

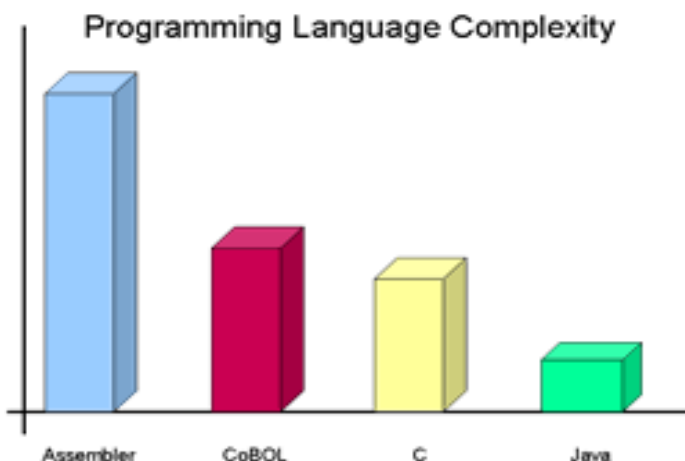


## Industry Issues around Assembler

- **Complex undocumented systems**
- **Ageing/ Greying Skill sets**
- **Availability of trained mainframe personnel**
- **Hidden business Rules**
- **Are your Assembler experts about to retire ?**

Today as never before IT departments are in an environment where rapidly changing business requirements and constantly evolving technologies are the daily norm. SML's products are one of the most cost-effective means of maintaining your Assembler environments or provide a path enabling evolutionary change from one environment to another.

Moving away from legacy systems will save annual maintenance charges, upgrade charges and future training costs. In addition, migrating systems to modern open systems technology will reduce CPU requirements and easily allow future migration of the current applications to another platform.



SML provides a rich toolset which will enable you to take your Assembler systems forward or to make an inventory of your Assembler estate prior to migrating to a new environment.

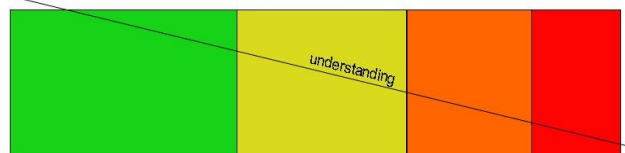
Within our Audit and Discovery service you will find extensive reports that will unveil your Assembler systems underlying complexity. You will be provided with factual complexity scores calculated from the way instructions are used within your environment. Over time you will be able to track whether your systems are becoming more complex. You can insist that complexity scores are not allowed to increase as system changes are delivered. You will be able to clearly see where your testing effort is required and it will put into context the estimating metrics provided by your developers. This is both a management tool and also a development tool.

It is well accepted that Assembler is the most complex environment for development. There were very good reasons for choosing Assembler at the time. When performance was a key factor and choice was limited. However there is no doubt that extensive Assembler estates are becoming less and less cost effective. This complexity also leads to many other issues such as lack of system documentation and knowledge being people dependent with the risks this brings. Many organisations are now looking to migrate to simpler environments like Cobol or 'C'. Although there is a temptation to try to move to the even newer world of Java the big issue here is crossing the paradigm into object orientation. Many organisations find this too big a step with huge risks. SML provides a migration path utilising the Audit and Discovery service to understand the asset base then using the SML Workbench to decide the sequence of migration and then the SML Migration Service itself to complete the transformation.

### System Complexity

#### Instruction Complexity

Standard      Interesting      Clever      High Risk



#### Always Easy instructions

Load address  
Add  
Compare  
Branch

#### Don't want too many

Channel command Word  
Addressing Mode  
Control Section (CSECT)  
Dummy Section (DSECT)

#### Difficult

Start Sub-channel  
Load Access Registers  
Insert Address space

#### Why?

Invalid Page Table Entries  
Move with Destination Key  
Signal Processor



#### Contact: Brian Moroney

email [brian.moroney@smltd.com](mailto:brian.moroney@smltd.com)

office 0044 (0) 8703 898699

mobile 0044 (0) 7739 615846

# Industry Issues—SML Solutions

## Audit and Discovery

- Understand the full extent of your Assembler Estate
- Understand the complexity of the modules that make up your systems

## SML Workbench

- Increase developer productivity by 30%
- Create a Knowledge Repository of your business rules

## SML Migration

- Migrate all or part of your Assembler into Cobol or 'C'
- Uses Automatic Migration with no manual coding
- Achieves functional equivalence

Systems developed using proprietary mainframe IBM Assembler can be a drain on IT departments' resources. Now, the SML migration service reverse engineers legacy applications written in Assembler

### Highlights

- Improves CPU usage, leading to lower overheads
- Generates well structured and hence easy to maintain programs

The SML migration service reads in the Assembler source and generates a functionally equivalent application in well structured COBOL with the appropriate database calls. This provides an identical application to the users and improves CPU usage because of the efficiency of COBOL code.

As COBOL is the most open, portable and widely used programming language in existence, these 'new' applications can, in turn, be easily ported to a number of different environments. Because the programs generated are well structured and consistent, maintenance work becomes a straightforward task. Furthermore, with the run-time free COBOL generated CPU upgrades will not incur additional upgrade charges. Thus, the organisation is left with lower overheads and 'portability' whenever it is needed.

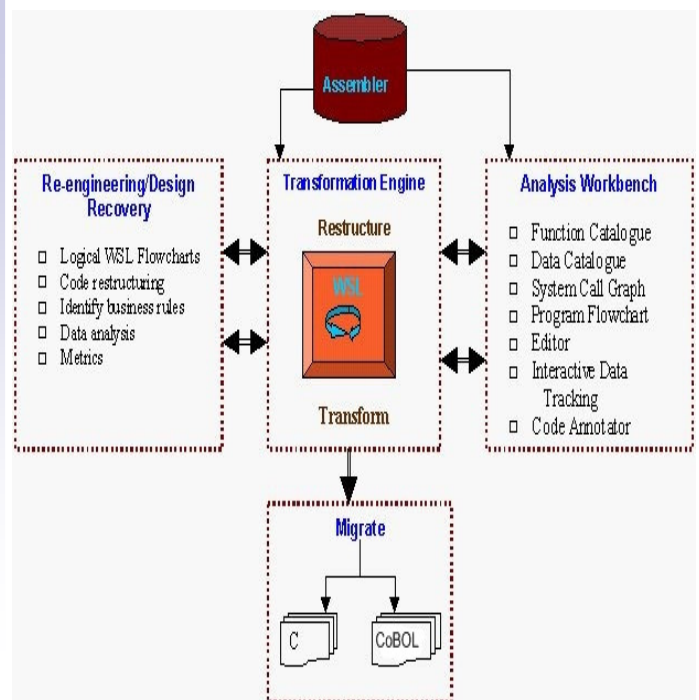
### Re-engineering vs New Development

If you have core business applications that have served your organisation well for many years, using the automated re-engineering option will provide the most cost-effective and speedy route to escaping from a proprietary environment.

Moving away from Assembler legacy systems will save you annual maintenance charges, upgrade charges and future training costs. In addition, migrating your systems to COBOL will reduce CPU requirements and place you in an ideal position should you wish to migrate the current applications to another platform.

### MicroFocus Cobol

If required, it is also possible to generate MicroFocus PC based Cobol, or alternatively, the generated mainframe Cobol can be maintained and/or run in the MicroFocus MainFrame Express environment without amendment.



Audit and  
Discovery

SML Workbench

SML Migration



### Contact: Brian Moroney

email brian.moroney@smltd.com  
office 0044 (0) 8703 898699  
mobile 0044 (0) 7739 615846