JOHN DEERE

SOA
the Good, the Bad and the Ugly

Duane Clarkson
Information Systems Architecture, March 2005

Agenda

- John Deere - the company
- Our Integration history
- Integration standardization
- Considerations in moving to a SOA
- Web Services and MDA
- Application integration program behaviors

- Questions, and your feedback

Our History

1837 – John Deere develops the world's first commercially successful, self-scouring steel plow, opening the West to economic development

1918 – Modern tractor era begins (Waterloo Boy)

1937 – Sales reach $100 million in centenary year

1956 – Construction equipment business launched
International manufacturing begins

1958 – John Deere Credit Company launched

1963 – Lawn and grounds care business launched

1998 – Earnings reach $1 billion

JOHN DEERE

John Deere Today

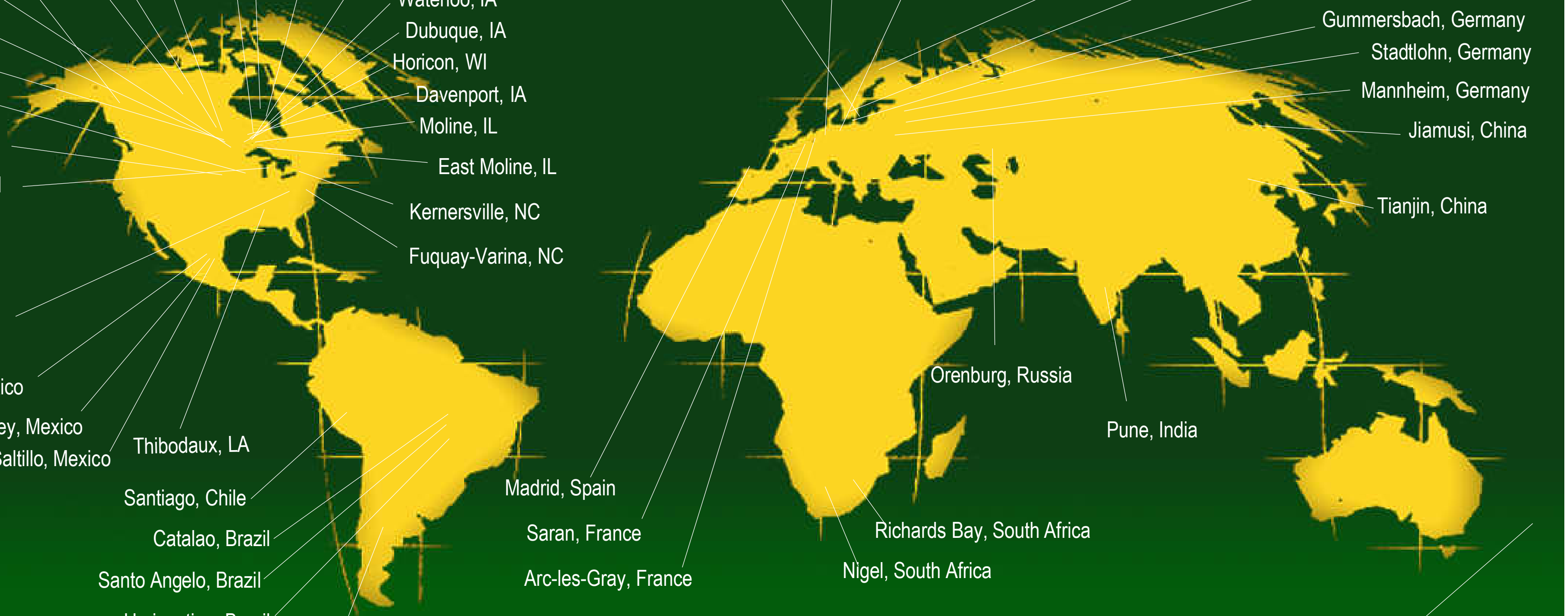
- The world’s leading producer of **Agricultural Equipment**

- A **Construction Equipment** industry leader and world’s premier producer of timber-harvesting equipment

- World leader in premium **Commercial & Consumer** turf-care equipment and work vehicles

JOHN DEERE

John Deere Today



Valley City, ND
Minneapolis, MN
Fargo, ND
Edmonton, Alberta
Woodstock, Ontario
Welland, Ontario
Langley, British Columbia
Klemme, IA
Rock Valley, IA
Waterloo, IA
Des Moines, IA
Dubuque, IA
Horicon, WI
Ottumwa, IA
Davenport, IA
Springfield, MO
Moline, IL
Coffeyville, KS
East Moline, IL
Greeneville, TN
Kernersville, NC
Fuquay-Varina, NC
Augusta, GA
Torreon, Mexico
Monterrey, Mexico
Thibodaux, LA
Saltillo, Mexico
Santiago, Chile
Catalao, Brazil
Santo Angelo, Brazil
Horizontina, Brazil
Rosario, Argentina

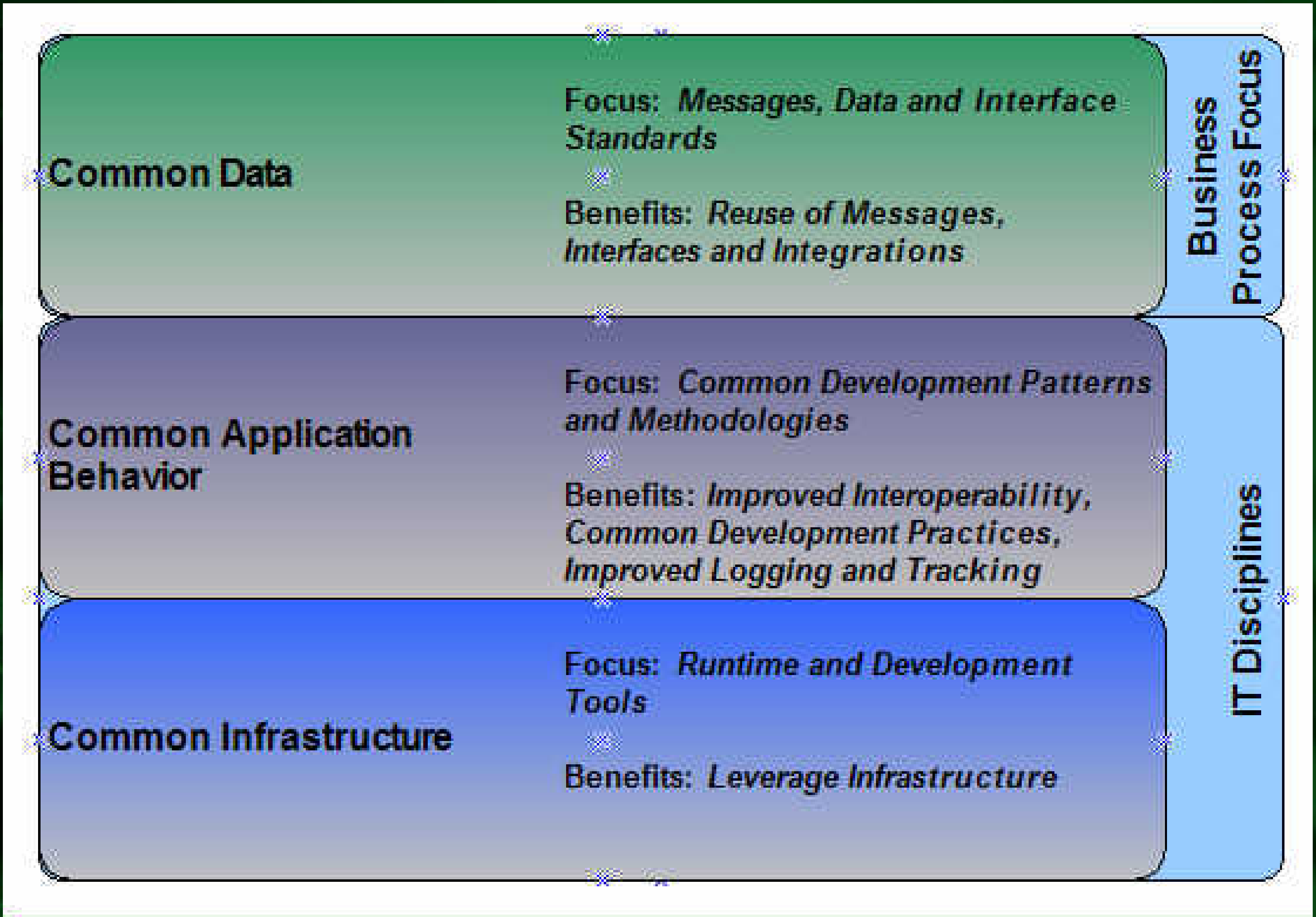
Zweibruecken, Germany
Bruchsal, Germany
Joensuu, Finland
Poznan, Poland
Enschede, Netherlands
Horst, Netherlands
Gummersbach, Germany
Stadtlohn, Germany
Mannheim, Germany
Jiamusi, China
Tianjin, China
Orenburg, Russia
Pune, India
Madrid, Spain
Saran, France
Arc-les-Gray, France
Richards Bay, South Africa
Nigel, South Africa
Tokoroa, New Zealand

*

Integration history

- Integrating applications since 1985
- Deere was the first commercial member of OMG in 1991
- Many environments
- Many integration architectures
- Narrowing the road
- Move toward current integration standards
- Adding more program behavior standards

Integration Standardization Focus



Common Data

Focus: *Messages, Data and Interface Standards*

Benefits: *Reuse of Messages, Interfaces and Integrations*

Common Application Behavior

Focus: *Common Development Patterns and Methodologies*

Benefits: *Improved Interoperability, Common Development Practices, Improved Logging and Tracking*

Common Infrastructure

Focus: *Runtime and Development Tools*

Benefits: *Leverage Infrastructure*

Moving to an SOA

- Need more than in the box
  - Procedures to bring it all together in a common approach
  - What types of integration patterns
  - Logging and tracking across components
  - Service and message naming
  - Security
  - Error handling
- Message design
  - Standards or not
  - How canonical do you go
- Not just a technical issue
  - Marketing the approach
  - Diverse organization, many development groups
  - Many development approaches in use
  - Agreement on best model

Applying MDA Concepts

- MOF Repository
  - Business message
  - Standard application headers
  - Reuse element and structure definitions
  - XML element validation
  - Usage information
- Messages generated from schemas
  - Repository generates schema
  - Schema brought into development environment
  - Java
  - Integration software

Web Services considerations

- Found needed to align to a set of standards, not just one
  - New to us, or not ready:
  - The tooling, which ones?
- The standards for Basic Profile 1.0:
  - SOAP 1.1
  - XML 1.0
  - HTTP 1.1
  - WSDL 1.1
  - XML Schemas
  - UDDI Version 2 XML Schema
  - TLS Version 1.0
  - SSL 3.0
  - X.509 Public Key Infrastructure
- Security, It’s a big deal
  - WS-Security
  - SAML

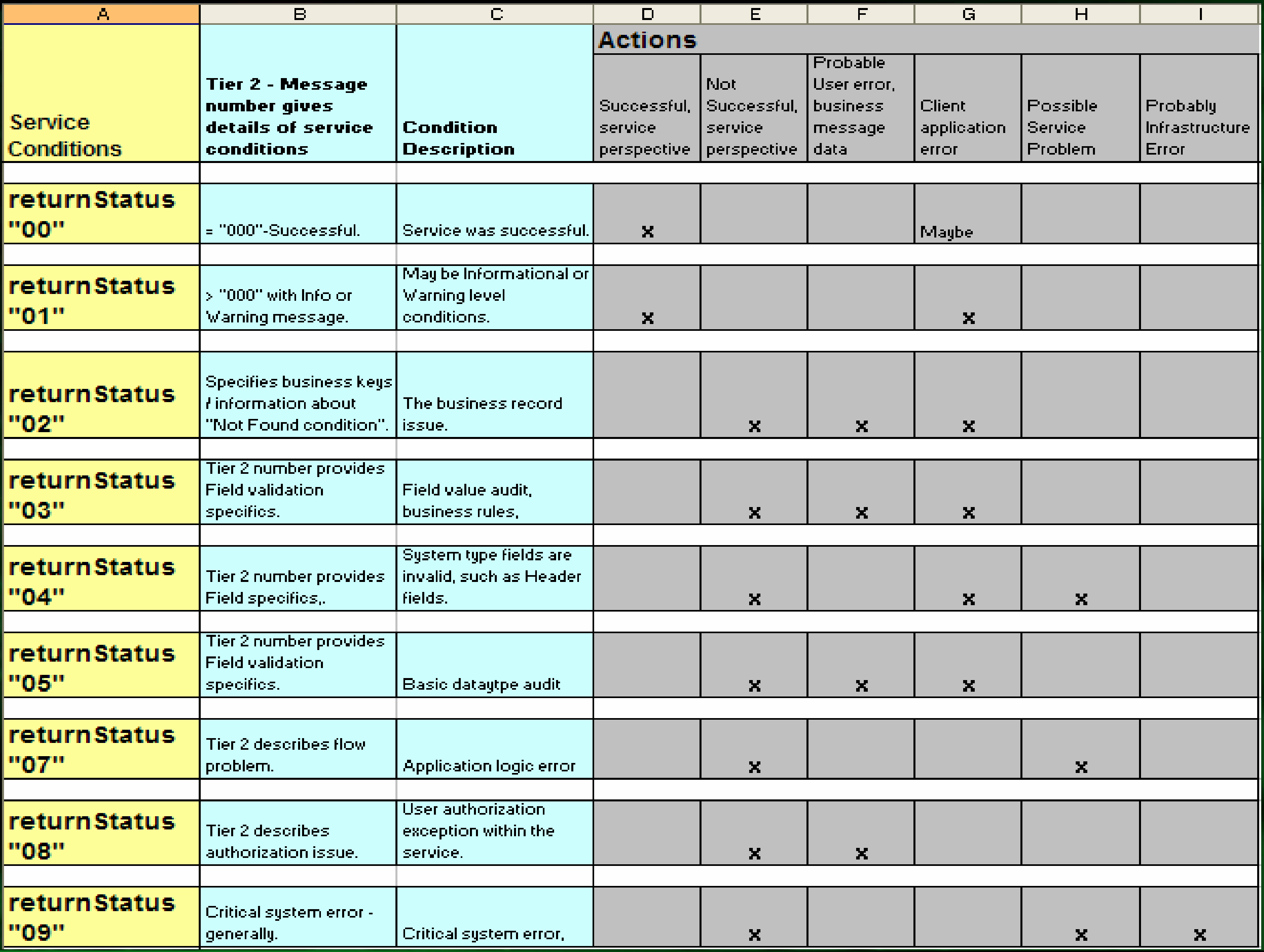
The program behaviors are important

- Open, loosely coupled, flexible integrations
- More to it than just adding XML, or Web Services
- Standard application headers
- Common code to create and process
- Common application design patterns
- Common logging and tracking component

Loosely coupling the applications

- Standardized messages
- General return status
  - Worked OK, worked with warnings
  - Message and Field errors
  - Integration errors,
  - Authorization and severe errors
- What errors to make public
  - A standard for the public interface
  - Fidelity of the messages
  - Opportunity for standardization
- To hide
  - Back end error codes
  - Back end details
- Standard integration patterns
  - Synchronous
  - Asynchronous
  - Asynchronous with reply
  - application headers
- Common code to create and process
- Common application design patterns

A Return Status



A	B	C	D	E	F	G	H	I
			Actions					
Service Conditions	**Tier 2 - Message number gives details of service conditions**	**Condition Description**	Successful, service perspective	Not Successful, service perspective	Probable User error, business message data	Client application error	Possible Service Problem	Probably Infrastructure Error
returnStatus "00"	= "000"-Successful.	Service was successful.	x			Maybe		
returnStatus "01"	> "000" with Info or Warning message.	May be Informational or Warning level conditions.	x			x		
returnStatus "02"	Specifies business keys / information about "Not Found condition".	The business record issue.		x	x	x		
returnStatus "03"	Tier 2 number provides Field validation specifics.	Field value audit, business rules,		x	x	x		
returnStatus "04"	Tier 2 number provides Field specifics,.	System type fields are invalid, such as Header fields.		x		x	x	
returnStatus "05"	Tier 2 number provides Field validation specifics.	Basic dataytpe audit		x	x	x		
returnStatus "07"	Tier 2 describes flow problem.	Application logic error		x			x	
returnStatus "08"	Tier 2 describes authorization issue.	User authorization exception within the service.		x	x			
returnStatus "09"	Critical system error - generally.	Critical system error,		x			x	x

Return Condition Specification

- *returnStatus (tier 1)*
- *tier2Condition*
  - *messageNumber*
  - *nameSpace*
  - *messageText*
- *linkbackEndServiceDetails (tier 3)*
  - *programName*
  - *messageCode*
  - *messageText*

What has the experience been like?

- The Good
- The Bad
- And the Ugly

Questions

Nothing Runs Like a Deere™

JOHN DEERE