



# *Info*Trust

Group

The Reality of Reuse in S1000D

A Practical Guide to Understanding and  
Getting Reuse Done

Jeff Deskins

# \$1000D Guy

## Speaker

Jeff Deskins

- ❖ Irvine/Boulder/Phoenix
- ❖ Product Engineering
- ❖ Product Manager

## Special Thanks

Gary Mayer

- ❖ Boulder
- ❖ Product Engineering
- ❖ Chief Architect



# Topic Overview



Part 1: Content Reuse, What and Why?

Part 2: Reuse Problems

Part 3: Classify Reuse

Part 4: Reuse Solutions

Part 5: Conclusions

# What is Content Reuse?

## Fundamental Concepts of Reuse

- Traditionally, ad hoc reuse use parts of existing docs to develop new ones, but probably more through cut/paste than structured reuse
- Now, our focus is turning to developing reusable components, for reuse in new documents
- Traditionally, one of the more common things to reuse were graphics or other media, partially because they exist as “components” so that they are stand alone, and are not adapted for different uses



# What is Content Reuse?

- Now, actual text content reuse is becoming more formally defined and widespread
- Reuse can be large chunks (Tasks, DM's) or small chunks (IPC PNR data, EIN) or anything in between
- ATA manages whole documents
- S1000D manages all content in DM's and PM's organize DM's for presentation to end users (i.e.. documents)
- You can also present/organize DM's in different, job oriented ways

# Why Reuse Content?

We've been doing reuse for quite some time

- Manufacturing Automobiles
- Software

Reusing content provides the following advantages:

- **Increased consistency**
  - Consistency leads to higher quality content
  - Furthers regulatory compliance
- **Ensures that content is consistent wherever it is used**
  - Amount of new content an author has to create is reduced
- **Reduced development and maintenance costs**
  - Rather than writing all new content
  - or taking the time to find and copy content to be reused
  - Rapidly available through improved management facilities
  - Make an update once, in one place
- **Rapid reconfiguration**
  - Reusable content is modular content (small self-contained components that can be used in combination with other components)
  - Easily change the order of modules, include new modules, exclude existing modules

# Problems with Reuse

What's the leading cause of difficulties with reuse?

Why is it sometimes difficult to achieve?

- **Change**
  - If information never changed, reuse would be easy.
  - Ripple effect
- **Locating Reusable Resources**
- **Don't control content of reused components**

# Classify Reuse - Evolution of Creation

- **Cut and Paste**
- **Opportunistic**
  - An author makes a conscious decision to reuse content
  - Lack of a system, free form management of reuse
- **Planned**
  - Authoring aids are available
  - System management of reuse and changes are controlled

# Classify Reuse - Evolution of Creation

- **Within planned reuse are four options:**
  - Literal reuse - As-is
  - Transformed reuse - automatically reapplied when source changes
  - Constrained reuse - retains essential structure allowing restricted changes
  - Derivative reuse - reused content can be changed and traceability is in place

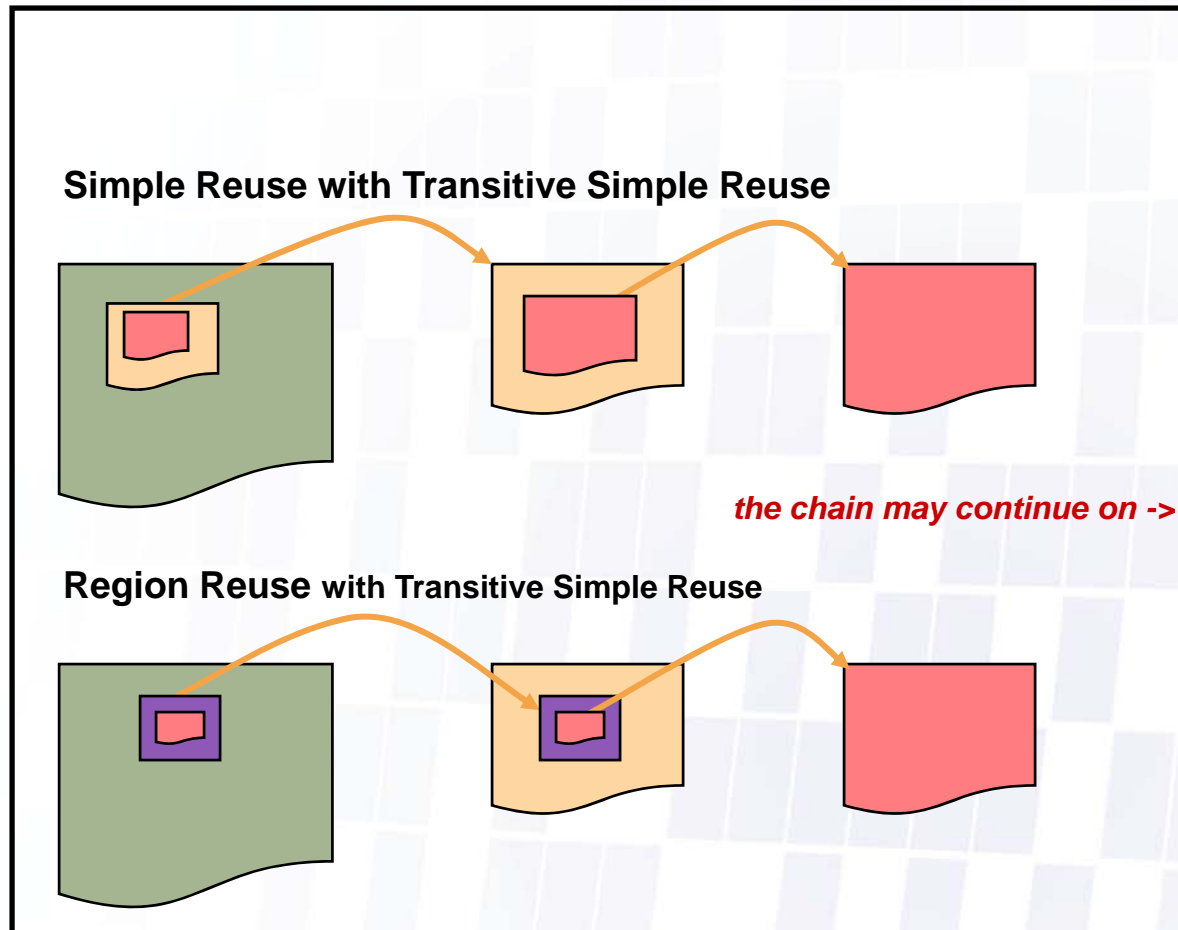
# Classify Reuse - Evolution of Delivery

- Phase 1: One document, one media
- Phase 2: Static Customized, multiple media
- Phase 3: Single Sourced, automatically processed for multiple media
- Phase 4: Dynamically Composed Content

# Classify Reuse - Basic Reuse Types

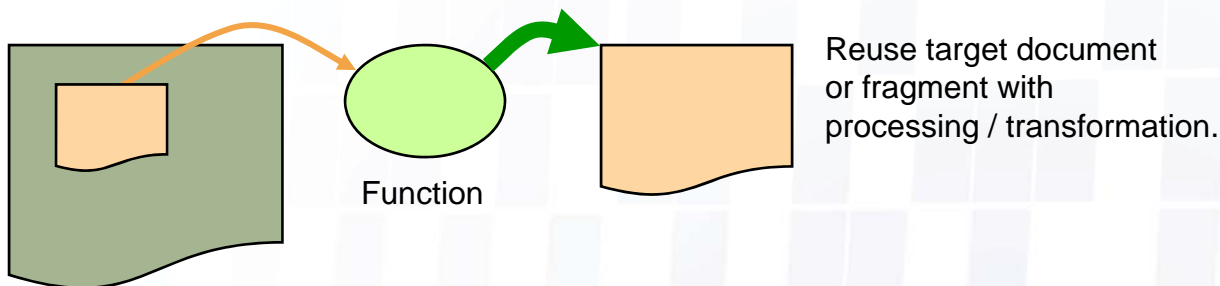


# Classify Reuse - Transitive Reuse

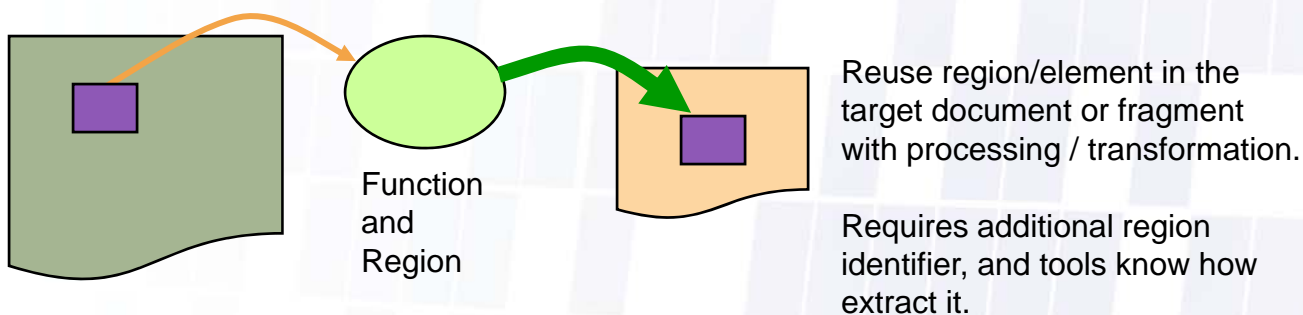


# Classify Reuse - Dynamic Reuse Types

## Dynamic Simple Reuse



## Dynamic Region Reuse



Dynamic reuse requires a function process, or transform, the targeted data as needed. This might be from one schema to another, or the extraction of only certain data.

## Reuse in S1000D

- 81 instances of the word “Reuse”
- Important Quote - “Developers of business rules must carefully consider the impact on reuse between projects and the potential for multiple customers and users of project data”
- S1000D has a lot less reuse than you might be expecting

# Reuse formally defined by S1000D

- Reusing Data Modules across different publications
- Reusing process data modules
- **The Technical Repository**
  - Technical information objects such as circuit breakers, access points, consumables
  - Quantity data, torque value or weights and measurements
  - Example of small chunk reuse
- **Container Data Module**
  - A production management mechanism to associate several data modules representing the same data.
  - Supports real-time resolution of reference based on applicability
  - Obscure form of reuse.

# Reuse Beyond S1000D

- **What other things should be considered?**
  - Cautions, warnings, and notes.
  - Components reused in Data Modules like:
    - <preliminaryRqmts>
    - <closeRqmts>
  - **Project Level Reuse**
    - BREX Modules
    - Standard Numbering System (SNS)
    - Information Codes
  - Exchange of this information

# Where Do You Go From Here?

- **What questions do you ask yourself?**
  - How do you manage and organize reuse?
  - What are the impacts of making a change?
  - How do you find it, track it and manage it (change control)?
  - Do you need to version it?
- **You can't do this by hand**

# Conclusions

- **Things a system should do to help facilitate reuse**
  - Notification when something changes
  - Help analyze and report the impact of a change
  - Assist the user in locating and finding, managing, and tracking reusable resources
  - Provide under-the-covers preservation of data integrity as reusability decisions are taken
  - Reuse must be part of the whole system from authoring to publishing to delivery
  - Preview and QA tools in the delivery process
- **Effective reuse starts early in the information modeling process**
  - Requires planning and design, it will not happen on its own

# Future Direction

- **Reuse never makes it way to the content consumer**
  - Reuse his concept is great for content originator but the content consumer could also benefit
  - Pass the reuse savings onto the content consumer
- **Start to exchange the technical repositories**
- **Formalize more project level reuse (SNS, InfoCode, Technames, Infoname)**
- **Formalize more component level reuse (W/C/N, components)**

# How to reach me

- InfoTrust Group
- 17671 Cowan Avenue, Suite 200
- Irvine, CA 92614  
555-555-5555

- Jeff Deskins
- Product Manager
- Product Engineering
- Office: (805) 468-4162
- Mobile: (949) 981-1746
- [jdeskins@infotrustedgroup.com](mailto:jdeskins@infotrustedgroup.com)

- Exhibit Hall Booth

