s legacy systems which trace an evolutionary path, with ad hoc changes & enhancements spread over a period of time and an inadequate change history. The inherent knowledge is normally contained in the code, which is written in an arcane language, understandable to a few and hard to maintain. Automated translation provides a convenient solution to these issues when maintaining the existing system is the main issue and the source and target languages share the same programming paradigms. TALK2C was developed to automate the conversion of code from SabreTalk to C language. SabreTalk a derivative of PL/1, runs on an earlier version of the TPF (Transaction Processing Facility) system.

1 Source: SabreTalk on TPF

Transaction Processing Facility (TPF), an IBM licensed S/390 operating system, is primarily used in the airline and banking industries since it provides extremely high transaction processing capabilities in a real time environment at very low costs per transaction.

SabreTalk, a special purpose language developed for IBM’s TPF family applications, does not exist on any other platform. SabreTalk programs are constrained to 4K physical containers called segments due to earlier TPF restrictions. Also, SabreTalk programs use the traditional file access methods (i.e. ACPDB, the predecessor of the TPF Data Facility), and uses a database mapping structure called an INCLUDE or INCLUDEAF whose equivalent in C is a header file. Current versions of TPF support TPFDF as opposed to ACPDB hence the need to move away from SabreTalk.

2 Mapping Phase

SabreTalk data structures were analyzed and migrated based on a set of transformation rules specific to the target language ‘C’. The complex features of the source language like ‘OPTIONS’ and variable names which can have #, $, @ values and run time label declarations, are mapped to their equivalent functions in the target system.

Where constructs of the source system (SabreTalk) did not have equivalents within the target system, corresponding functions were added as libraries in the target system.

3 Translation Phase

Figure 1: Design of TALK2C

This translator [as shown in the Figure 1] runs on the Unix platform on SUN Operating System. The source code was preprocessed and fed into the translator. While TALK2C was developed according to the standard compiler writing techniques, it has certain key differentiators as follows.
3.1 Key Differentiators

- The comments are transferred from the source text to the target code at the appropriate place, to enhance readability and maintainability.

- SabreTalk code, being specific to the TPF environment, contains numerous TPF/ACP calls or Macros (approx. 25 - 30%). TPF, the later version of ACP, differs significantly from its earlier version. To migrate the ACP macros into the newer version (TPF) which is compatible with C, the macros are parsed separately and converted into TPF calls on the fly.

- The user defined macros are replaced by equivalent C code. The indentation is provided to facilitate maintenance.

4 Tool Evaluation

4.1 Position of the Tool

The positioning of reengineering tools in general, can be done on various aspects [1]. As this project dealt with major constraints like complete automation and a very large legacy system with a domain specific middle-layer, the tool positioning is done as shown in the figure.

4.2 Tool Performance

4.2.1 Time Taken for Translation

<table>
<thead>
<tr>
<th>Code Size</th>
<th>Time in Min:Sec*</th>
</tr>
</thead>
<tbody>
<tr>
<td>53 K LOC</td>
<td>03:30 (M:S)</td>
</tr>
<tr>
<td>44 K LOC</td>
<td>01:23 (M:S)</td>
</tr>
<tr>
<td>63 K LOC</td>
<td>02:14 (M:S)</td>
</tr>
<tr>
<td>0.102 M LOC</td>
<td>03:17 (M:S)</td>
</tr>
</tbody>
</table>

Table 1: Time Taken for Translation

4.2.2 Comparison of Source and Output

<table>
<thead>
<tr>
<th>Source</th>
<th>Translated System</th>
</tr>
</thead>
<tbody>
<tr>
<td>53 K LOC</td>
<td>61 K LOC</td>
</tr>
<tr>
<td>44 K LOC</td>
<td>52.9 K LOC</td>
</tr>
<tr>
<td>63 K LOC</td>
<td>74.1 K LOC</td>
</tr>
<tr>
<td>0.102 M LOC</td>
<td>0.118 M LOC</td>
</tr>
</tbody>
</table>

Table 2: Comparison between Source & Target

The migrated system is 3-7% larger in size mainly because the source system has embedded comments whereas the translated system has comments in adjacent lines.

System performance is calculated using basic parameters like, ‘Execution Speed’ and ‘Memory Usage’. Herein, the execution speed remains the same as there is no design modification¹, and ‘C’ language is far superior to SabreTalk in handling memory usage.

5 Acknowledgement

The authors would like to thank Sriram Prabu, K., Umapathy, C.S., and Annie Priya, C., of Applied Technology Group, Tata Infotech for their immense help and technical support, without which this effort might have been impossible.

References


¹Depends on the particular transaction and the integration with other systems.