Part 15

Fourth Generation Languages
5 Generations of Languages

1st Generation - Machine Language
- zeroes and ones
- wired panels
- switches

2nd Generation - Symbolic Assembly Language - 1950's
- op code mnemonics
- symbolic addresses
- 1-1 to machine language

3rd Generation - High-Level Languages - 1960's
- ALGOL, FORTRAN, COBOL, PL/I, Pascal, Ada, C
- procedural (how)
- 1-many to machine language

4th Generation – High Productivity Languages - 1980's
- some non-procedural aspects
- screen-oriented tools
- data base / data dictionary

5th Generation, - Artificial Intelligence Languages - 1990's
- inference processing
- encodes complex knowledge
- rule base / inference engine
4GL Objectives

- Speed up application building
- Ease of maintenance
- Minimize testing and debugging
- Generate bug-free code from specification
- User-friendly
- End-user generated

4GL Goals

- Order of magnitude less code
- Order of magnitude less programming time
- Based on the observed phenomenon that the number of lines of documented, delivered code produced per programmer per day is constant, regardless of the language
4GL Classification

(Martin) All languages possessing non-procedural aspects are 4GL's
- some have very limited functionality
e.g. Datatrieve, SQL

(Martin) Full-function 4GL
- builds anything that could be built in Cobol
  MANTIS, IDEAL, NATURAL, APPLICATION FACTORY, IEF, IEW
Tool Categories

Programmer Tools (for 2 GLs and 3 GLs)

User Tools (frequently standalone)

Productivity Tools (4 GLs)

A.I. Tools (5 GLs)

CASE Tools (disposable source code)
Programmer Tools

Assemblers (2 GL to speed up 1 GL development)
Macros (originally a 2 GL feature, now used in 3 GLs)
Compilers (originally a 3 GL necessity, now also used in 4 GLs)
Editors
Librarian
JCL
Debug
Test Data Generator
Comparators
Performance Monitors
Optimizers
Code Profilers
Exception Handling Checkers
User Tools

Statistical Packages
Spreadsheets
Simulators
Linguistic Analyzers
Project Managers
Databases
Active Data Dictionaries
Query By Example
Drawing
Desktop Publishing
CAD/CAM
  • Packaged Applications
Productivity Tools (4 GLs)

Spreadsheet Compilers
Database 4 GLs
Context Sensitive Editors
Source Code Control
Make Facility
Software Configuration Management
Code Restructurers
Rapid Prototypers
Screen Simulators
Re-engineering
Full-screen Debuggers
Integrated Query By Example
Application Generators
Report Generators
Graphics Languages
Decision Support
Very High Level Languages
A. I. Tools (5 GLs)

Expert System Shells
Theorem Provers
Natural Language Interpreters
Speech Recognizers
Speech Synthesizers
Pattern Recognition
Game Players
Composers
Robotics Languages
CASE Tools

Enterprise Analysis  
Logical Process Analysis  
Logical Data Analysis  
Design Aids  
Normalizers  
Balancers  
Integrated Project Management  
Code Generators  
Code Analyzers  
Analytic Test Generators  
Metric Analyzers  
Reverse Engineering  
Repository  
Automated Documenters  
Capture  
Toolkits  
Workbenches
Language Generations (James Martin)


Machine Code

Assembler Code

High-level Language

4GLs

Structured Techniques

Knowledge-based Systems

CAD Techniques

A.I. Techniques

O.O.
“Popular” 3 GL Programming Languages in 1987

Ada
Algol
APL
BAL
BASIC
C
C++
Cobol
Coral
Fortran
GPSS
Lisp
Logo
Modula
Pascal
Pilot
PL/1
Prolog
RPG
Simscript
Simula
SmallTalk
Snobol
YACC
Jean Sammet’s Collection of 3 GL Languages in 1969

<table>
<thead>
<tr>
<th>A-2</th>
<th>A-3</th>
<th>ADAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>AED</td>
<td>AESOP</td>
<td>AIMACO</td>
</tr>
<tr>
<td>ALGY</td>
<td>ALTRAN</td>
<td>AMBIT</td>
</tr>
<tr>
<td>AMTRAN</td>
<td>Animated Movie</td>
<td>APT</td>
</tr>
<tr>
<td>B-0</td>
<td>BACAIC</td>
<td>BASEBALL</td>
</tr>
<tr>
<td>BUGSYS</td>
<td>C-10</td>
<td>CLIP</td>
</tr>
<tr>
<td>CLP</td>
<td>COGENT</td>
<td>COGO</td>
</tr>
<tr>
<td>COLASL</td>
<td>COLINGO</td>
<td>COMIT</td>
</tr>
<tr>
<td>Commercial Translator</td>
<td>Computer Compiler</td>
<td>Computer Design</td>
</tr>
<tr>
<td>CORC</td>
<td>CPS</td>
<td>Culler-Fried</td>
</tr>
<tr>
<td>DAS</td>
<td>DATA-TEXT</td>
<td>DEACON</td>
</tr>
<tr>
<td>DIALOG</td>
<td>DIAMAG</td>
<td>DIMATE</td>
</tr>
<tr>
<td>DOCUS</td>
<td>DSL/90</td>
<td>DYANA</td>
</tr>
<tr>
<td>DYNAMO</td>
<td>DYSAC</td>
<td>English</td>
</tr>
<tr>
<td>473L Query</td>
<td>FACT</td>
<td>FLAP</td>
</tr>
<tr>
<td>FLOW-MATIC</td>
<td>FORMAC</td>
<td>FORTRANSIT</td>
</tr>
<tr>
<td>FSL</td>
<td>GAT</td>
<td>GECOM</td>
</tr>
<tr>
<td>GPL</td>
<td>GRAF</td>
<td>Graphic Language</td>
</tr>
<tr>
<td>ICES</td>
<td>IDS</td>
<td>Information Algebra</td>
</tr>
<tr>
<td>IPL-V</td>
<td>IT</td>
<td>JOSS</td>
</tr>
<tr>
<td>JOVIAL</td>
<td>Klere-May</td>
<td>L6</td>
</tr>
<tr>
<td>Laning and Zierler</td>
<td>LDT</td>
<td>Lincoln Reckoner</td>
</tr>
<tr>
<td>LOLITA</td>
<td>LOTIS</td>
<td>MAD</td>
</tr>
<tr>
<td>MADCAP</td>
<td>Magic Paper</td>
<td>MAP</td>
</tr>
<tr>
<td>MATHLAB</td>
<td>MATH-MATIC</td>
<td>Matrix Compiler</td>
</tr>
<tr>
<td>META 5</td>
<td>MILITRAN</td>
<td>MIRFAC</td>
</tr>
<tr>
<td>NELIAC</td>
<td>OCAL</td>
<td>OMNITAB</td>
</tr>
<tr>
<td>OPS</td>
<td>PAT</td>
<td>PENCIL</td>
</tr>
<tr>
<td>PRINT</td>
<td>Proposal Writing</td>
<td>Protosynthex</td>
</tr>
<tr>
<td>QUICKTRAN</td>
<td>Short Code</td>
<td>Simulating Digital</td>
</tr>
<tr>
<td>Systems</td>
<td>SOL</td>
<td>Speedcoding</td>
</tr>
<tr>
<td>SPRINT</td>
<td>STRESS</td>
<td>STROBES</td>
</tr>
<tr>
<td>Symbolic Mathematical Laboratory</td>
<td>TMG</td>
<td>TRAC</td>
</tr>
<tr>
<td>TRANDIR</td>
<td>TREET</td>
<td>UNCOL</td>
</tr>
<tr>
<td>UNICODE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
James Martin’s List of 4 GLs in 1986

<table>
<thead>
<tr>
<th>GL Name</th>
<th>Company/Developer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADS/OnLine</td>
<td>by Cullinet</td>
</tr>
<tr>
<td>Application Builder</td>
<td>by James Martin Associates</td>
</tr>
<tr>
<td>Application Factory</td>
<td>by Cortex</td>
</tr>
<tr>
<td>Datatrieve</td>
<td>by Digital Equipment Corporation</td>
</tr>
<tr>
<td>EASYTRIEVE</td>
<td>by Pansophic Systems</td>
</tr>
<tr>
<td>EXPRESS</td>
<td>by Management Decision Systems</td>
</tr>
<tr>
<td>FOCUS</td>
<td>by Application Builders</td>
</tr>
<tr>
<td>GENER/OL</td>
<td>by Pansophic Systems</td>
</tr>
<tr>
<td>IDEAL</td>
<td>by Applied Data Research</td>
</tr>
<tr>
<td>INFO</td>
<td>by Henco</td>
</tr>
<tr>
<td>INQUIRE</td>
<td>by Infodata Systems</td>
</tr>
<tr>
<td>INTELLECT</td>
<td>by Artificial Intelligence Corporation</td>
</tr>
<tr>
<td>LINC</td>
<td>by Burroughs</td>
</tr>
<tr>
<td>MANTIS</td>
<td>by Cincom</td>
</tr>
<tr>
<td>MAPPER 10</td>
<td>by Sperry (Unisys)</td>
</tr>
<tr>
<td>MARK V</td>
<td>by Informatics General</td>
</tr>
<tr>
<td>MIMER</td>
<td>by Uppsala Data Center</td>
</tr>
<tr>
<td>NATURAL</td>
<td>by Software AG</td>
</tr>
<tr>
<td>NOMAD2</td>
<td>by D&amp;B Computing Services</td>
</tr>
<tr>
<td>Personal Data Query</td>
<td>by Honeywell Information Systems</td>
</tr>
<tr>
<td>RAMIS II</td>
<td>by Mathematica Products Group</td>
</tr>
<tr>
<td>SAS</td>
<td>by SAS Institute</td>
</tr>
<tr>
<td>SPSS</td>
<td>by SPSS</td>
</tr>
<tr>
<td>SYSTEM W</td>
<td>by COMSHARE</td>
</tr>
<tr>
<td>TELL-A-GRAF</td>
<td>by Integrated Software Systems</td>
</tr>
<tr>
<td>UFO</td>
<td>by Oxford Software</td>
</tr>
<tr>
<td>Umbrella System</td>
<td>by Hogan Systems</td>
</tr>
<tr>
<td>Universe</td>
<td>by Computer Associates International</td>
</tr>
<tr>
<td>USE-IT</td>
<td>by Higher Order Software</td>
</tr>
</tbody>
</table>
## List of 5 GLs in 1988

<table>
<thead>
<tr>
<th>ADS</th>
<th>ALEX</th>
<th>Arity Expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>ART</td>
<td>Auto-Intelligence</td>
<td>C++</td>
</tr>
<tr>
<td>Emycin</td>
<td>Envisage</td>
<td>ES Expert</td>
</tr>
<tr>
<td>ESP Advisor</td>
<td>ExpertEase</td>
<td>Expert-2</td>
</tr>
<tr>
<td>EXSYS</td>
<td>FLOPS</td>
<td>Goldworks</td>
</tr>
<tr>
<td>Guru</td>
<td>HUMBLE</td>
<td>IN-ATE</td>
</tr>
<tr>
<td>Insight 2+</td>
<td>Intelligence Compiler</td>
<td>KDS3</td>
</tr>
<tr>
<td>KEE</td>
<td>KES</td>
<td>Keystone</td>
</tr>
<tr>
<td>Knowledge Craft</td>
<td>Knowledge Maker</td>
<td>Knowledge Pro</td>
</tr>
<tr>
<td>LISP</td>
<td>MacSMARTS</td>
<td>MIS Ex</td>
</tr>
<tr>
<td>M1/S1</td>
<td>NEXPERT/Object</td>
<td>OPS5</td>
</tr>
<tr>
<td>OPS83</td>
<td>PC-Easy/PC-PLUS</td>
<td>Personal Consultant</td>
</tr>
<tr>
<td>PROLOG</td>
<td>Rbest</td>
<td>Rulemaster</td>
</tr>
<tr>
<td>Superexpert</td>
<td>VP-Expert</td>
<td>WizdomMice</td>
</tr>
<tr>
<td>Xi Plus</td>
<td>XSYS</td>
<td>YES/OPS</td>
</tr>
<tr>
<td>1st-CLASS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Selection Criteria – 1st Generation

Basically the selection criteria for a processor
   Registers available
   Addressing modes
   Memory addressing scheme
   Symmetry of the instruction set
   Types of arithmetic and operations supported
   Compatibility with other instruction sets
   Operating system(s) available
   Microcoded / Trapped instructions (-)
   Interrupts / buffers
   Block / parallel instructions

Tools to consider
   Availability of 2nd generation languages
   Availability of cross-assemblers
   Availability of editors, debuggers, etc.
Selection Criteria – 2\textsuperscript{nd} Generation

Op code / addressing mode facility
Symbolic addressing / forward or backward referencing
Pseudo ops
Symbolic debugger
Checkpoing / breakpoint facility
Assembly / disassembly speed
Macro facility
  - accept parameters
  - extent of items allowing parameterization
Macro packages available
Operating system interaction
  - symbolic links to operating system addresses
  - input/output control system (IOCS) calls
  - interrupt handling
Library / Source code control facility (SCCF)
Selection Criteria – 3rd Generation

Data types and structures
Storage management / arrays
Subprogram / function / procedure facility
Libraries of available routines
Program flow control structures
Data control mechanisms
Compiled vs. interpreted
Binding time
Operating environment
Theoretical model
Support for structured / object-oriented methodologies
Portability
Selection Criteria – 4th Generation

Simply query facility
Query by example
Full-function query / update / management facility
Report writer facility
Graphics interface facility
Decision support facility
Application generator facility
Active data dictionary
Specification language
Semi-procedural programming language
Existing parameterized application packages
Selection Criteria – 5th Generation

Knowledge representation schemes
Knowledge engineering tools
  - problem description
  - knowledge base
  - rule base
Inference / search mechanism
Rapid prototyping / expansion / porting capability
User interface
  - defaults
  - solution status
  - graphics
  - stubs
Performance / productivity
Tools
  - interpreters
  - compilers
  - debuggers
Suitability for a varied audience
Learning tools (for human user)
Pre-existing knowledge
Carma McClure’s List of 4GL Desired Features

Functionality – same as any 3GL
Query languages – SQL and more
Data entry / update facilities – can get to procedural
Procedural language – can reduce to textual source code
Screen design / screen handling
Data validation – against an active data dictionary
Integrity controls – foreign key, nulls, etc.
Auditability – audit rollback and recovery
Infra structures – data structure hierarchy, objects, conceptual integrity
Testing aids – tests for completeness and correctness
Training – generate training documents from code
User friendliness
Reports and graphics generators
Decision support tools – at least simple decision trees
Word processing
Database characteristics
Library facilities
Security features
System characteristics – homogeneous look and feel
Performance
Documentation – on use of tool