When & Where to Use Knowledgeware, Generative Scripts & VB Tools

Brian Prasad¹, Robert Garrison²
Jeff Rogers¹ and Christian Isaacs²

¹Parker Hannifin / Parker Aerospace
²Rand Worldwide / Rand Professional Services
Presentation Topics

• Parker Actuator Configurator Redux:
  • Creation & implementation of a generic KBE application that supports *engineer2configure for all possible actuator families*.
  • Rationale for using Knowledgeware tools, scripting and advanced KBE concepts to derive parts from specifications.

• Project Linchpin – The User Interface:
  • Leveraging the dynamic nature of the core system architecture.
  • Rationale of using Visual Studio .Net with Knowledgeware.

• Project Challenges:
  • A unified system architecture definition, case-based UI coding, development & deployment throughout.

• Lessons Learned
A Typical New Product Design & Development Scenario
What’s wrong with this?

- Knowledge is fragmented
- Subject matter experts (SME) often scarce and busy
- Less uniformity and consistency
- Time-intensive, manpower dependent
- When people retire, information is lost
- Often design is done via trial and error—case-based reasoning
Parker Configurator: Knowledge-centric approach
A CATIA V5 implementation

• System Architecture
  • JustOne system model and a common tree structure for several applications

• Generative Rule Bodies
  • Rule bodies create more rules dynamically on the tree; asleep until awaken (CATGScripts)
  • Retrieve templates; no generative geometry (Knowledgeware)

• Internal Linking
  • Two generalized automation methods to pass/exchange information intrapart & interpart (CATScripts)
### Specs Definitions (Excel Inputs)

<table>
<thead>
<tr>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Constraint Name</td>
<td>Type</td>
<td>Value (or Constraint Orient)</td>
<td>First Product</td>
<td>First Publication</td>
<td>Second Product</td>
<td>Second Publication</td>
<td>Compute</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CY238P Axial Coincidence</td>
<td>Coincidence</td>
<td>Cat/Cat/Diagonal Cylinder</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CY238E Axial Coincidence</td>
<td>Coincidence</td>
<td>Cat/Cat/Diagonal Cylinder</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CY238E Transverse Parallel</td>
<td>Parallel</td>
<td>Cat/Cat/Diagonal Cylinder</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CY238G Contact</td>
<td>Coincidence</td>
<td>Cat/Cat/Diagonal Cylinder</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>CY238G Lock/Nut Axial Coincidence</td>
<td>Coincidence</td>
<td>Cat/Cat/Diagonal Cylinder</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CY238G Lock/Nut Transverse Parallel</td>
<td>Parallel</td>
<td>Cat/Cat/Diagonal Cylinder</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CY238G Contact</td>
<td>Coincidence</td>
<td>Cat/Cat/Diagonal Cylinder</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CY238G Lock/Nut Contact</td>
<td>Contact</td>
<td>N/A</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CY238G Axial Coincidence</td>
<td>Coincidence</td>
<td>Cat/Cat/Diagonal Cylinder</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Cylinder_AxialLine</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Nom Supply Pressure (psi)</td>
<td></td>
<td>41000.001</td>
<td>21000.001</td>
<td>4057.001</td>
<td>SH</td>
<td>160</td>
<td>600000.001</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Nom PTank Pressure (psi)</td>
<td></td>
<td>175.001</td>
<td>150.001</td>
<td>600.001</td>
<td>SH</td>
<td>160</td>
<td>600000.001</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Nom Stank Pressure (psi)</td>
<td></td>
<td>100.001</td>
<td>150.001</td>
<td>600.001</td>
<td>SH</td>
<td>160</td>
<td>600000.001</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Limit Load Compression (lbf)</td>
<td></td>
<td>120060.001</td>
<td>180060.001</td>
<td>616000.001</td>
<td>160</td>
<td>600000.001</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Limit Load Tension (lbf)</td>
<td></td>
<td>112140.001</td>
<td>112140.001</td>
<td>616000.001</td>
<td>160</td>
<td>600000.001</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Ultimate Load Compression (lbf)</td>
<td></td>
<td>168210.001</td>
<td>168210.001</td>
<td>92401.001</td>
<td>160</td>
<td>600000.001</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Ultimate Load Tension (lbf)</td>
<td></td>
<td>800</td>
<td>800</td>
<td>400</td>
<td>160</td>
<td>600000.001</td>
<td>0.25</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Endurance Load Fatigue (lbf)</td>
<td></td>
<td>104400.001</td>
<td>104400.001</td>
<td>51394.001</td>
<td>160</td>
<td>600000.001</td>
<td>0.25</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Proof Supply Pressure (psi)</td>
<td></td>
<td>600.001</td>
<td>600.001</td>
<td>600.001</td>
<td>160</td>
<td>600000.001</td>
<td>0.25</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Proof Tank Pressure (psi)</td>
<td></td>
<td>1000.001</td>
<td>1000.001</td>
<td>10820.001</td>
<td>160</td>
<td>600000.001</td>
<td>0.25</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Impulse Supply Pressure (psi)</td>
<td></td>
<td>600.001</td>
<td>600.001</td>
<td>600.001</td>
<td>160</td>
<td>600000.001</td>
<td>0.25</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Impulse Tank Pressure (psi)</td>
<td></td>
<td>400.001</td>
<td>400.001</td>
<td>1275.001</td>
<td>160</td>
<td>600000.001</td>
<td>0.25</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Impulse Tank Pressure (psi)</td>
<td></td>
<td>600.001</td>
<td>600.001</td>
<td>2125.001</td>
<td>160</td>
<td>600000.001</td>
<td>0.25</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Impulse Tank Pressure (psi)</td>
<td></td>
<td>2000.001</td>
<td>2000.001</td>
<td>700.001</td>
<td>160</td>
<td>600000.001</td>
<td>0.25</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Impulse Load Cycles Fatigue</td>
<td></td>
<td>1000.001</td>
<td>1000.001</td>
<td>5000.001</td>
<td>160</td>
<td>600000.001</td>
<td>0.25</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Stroke Nominal (in)</td>
<td></td>
<td>9.487</td>
<td>9.487</td>
<td>8.592</td>
<td>160</td>
<td>500000.001</td>
<td>0.25</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Retract Length (in)</td>
<td></td>
<td>24</td>
<td>34.957</td>
<td>33.394</td>
<td>160</td>
<td>500000.001</td>
<td>0.25</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Bearing Friction Coeff</td>
<td></td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>160</td>
<td>500000.001</td>
<td>0.25</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Bearing Friction Coeff</td>
<td></td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>160</td>
<td>500000.001</td>
<td>0.25</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Bearing Friction Coeff</td>
<td></td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>160</td>
<td>500000.001</td>
<td>0.25</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The table contains various specifications and their values, organized into categories such as pressure, load, and other mechanical properties.
Achieving a Generic Product Configurator
Merits of modular process

- Product-Independent
  - Generative Architecture
  - Generic Systematization
- Part-Independent
  - Reusable Templates
  - Pattern Decomposition
- Tool-Independent
  - General-purpose parameters Interchange Methods
Shifting Focus to the User...

• CATIA Knowledgeware:
  • Has a wealth of tools.
  • Spartan user interface.
  • Executing the Actuator Configurator:
    • Modular & flexible, but required numerous operations to be executed manually and in a specific sequence.

• Requirements for a good user experience:
  • Make it easy to use effectively.
  • Make the experience of using the application enjoyable.
  • Failsafe the architecture to eliminate user mistakes.

• Parker Configurator--UI Needs
  • JustOne Interface and seamless dynamic interaction with the tree, Knowledgeware parameters and generative rules (no hard coding).
...While Leveraging the Original Investment

Modular & Flexible Approach...

Yields Wider Applicability
Development of a User Experience (UX)

A classic software development Triad – all components of which may not be found in a typical design or manufacturing plant.

Development investment should focus on *modularity* and *portability*.

*Industrial Designers*

*Graphic Designers*

*Interaction Designers*

*Behavior*

*Form*

*Content*

*Knowledge Engineers*
UX Development Tools for use with CATIA KBE

- **Visual Basic for Applications (VBA):**
  - Good for smaller internal deployments.
  - Does not allow development of processes that run outside of CATIA.

- **Visual Studio 6:**
  - Allows development of application processes outside of CATIA.
  - Architecture is old.
    - Code is unmanaged.
    - UX design tools are not modern.
UX Development Tools for use with CATIA KBE

• Visual Studio .NET:
  • VS2003 – 1.0 Framework
    • Suitable, but development tools were already being deprecated.
  • VS2005 – 2.0 Framework
    • Targeted development environment for this project:
      • *Out-process capable.*
      • *Exhaustive component libraries for form design.*
      • *Robust development environment.*
  • VS2008 – 3.0 / 3.5 Framework
    • Not on the market at the time of this project.
Actuator Configurator UX Development

• Leverage a modular architecture and case-based coding methodology (~8,000 lines) for UI in order to give user a better experience.
  • Provide hooks to the existing triggers:

• Facilitate template access & table use:
Actuator Configurator UX Development

• Flow with Run Time operations & hook to Reporting:

- UX coding approach:
  - Modular architecture and a case-based coding methodology. Frequent dynamic Interactions with tree.
  - Tools and forms reusable for future projects.
  - Limited to manipulating already extant KBE features in CATIA.
With minimal effort, this form & its underlying code can be adapted to any KBE-based equipment (SmartParts) deployment built in CATIA V5.
Actuator Configurator UX Development

• Enterprise challenges:
  • CATIA Knowledgeware is built on a foundation of simple syntax that most engineers can master.
    • Facilitates development of small or large KBE tools.
  • Many companies have very few or no software engineers conversant with CATIA Automation methods.
  • Simple VB Scripts built with the assistance of macro-recording might be the limit of Automation capability.
Actuator Configurator UX Development

- Modular and still ~ 8,000 lines of code?
- Code development breakdown for this project:

54% Hand-Coded
46% Windows-Generated
Actuator Configurator UX Development

• Code reuse breakdown for this project:

- 90% Reusable Code
- 10% Non-Reusable Code
Actuator Configurator UX Development

Classes & Modules providing 100% code reuse for similar deployments

Classes & Modules @ 90% reusability

Modular project structure promotes code reuse in similar future projects
Actuator Configurator UX Development

• Modular code by modular developers:
  • Content was already present.
  • Ran independent of UI.
    • Parker designed the KBE tools used.
    • Tools were embedded in CATIA V5.
  • Form was created to support access of Content.
  • Behavior was created to interface Form with Content.
    • Developer: Christian Isaacs (Rand Worldwide)
  • ~ 6 month hiatus ensued between first release and commencement of final release.
Actuator Configurator UX Development

• Modular code by modular developers:
  • Project handed off to second developer.
    • Minimal dialog between developers.
    • If done properly, code says it all.
  • Form was extended to encompass Content previously not included.
  • Behavior was extended to meet project changes.
    • Developer: Robert Garrison (Rand Worldwide)
  • Code hand-off to Parker for further reuse on this and other projects.
    • Internal Software Engineer received a half-day code review from second developer.
Actuator Configurator UX Development

• In retrospect:
  • Content is within reach of any user.
    • Developed & deployed solely in CATIA.
    • Exploits rich Knowledgeware toolset.
  • Form & Behavior are also easily within reach.
    • Many .NET Express Edition tools are free from Microsoft, so even small companies have access.
    • Skilled partners can develop the modular and extendable architecture that you then carry forward.
    • Training in Automation of CATIA is also readily available through a variety of sources.
      • Good option for projects of smaller scope & for proving out the benefits.
The benefits become clear, when the project rolls out to production.
Lessons Learned

• Maintaining a generative structure and dynamic creation of knowledge at run-time helps reusability and promotes generality of applications.

• Consider a similar structure for the User Interface when deploying Knowledge Systems.
  • CATIA’s KBE interface is not intuitive for users.
  • Too many interactions required by the user can result in non-use by those who would benefit most.

• When deploying complex systems, commit to an agile architecture for UI.
  • Use modern Development Environments & schemas:
    • .NET, XML, etc.
    • Design the User Experience in tandem with the system.
Lessons Learned

• Make the system flexible and portable.
  • Strive to develop projects that minimize or eliminate redundant coding.
  • Strive for generic classes that can be reused.
  • Resist the temptation to build KBE features with your UX.
    • Keep Contents separate from Form and Behavior.

• A well structured project presents few problems in expansion or adaptation.
Questions?

• For more information, contact:

Brian Prasad
Leader, Knowledge Engineering Team
Parker Aerospace, Control Systems Division
Irvine, California
bprasad@parker.com

Robert Garrison
Solutions Architect
Rand Worldwide, Rand Professional Services
Seattle, Washington
rgarrison@rand.com