COE 2008 Annual PLM Conference & TechniFair
April 27-30, 2008 • Walt Disney World Swan and Dolphin • Lake Buena Vista, Florida

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

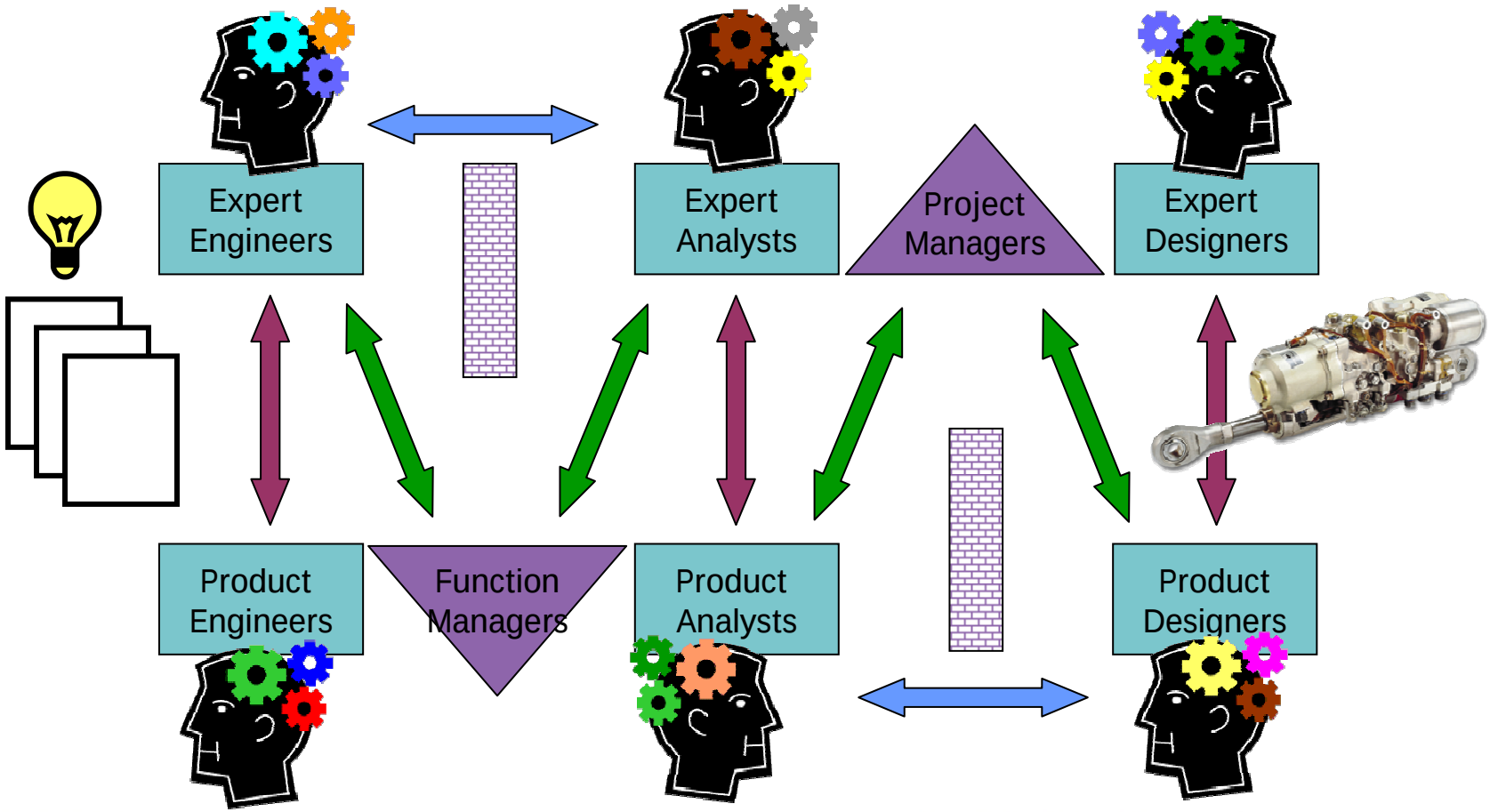
Brian Prasad[1], Robert Garrison[2]
Jeff Rogers[1] and Christian Isaacs[2]

[1]Parker Hannifin / Parker Aerospace
[2]Rand Worldwide / Rand Professional Services

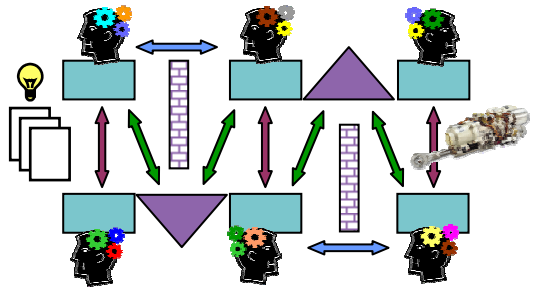
Presentation Topics

- 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

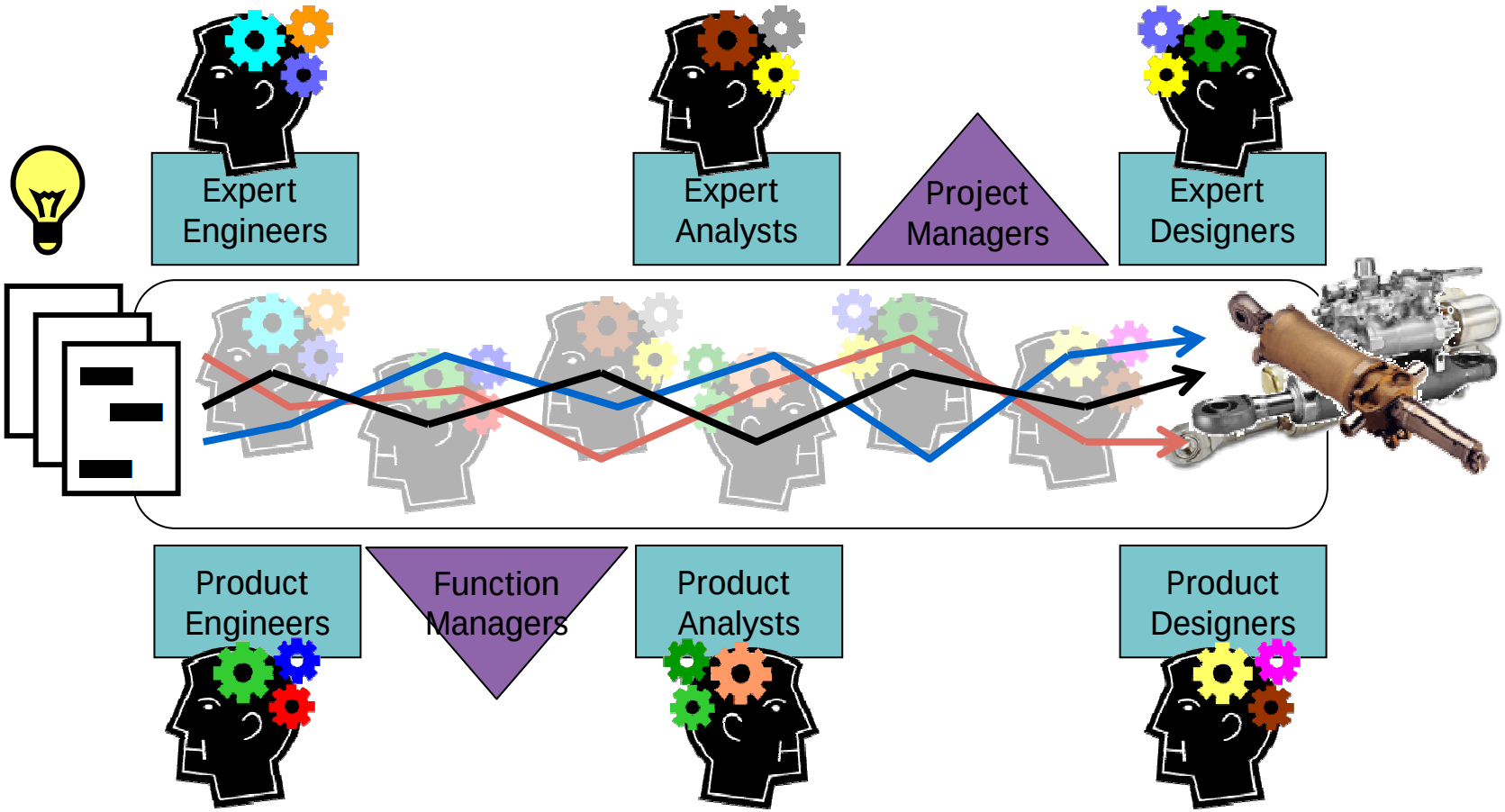


What's wrong with this?



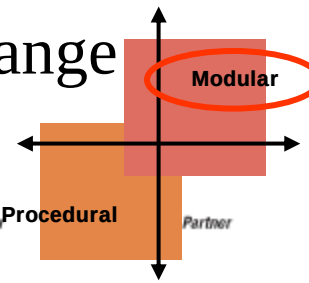
- Knowledge is fragmented
- Subject matter experts (SME) often scarce and busy
- Less uniformity and consistency
- Time-intensive, manpower dependent
- When people retire, information is lost
- Often design is done via trial and error—case-based reasoning

Parker Configurator: Knowledege-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)

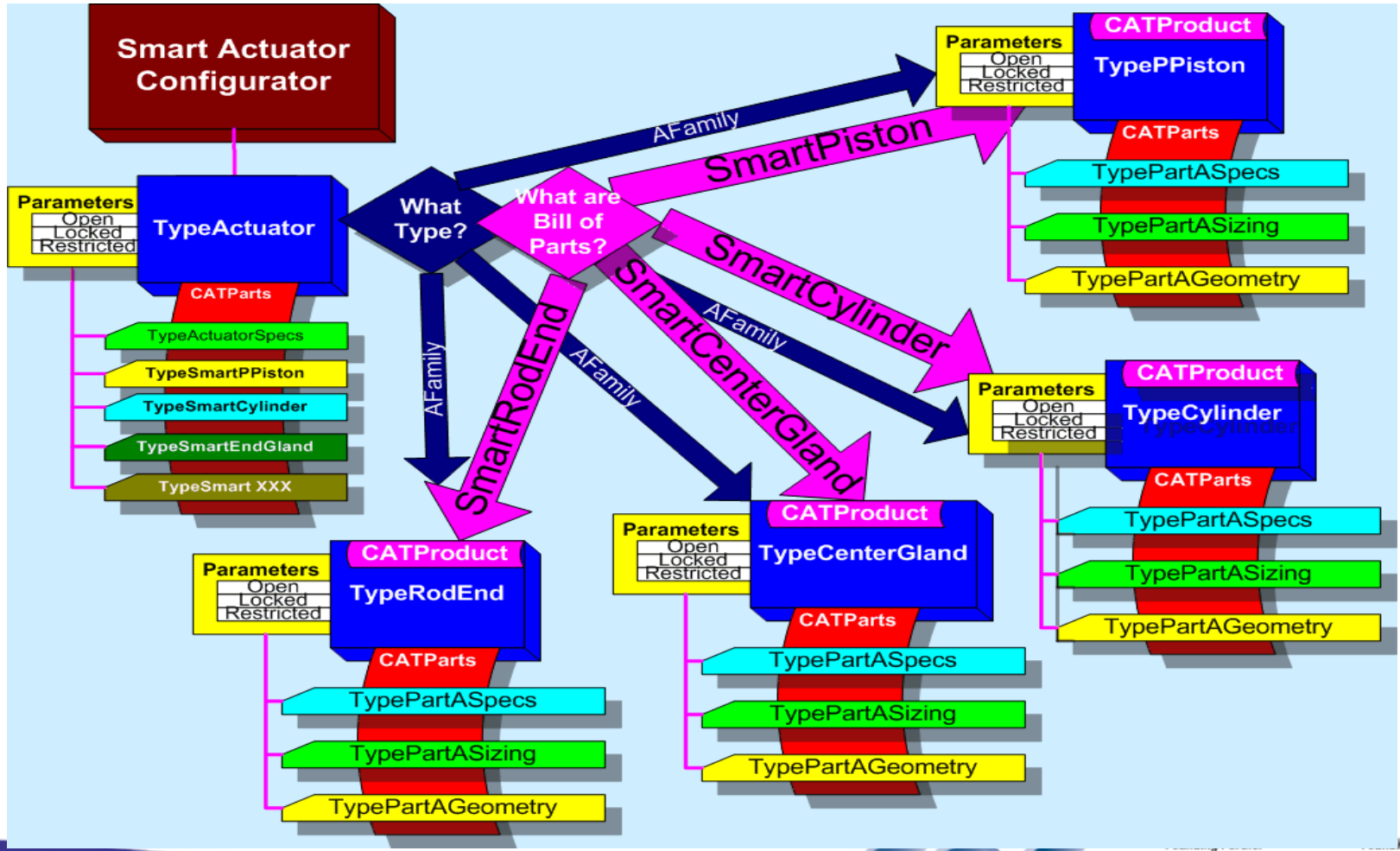


Specs Definitions (Excel Inputs)

	B	C	D	E	F	G	H	I	J
1	Constraint Name	Type	Value (in	Constraint Orienta	First Product	First Publication	Second Pro	Second Publication	Compute
2	CY2PP_Axial_Coincidence	Coincidence		CatCstOrientUndefine	Cylinder	Cylinder_AxisLine	PPiston	PPiston_AxisLine	Y
3	CY2REEG_Axial_Coincidence	Coincidence		CatCstOrientUndefine	Cylinder	Cylinder_AxisLine	REEndGland	REEG_AxisLine	Y
4	CY2REEG_Transverse_Parallel	Parallel		CatCstOrientSame	Cylinder	Cylinder_TransverseLine	REEndGland	REEG_TransverseLine	Y
5	CY2REEG_Contact	Coincidence		CatCstOrientOpposite	Cylinder	CylinderNutBoreThread_REEndGland_Contac	REEndGland	REEGMiddleRing_Cylinder_ContactPla	Y
6	CY2EGLockNut_Axial_Coincidence	Coincidence		CatCstOrientUndefine	Cylinder	Cylinder_AxisLine	EGLockNut	EGLockNut_AxisLine	Y
7	CY2EGLockNut_Transverse_Parallel	Parallel		CatCstOrientSame	Cylinder	Cylinder_TransverseLine	EGLockNut	EGLockNut_TransverseLine	Y
8	CY2EGLockNut_Contact	Contact		N/A	Cylinder	Cylinder_EGLockNut_ContactFace	EGLockNut	EGLockNut_Cylinder_ContactFace	N
9	CY2CEEG_Axial_Coincidence	Coincidence		CatCstOrientUndefine	Cylinder	Cylinder_AxisLine	CEEndGland	CEEG_AxisLine	Y
10	CY2CEEG_Transverse_Parallel	Parallel		CatCstOrientSame	Cylinder	Cylinder_TransverseLine	CEEndGland	CEEG_TransverseLine	Y

11	Nom_Supply_Pressure (psi)	4100.001	4100.001	4057.001	SH	160	600000.001	0	3		
12	Nom_PTank_Pressure (psi)	175.001	150.001	600.001	SH	160	600000.001	0	3		
13	Nom_STank_Pressure (psi)	100.001	150.001	600.001		160	600000.001	0	3		
14	Limit_Load_Compression (lbf)	120060.001	180060.001	61600.001		160	600000.001	0	3		
15	Limit_Load_Tension (lbf)	112140.001	112140.001	61600.001	G	160	50000.001	0	3	5c4	4a0
16	Ultimate_Load_Compression (l	180090.001	180090.001	92401.001	RMS	160	600000.001	0	2		
17	Ultimate_Load_Tension (lbf)	168210.001	168210.001	92401.001		160	600000.001	0.25	3		
18	Ultimate_Load_Side (lbf)	800	800	400		160	50000.001	0.25	2.5		
19	Endurance_Load_Fatigue (lbf)	104400.001	104400.001	51334.001	G	160	50000.001	0.25	2.5		
20	Proof_Supply_Pressure (psi)	6000.001	6000.001	6500.001		160	50000.001	0.25	2.5		
21	Burst_Supply_Pressure (psi)	10000.001	10000.001	10820.001	G	160	50000.001	0.25	2.5		
22	Impulse_Supply_Pressure (psi)	6000.001	6000.001	6100.001	G	160	50000.001	0.25	2.5		
23	Proof_Tank_Pressure (psi)	4000.001	4000.001	1275.001	G	160	50000.001	0.25	2.5		
24	Burst_Tank_Pressure (psi)	6000.001	6000.001	2125.001		160	50000.001	0.25	2.5		
25	Impulse_Tank_Pressure (psi)	2000.001	2000.001	700.001							
26	Impulse_Load_Cycles_Fatigue	1000.001	1000.001	5000.001							
27	Stroke_Nominal (in)	9.487	9.487	8.592							
28	Retract_Length (in)	24	34.957	33.394							
29	Bearing_Friction_Coeff_Proof	0.15	0.15	0.15	G	160	50000.001	0.25	3		
30	Bearing_Friction_Coeff_Burst	0.2	0.2	0.2							
31	Bearing_Friction_Coeff_Fatigu	0.1	0.1	0.1							

Achieving a Generic Product Configurator



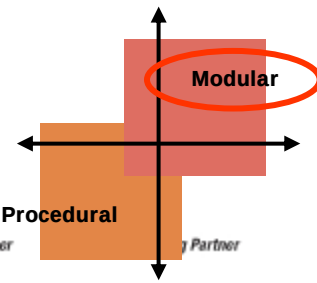
Merits of modular process

- Product-Independent
  - Generative Architecture
  - Generic Systematization
- Part-Independent
  - Reusable Templates
  - Pattern Decomposition
- Tool-Independent
  - General-purpose parameters Interchange Methods

Product Definition Configurator

Product Solution Configurator

SmartPart Configurator

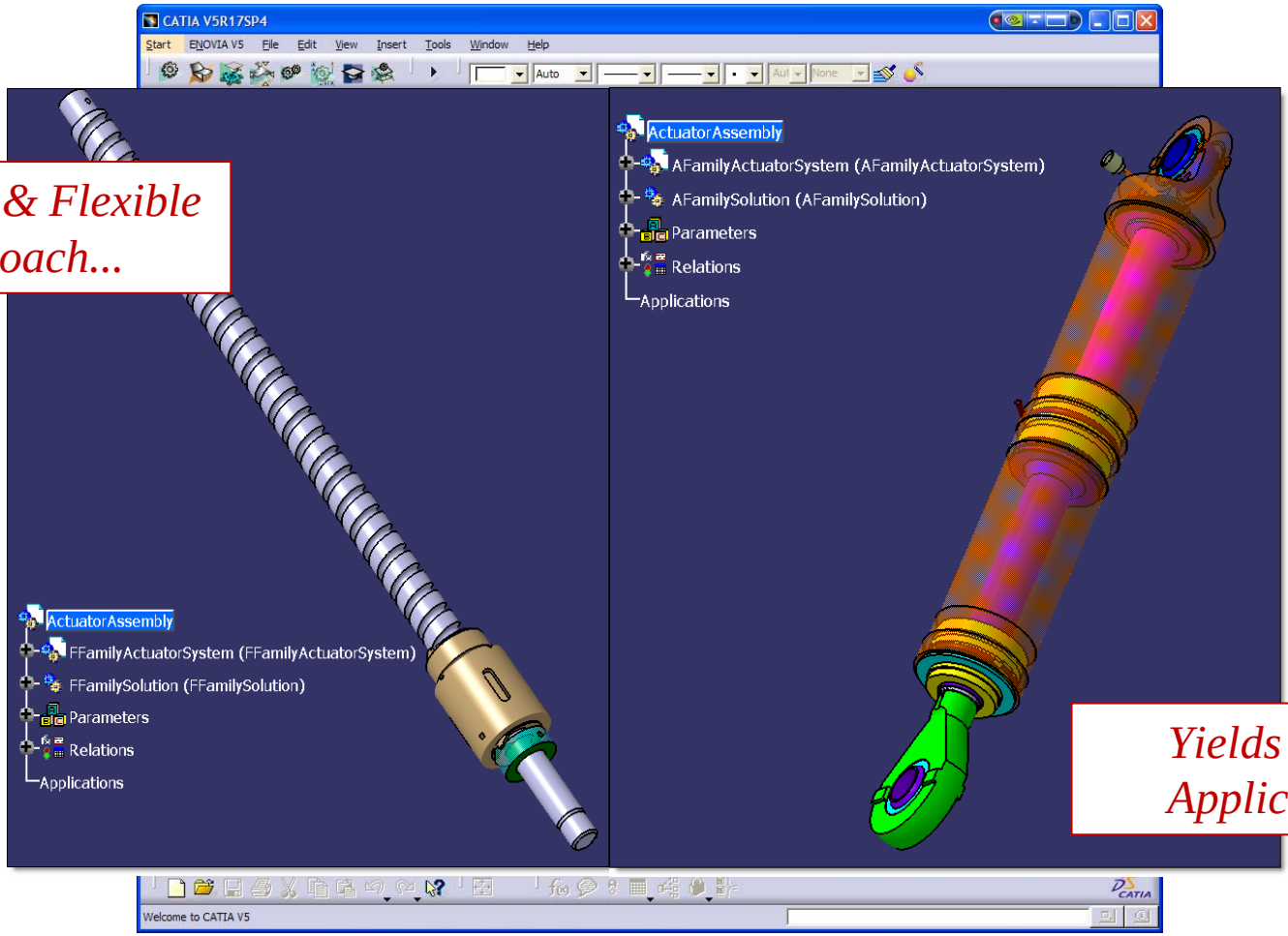


Shifting Focus to the User...

- 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...

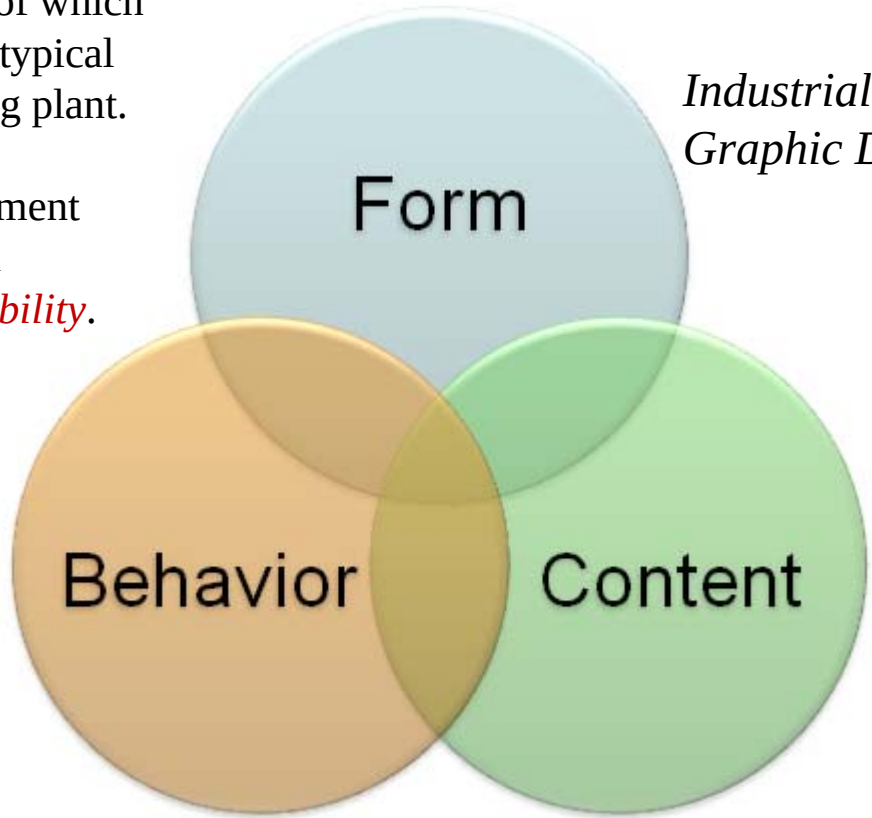


Yields Wider Applicability

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

Knowledge Engineers

UX Development Tools for use with CATIA KBE

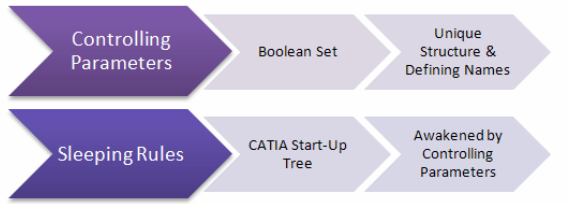
- Visual Basic for Applications (VBA):
  - Good for smaller internal deployments.
  - Does not allow development of processes that run outside of CATIA.
- Visual Studio 6:
  - Allows development of application processes outside of CATIA.
  - Architecture is old.
    - Code is unmanaged.
    - UX design tools are not modern.

UX Development Tools for use with CATIA KBE

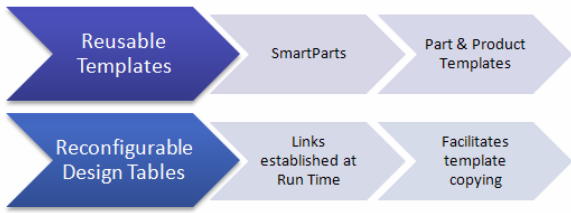
- Visual Studio .NET:
  - VS2003 – 1.0 Framework
    - Suitable, but development tools were already being deprecated.
  - VS2005 – 2.0 Framework
    - Targeted development environment for this project:
      - *Out-process capable.*
      - *Exhaustive component libraries for form design.*
      - *Robust 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:

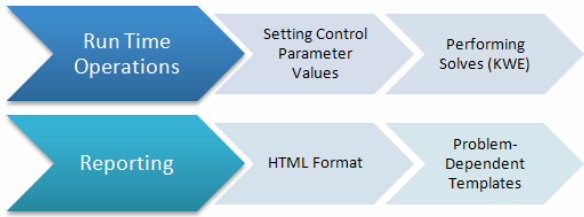


  - Facilitate template access & table use:



Actuator Configurator UX Development

- Flow with Run Time operations & hook to Reporting:

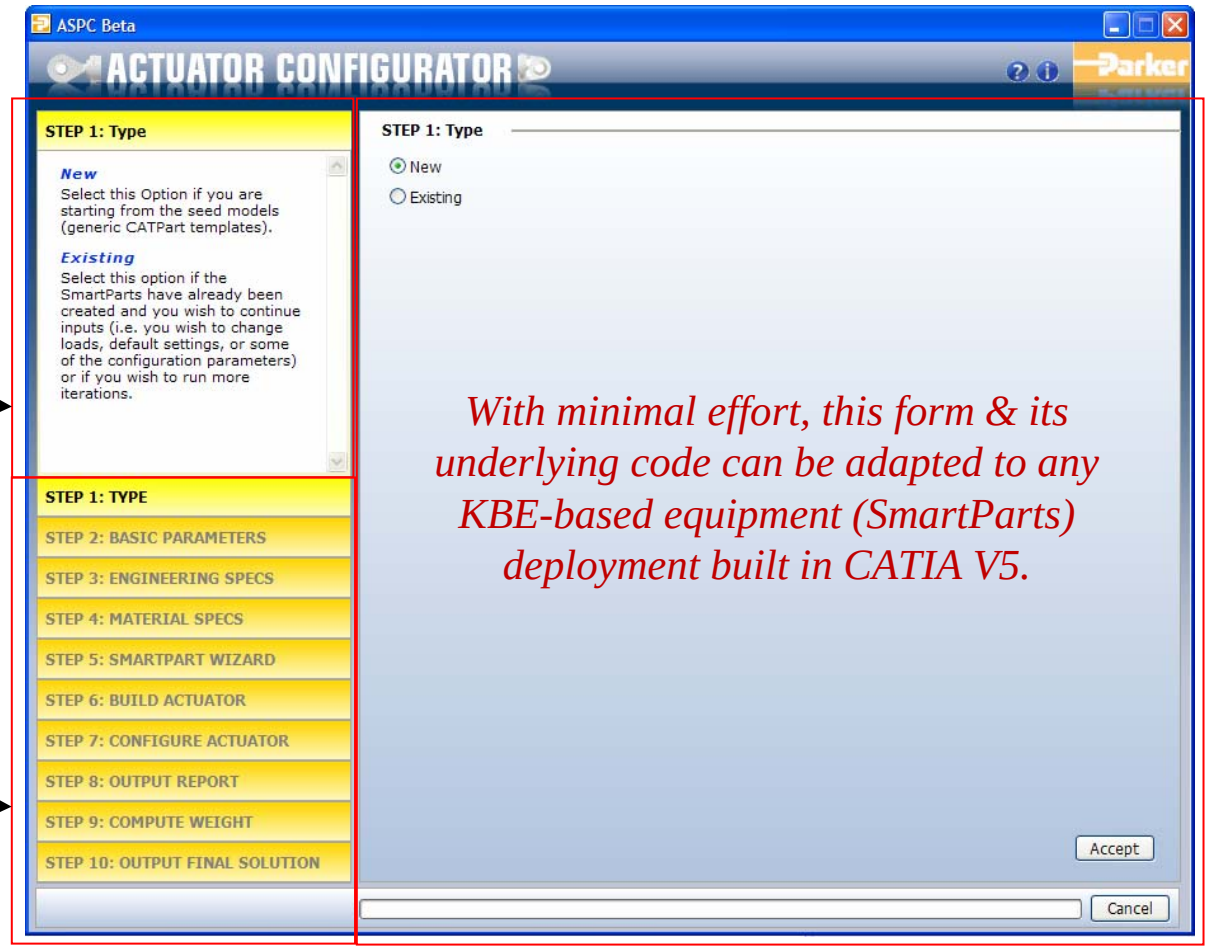


- UX coding approach:
  - Modular architecture and a case-based coding methodology. Frequent dynamic Interactions with tree.
  - Tools and forms reusable for future projects.
  - Limited to manipulating already extant KBE features in CATIA.

Actuator Configurator UX Development

User Help

Feedback on Progress



With minimal effort, this form & its underlying code can be adapted to any KBE-based equipment (SmartParts) deployment built in CATIA V5.

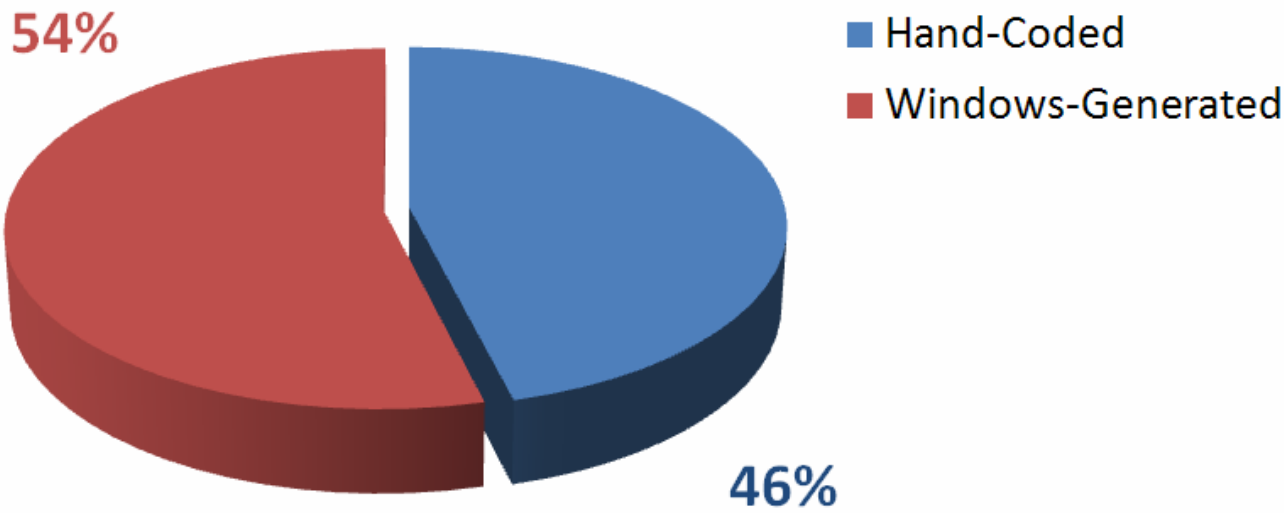
User Interaction & Program Output

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.

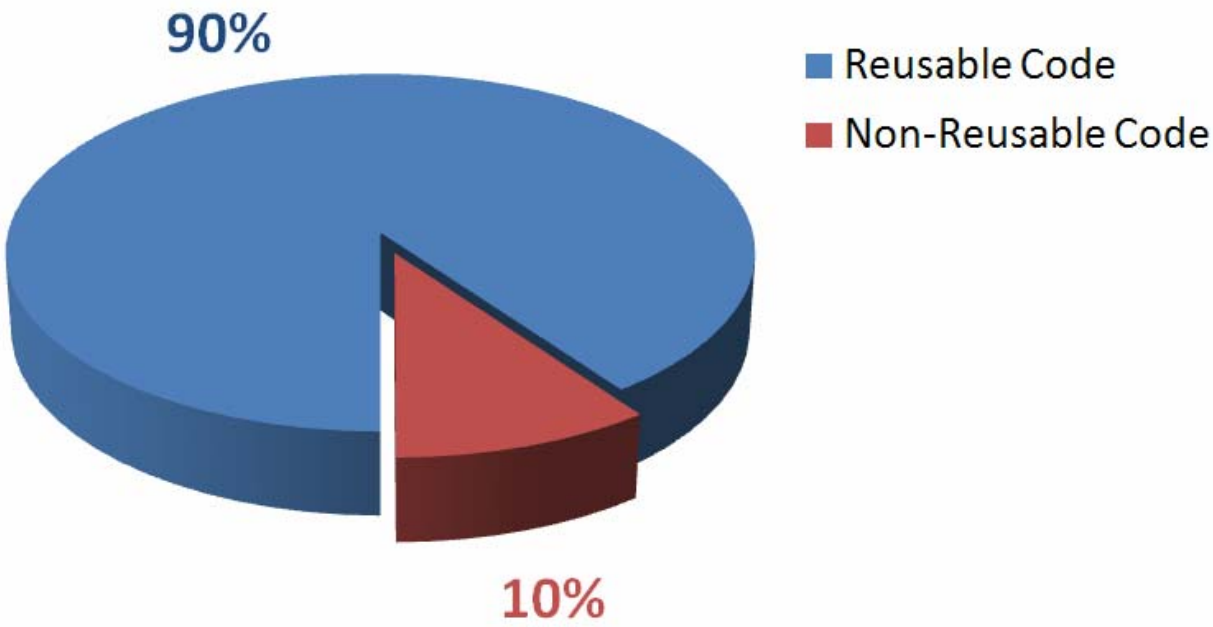
Actuator Configurator UX Development

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

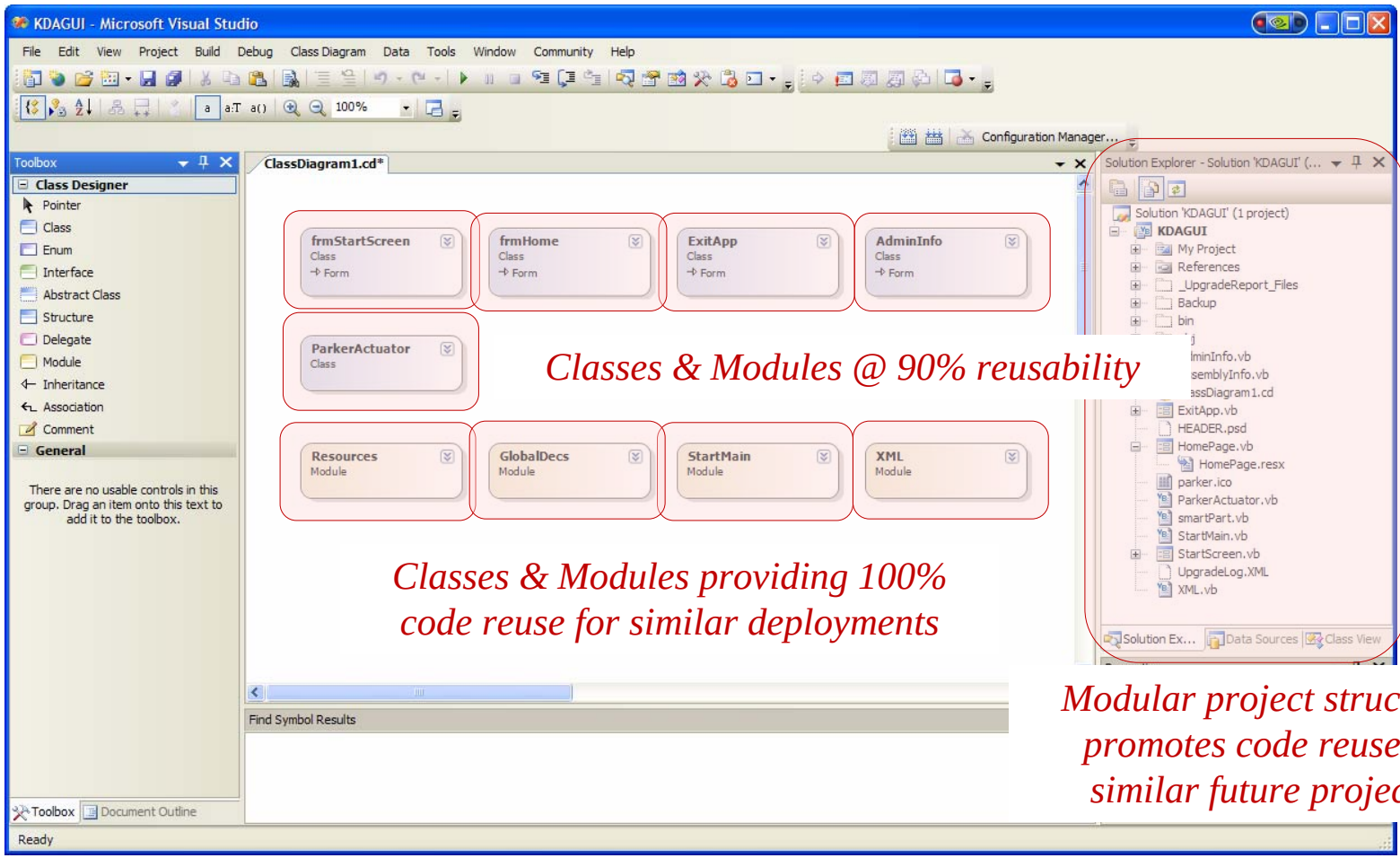


Actuator Configurator UX Development

- Code reuse breakdown for this project:

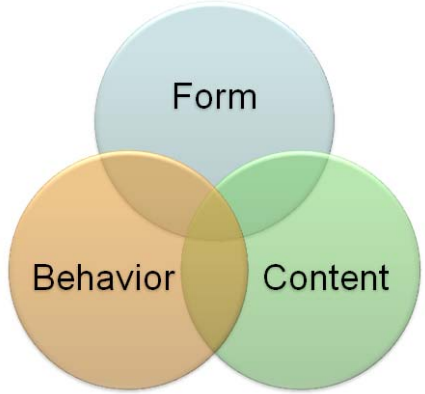


Actuator Configurator UX Development



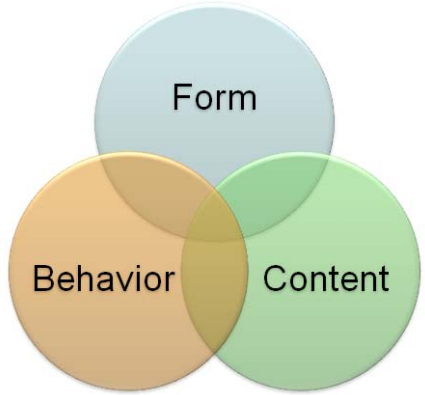
Modular project structure promotes code reuse in similar future projects

Actuator Configurator UX Development



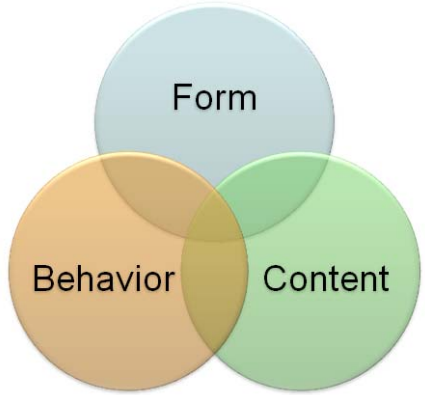
- Modular code by modular developers:
  - Content was already present.
  - Ran independent of UI.
    - Parker designed the KBE tools used.
    - Tools were embedded in CATIA V5.
  - Form was created to support access of Content.
  - Behavior was created to interface Form with Content.
    - Developer: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

Actuator Configurator UX Development



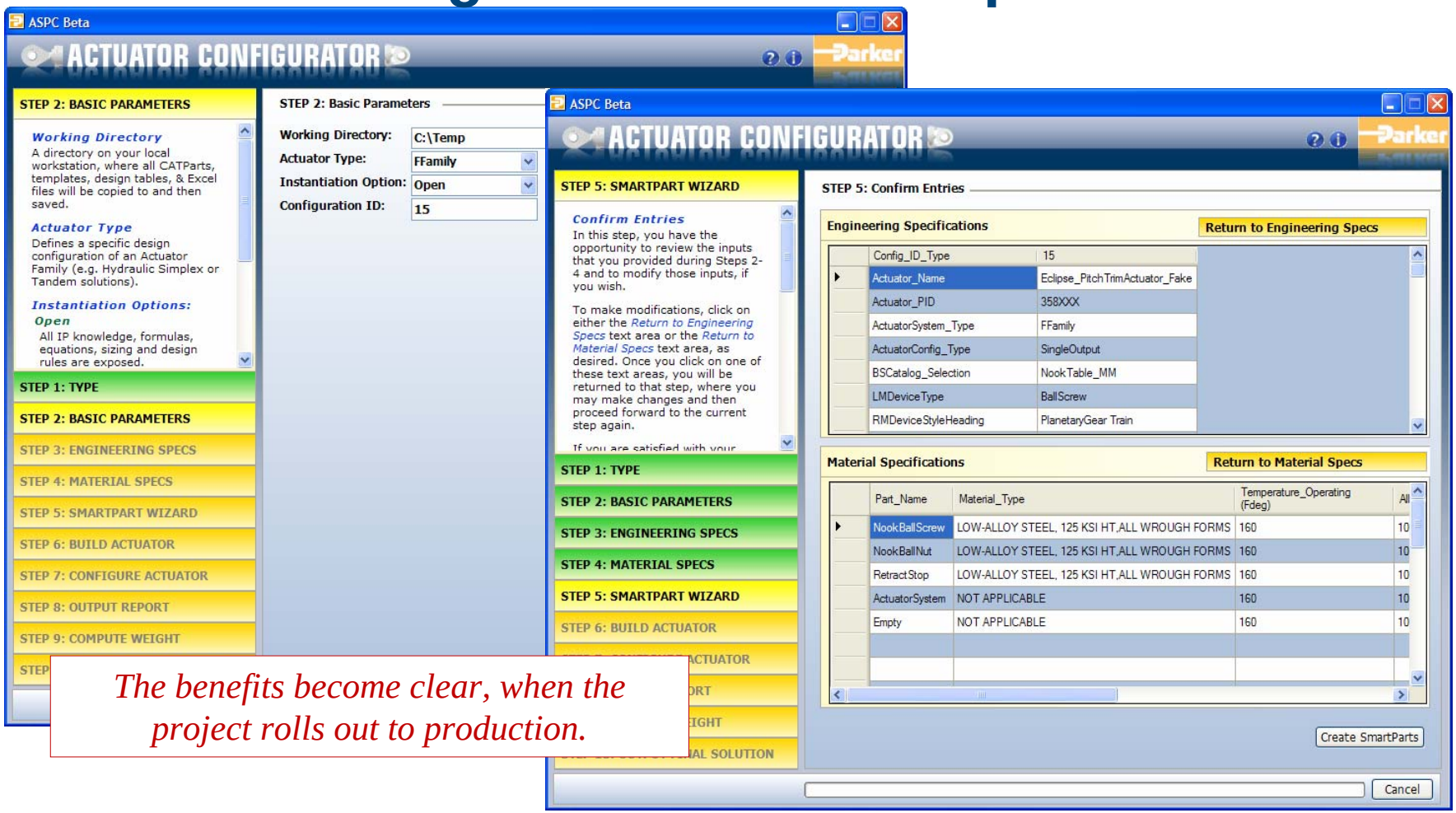
- Modular code by modular developers:
  - Project handed off to second developer.
    - Minimal dialog between developers.
    - If done properly, code says it all.
  - Form was extended to encompass Content previously not included.
  - Behavior was extended to meet project changes.
    - Developer: *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.*

Actuator Configurator UX Development



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