Introduction to RPG

iSoftwerks, Inc

Steve Croy

- RPG, Report Program Generator
  - Released in 1959 to replicate punch cards on the IBM 1401 computer
- RPG II
  - Introduced in 1969 for IBM S/3
- RPG III
  - Released for S/38 (1979) with new logic op codes
- RPG/400
  - AS/400 version – (renamed) same compiler as RPG III
- RPG IV introduced in 1994
  - OS/400 V3R1
  - ILE compiler available
- RPG free-format 2001
  - OS/400 V5R1
- RPG OA 2010
  - RPG Open Access

RPG, a Half-Century and Counting

C Programming Language

- Bell labs 1969
- ANSI C 1989 (C89)
- ISO revision 1995 (ISO standard)
- Revised 1999 (C99)
- Updated December 2011 (C11)
- Perception – Modern Language
  - Reality – over 40 years old

Ancient RPG (Perception vs. Reality)

RPG Programming Language

- IBM 1959
- RPG II 1969
- RPG III 1979
- RPG IV 1994 V3R1
- RPG Free 2001 V5R1
- RPG OA 2010 V7R1
- Perception – Obsolete Language
  - Reality – as current as any tool

RPG II
Column Indicator Driven

```
C     21NU2                  SETON                                    62
C     21N62 U2               GOTO END
C     21       EMPNO         CHAINMASTER                   99
C     21                     ADD  1          CNT       30
C     21                     ADD  CSALE      TCSALE    80
C     21                     ADD  CGROSS     TCGROS    80
C     21                     ADD  CPD        TCPD      80
C     21                     ADD  CCL        TCCL      80
C     21                     ADD  CRES       TCRES     80
C     21                     ADD  YSALE      TYSALE    80
C     21                     ADD  YGROSS     TYGROS    80
C     21                     ADD  YPD        TYPD      80
C     21                     ADD  YCL        TYCL      80
C     21                     ADD  YRES       TYRES     80
C     21                     Z-ADDCPERG      CPERG1    52H
C     21                     Z-ADDYPERG      YPERG1    52H
C              END           TAG
```

Fixed Column but With Logic Operators

```
C                              DOU        EOFIND = *ON
C                              READ       SYNBCHL1                                      54
C                              MOVE       *IN54           EOFIND
C                              IF         EOFIND = *OFF
C                              ADD        1               SFLRCN
C                              ADD        1               SFLLOD
C                              Z-ADD      SFLRCN          RRNSI
C                              Z-ADD      SFLRCN          Z$RRN1
C                              Z-ADD      DBRRN           Z$DBRN
C                              EVAL       Z$OPT = *BLANKS
C      *CYMD                   TEST(D)                    STENDT                        99
C      *IN99                   IFEQ       *OFF
C      *CYMD                   MOVE       STENDT          $$DATE
C      *MDY                    MOVE       $$DATE          Z$ENDT
C                              ENDIF
C                              Z-ADD      STGDCT          Z$TOTR
C                              ADD        STERCT          Z$TOTR
C                              WRITE      SYN910S1
C                              IF         SFLLOD >= SFLMAX
C                              MOVE       *ON             EOFIND
C                              ENDIF
C                              ENDIF
C                              ENDDO
```

```
DOU NoMoreRecords or sfllod >= sflmax;
   MoreRecordsRemain = NextObject();
   IF MoreRecordsRemain;
       OBJECT = GetObjectData();
       sflrcn = sflrcn + 1;
       sfllod = sfllod + 1;
       RRNSI = SFLRCN;
       Z$RRN1 =SFLRCN;
       z$opt = *BLANKS;
       z$desc = exdesc;
       zfstat = *blanks;
       zfobnm = dspatr + exobnm + typatr + exobtp;
       WRITE SC0320S1;
   ELSE ;
       Sflend = *ON;
       NoMoreRecords = *ON;
   ENDIF;
ENDDO;
```

```
for (int i=0; i < columns; i++)
{ // parse the value for this column
  rs = getParameter("C" + i);
  if (rs != null) {
    try { values[i] = Integer.parseInt(rs); }
    catch (NumberFormatException e) { values[i] = 0; }
  }
  if (values[i] > max) { max = values[i];
  }

}
```

Convergence: Free Format RPG resembles Java, more than it resembles Fixed RPG II

```
for x = 1 to 36;
    if %subst(arr(x):1:1) = 'A';
      arr(x) = 'F' + %subst(arr(x):2);
   endif;
   if arr(x)<> *blank;
      info(i) = arr(x);
      i  +=  1;
   endif;
endfor;
```

- Job sites:
  - 10X more Java than RPG jobs
  - 5X more PHP that RPG jobs
- TIOBE index Sep 2013
  - Number 1, C
  - Number 2, Java
  - Number 94, RPG
- RPG, Role Playing Games are popular, however!

Is RPG Popular?
That Depends on Interpretation

- Card decks, disk, CRT's
  - RPG II simple business language
- S/38, relational DB
  - RPG III with structured programming constructs
- PC
  - VB, PowerBuilder, other graphical languages
- Portable OS/Unix/Linux
  - C and Perl
- WWW
  - Java, PHP, JavaScript and CGI
- Mobile Devices?
  - LUA?

Language Adoption is a Reaction to Changing Technology

- Component-based Design
  - Black box process that is able to communicate with other components with well defined arguments and return value
    - In RPG think in terms of procedures and service programs
- Object-oriented Programming
  - Components built with encapsulation, modularity, polymorphism, and inheritance
    - Think Java, Class, methods
- Declarative Programming
  - Declarative programming is often defined as any style of programming that is not imperative
    - UIM, define the contents of a panel, Presentation Manager arranges the the display.
    - In the web world: HTML, it describes what goes on a page, not how it goes on a page. (The browser or CSS defines the presentation.)

No Software Architecture Change

- Allows for subroutines
- Supports multiple language calls
  - Subprogram invocation
- Capable of CGI through API calls

Component Based Design & the RPG ILE Model

- Supports subprocedures
  - Internal invocation
- Supports dynamic calls to subprograms
- Supports calls to service programs
- Offers binding of same language, or multi-language modules into an executable program or service program
  - Few IBM i shops embrace CBD

Why No Rush to Java?

- IBM actions have slowed GUI for i/OS
    - RPG is backward compatible
    - Moving to new OS release does not require re-development
- Green apps are efficient
    - 5250 DE is still faster than GUI
    - It is often faster (project-wise) and cheaper to leverage legacy applications than to develop new applications
- Java changes
    - Sun Microsystems Java EE, Java ME, Java SE
        - Remember the law suit?
    - Java lacks any formal standardization recognized by Ecma International, ISO/IEC, ANSI, or other third-party standards organization
        - Sun Microsystems made a Java submission, but subsequently withdrew it.

- It works—and it works consistently from machine to machine, company to company, from one release level to the next.

- ILE RPG has integrated modular API support allowing it to incorporate other language modules

- Existing staff knows RPG.

- SQL integration allow RPG procedures to be registered as UDF's

- Even ancient code (such as RPG II) works in new OS releases

    - Allows leveraging of existing applications.

- Solid integration with the DB and OS.

    - Efficient application execution

- Extensive multi-layered exception handling (arguably the best job log in the industry).

Reasons to Develop in RPG

- Perceptions need to be overcome
  - *The green screen looks like DOS, something interpreted as old and obsolete, by those of an age to remember what DOS looked like.*
- Even if legacy RPG is still the primary business tool, a web enablement solution is a must going forward.

```
0137.00     IF pfcpo <> *blanks;
0138.00      IF workSel <> *blanks;
0139.00         workSel = %trim(worksel) + ' and ';
0140.00      ENDIF;
0141.00      WorkSel = %trim(workSel) + ' adhpo >= ' + qt + pfcpo + qt;
0142.00     ENDIF;
0143.00
0144.00     // *---
0145.00     // * Check Order
0146.00     // *---
0147.00
0148.00
0149.00
0150
```

Reasons not to use RPG

*"It's Not Easy Being Green"* –
Kermit the Frog

- IBM said the solution to moving to GUI was Java
  - 1998 Ads pushed Java as an application solution
- IBM then announced that the path to GUI was with WebFacing
- IBM then announced that Host Access Transformation Services (HATS) was the method of choice
- Most recent direction from IBM is EGL (Enterprise Generation Language) or EDT (EGL Development Tools)
  - *Or is it PHP?*

IBM's Roadmap: Subject to Change without prior warning

Beyond RPG: ILE

- SQL
  - Use Embedded SQL to take advantage of aggregate functions
  - DDS to DDL to create tables
    - Move RI and constraint logic to DDL
  - Use SQL to create stored procedures
    - To interact with web-based services
- Use the array of features of ILE
  - Build modular applications
    - Separate business rules from presentation
    - Create service programs
      - To promote reusable code
      - To perform common functions
    - Create binding directories to simplify program creation
    - Use binder source to manage multi-version software implementation
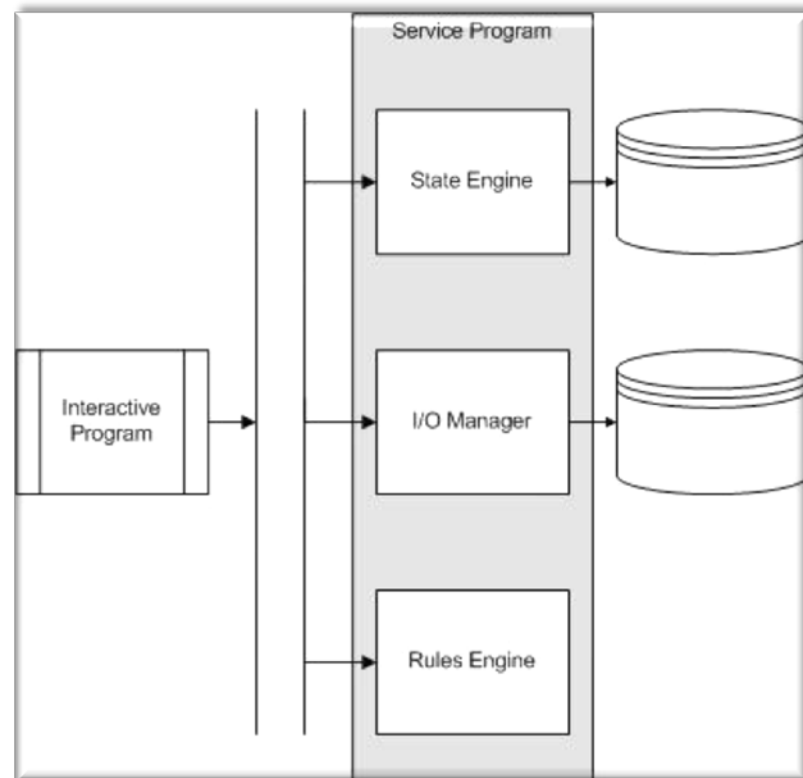
```
%FUNCTION(DTW_DIRECTCALL)
    Get_Menu ( IN CHAR(10) USR,CHAR(10) MNU,
               INOUT DTWTABLE MnuTable )
               { %exec {/QSYS.LIB/CGIPX010.LIB/WEB700RP.PGM %}
```

- Define columns for HTML Table

```
ndRC = dtw_SetCols(ndTable:5)
ndRC = dtw_SetN(ndTable:'MNIMG1':1)
ndRC = dtw_SetN(ndTable:'MNITM1':2)
ndRC = dtw_SetN(ndTable:'MNIMG2':3)
ndRC = dtw_SetN(ndTable:'MNITM2':4)
ndRC = dtw_SetN(ndTable:'RECID':5)
ndRow = *ZEROS
```
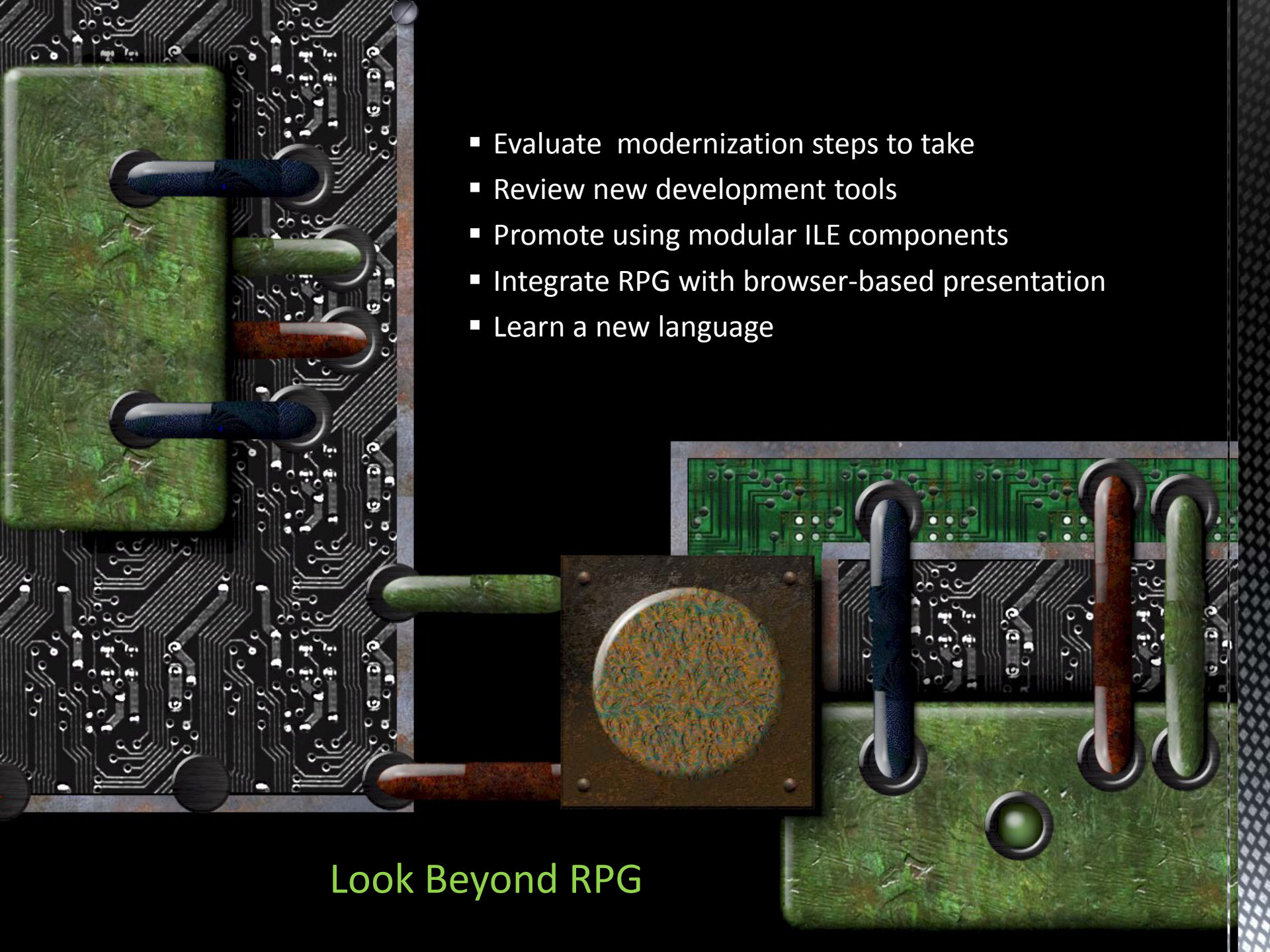
RPG and GCI

- Append row to HTML table

```
ndRC = dtwAppRow(ndTable:1)
ndRow += 1
ndRC =  dtw_SetV(ndTable:%trim(W$IMAG1):ndRow:1)
ndRC = dtw_SetV(ndTable:%trim(W$LINK1):ndRow:2)
ndRC = dtw_SetV(ndTable:%trim(W$IMAG2):ndRow:3)
ndRC = dtw_SetV(ndTable:%trim(W$LINK2):ndRow:4)
ndRC = dtw_SetV(ndTable:W$FMT:ndRow:5)
```

- Programming skills should keep evolving toward:
  - A newer, better RPG
  - RPG ILE as a component of other languages (*plural*)
  - Most RPG developers already know several languages:
    - CL
    - DDS
    - SQL
    - REXX
- Learn GUI presentation
  - HTML
  - CSS
  - JavaScript
  - Ajax
  - PHP

RPG Will Remain Viable as long as IBM offers i OS

- Evaluate modernization steps to take
- Review new development tools
- Promote using modular ILE components
- Integrate RPG with browser-based presentation
- Learn a new language

Look Beyond RPG