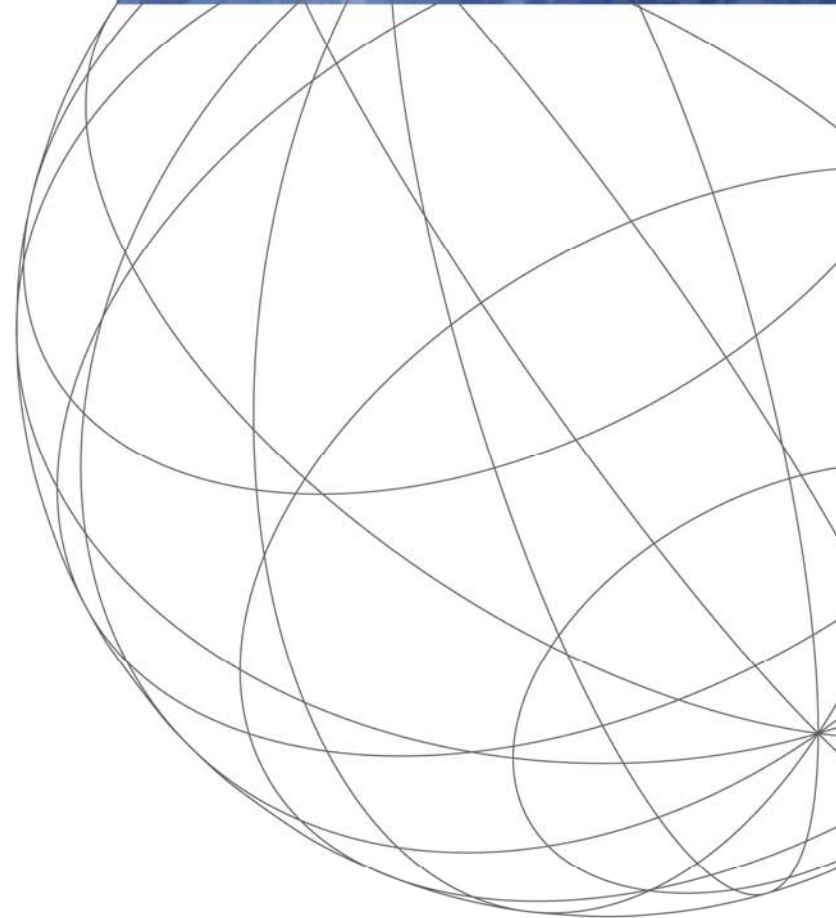




Introduction to P&C/Surety XML Version 2

Mark Orlandi, ACORD



Agenda: Monday, May 22, 8:30 – 9:45 pm

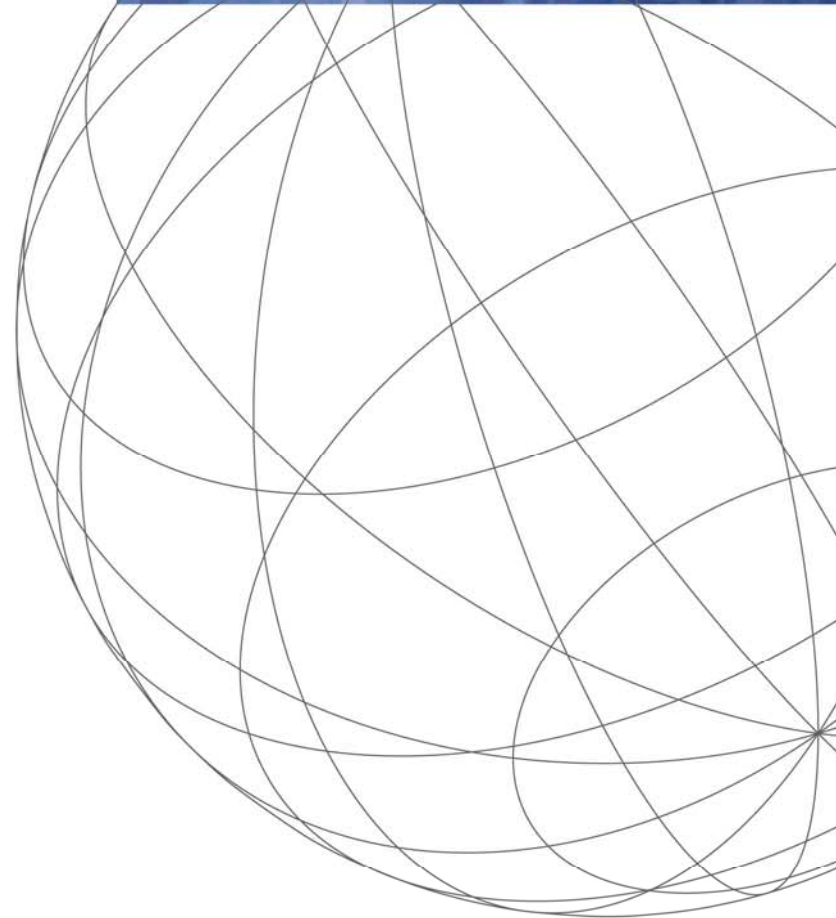
Part 1 – General Overview

- **1.1 Introduction**
 - History & Objectives
 - ACORD Committee Structure
 - Comparison with V1.x
 - Transition Plan
 - Work-To-Date
- **1.2 Design Overview**
 - Message Design
 - Design Principles
 - Architecture
- **1.3 Common Business Entities**
 - Party
 - Address





1.1 Introduction



History of ACORD PCS XML

- 1999
 - ACORD announced development of P&C/Surety XML and began business message specification design

- 2001:
 - ACORD P&C/Surety XML V1.0.0 first published in July

- Today:
 - The V1.x Standard is now in daily use
 - ▶ See OARS for Statistics and Details
 - Just published V1.10.0 Draft (May 12)



Why V2?

- After 4-7 years (where we are now), the need for a break from the past is becoming evident
 - Implementations have surfaced design issues with v1.x
 - XML technology itself has come a long way in the last three years
 - XML-based web services are now an everyday feature of data processing life for a number of companies
 - Companies now use XML technology internally as well as for external data exchange
- Version 2 Working Group
 - Commenced April 2003
 - Teleconferences & Face-To-Face Meetings (Teams)



V2: Overall Philosophy

- Originally envisioned as Evolutionary, but analysis proved need for major re-design
- No gratuitous changes
 - every change has a productivity payoff
- Keep what works – major changes as needed
- Better fit with horizontal/cross-industry standards (e.g. web services)
- Leverage non-insurance software tools
 - e.g. code generators
 - immediate payoff in developer productivity



V1 vs. V2: Similarities

- Business content of V2 encompasses v1.x, including corrections and improvements (e.g. remove all deprecated items)
- Human Readable
 - A major distinguishing factor from AL3 or “traditional” EDI
- XML
 - XML itself has evolved a lot since 2001
 - Schema over DTD
- Purposeful Use of Attributes (id / id ref)
- Error Handling
- Change Processing (<ChangeStatus>)



V1 vs. V2: Differences

What Makes V2 Better Than V1?

- Fewer documents; fewer messages
- More explicit data model
- Better XML Technology:
 - Uses XML namespaces
 - ▶ Allows for use of Web Services but impacts XPath
 - More fully uses the power of schema
 - ▶ Thereby retiring the DTD
 - Usable with code generation tools
 - Supports direct data exchange with relational databases
 - Extensible via schema



V1 vs. V2: Differences

What Makes V2 Better Than V1? (continued)

- **Improved human readability**
- **Removed deprecated tags/codes**
- **More referencing**
 - Less repetition
 - Less inconsistency
- **More rigorous validation**
 - Strong data typing
 - Data type hierarchies
- **More tags and fewer codes**
 - Will see that with Party and Address



Transition Plan

■ Phased Approach

- 3 Phases = 3 “Candidate Drafts”
- “Broad foot-print” Concept
 - ▶ 1st Phase includes a little bit of everything
 - ▶ Phase 1.0 – Delivered Dec 2005
 - ▶ Phase 1.1 – Delivered Feb 2006 (Help file)

■ Candidate Recommendation “CR”

- At conclusion of Phase 3, consider movement to “CR” status like W3C
- All messages are “CR” until successfully piloted

■ Official/Final Standard

- Occurs post-pilot, membership / steering committee approval



Phase 1 – Business Content (Dec 2005)

■ Claims

- Workers Comp, FROI-SROI

■ Commercial Lines

- Auto
- Workers Comp

■ Personal Lines

- Auto
- Homeowners
- Dwelling Fire
- Inland Marine
- Package
- Watercraft

■ Inquiry

- Billing

■ Statistical/Regulatory Reporting

- Workers Comp

■ Surety

- Bond
- Report Of Execution



Phase 2 – Business Content (June 2006)

- **Claims**
 - Property, FNOL
 - Liability, FNOL
 - Auto, FNOL
 - Loss Runs
 - Investigation
- **Commercial Lines**
 - BOP
 - GL
 - Property
 - Package
- **Farm**
 - Package
- **Personal Lines**
 - Umbrella
- **Inquiry**
 - Policy
 - View
- **Statistical/Regulatory Reporting**
 - Workers Comp
- **Surety**
 - Bond Add
 - Contract Audit +
- **Miscellaneous**
 - Policy Sync
 - Dwelling Valuation
 - Premium Audit
 - Policy Finance



Phase 3 – Business Content (Dec 2006)

■ Claims

- Download
- Status Inquiry

■ Commercial Lines

- Boiler & Machinery
- Crime
- D&O
- E&O
- EPLI
- Inland Marine
- Schedules *
- Umbrella

■ Inquiry

- Account

■ Statistical/Regulatory Reporting

- Workers Comp

■ Surety

- Billing Summary Notify
- Inquiry
- Bond Number Sync
- Rate Sync

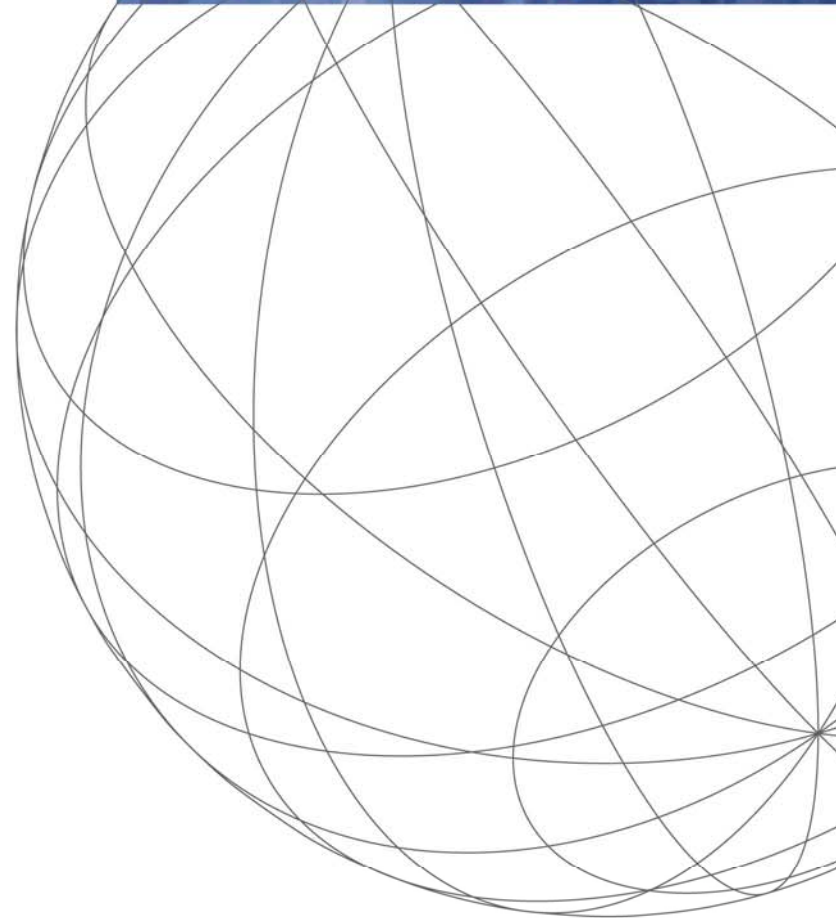
■ Miscellaneous

- Activities & Notes
- Certificate Of Insurance
- Lender Replace
- Flood





1.2 Design Overview



Design Overview Agenda

■ Architecture

- Type Hierarchies
- Data Types
- Naming Conventions
- Code Lists
- Extensibility - elements and code lists

■ Message Design

- Background - Count of messages and reduction
- De-coupled notify messages



Why Do We Use “Type Hierarchies”?

- In insurance, many common business concepts are “the same but different”
 - Parties
 - Policies/Surety Bonds
 - Locations/Sublocations
 - Coverages/Adjustments
 - Business Messages
- Version 1.x uses Entities (Copy/Paste)
 - %PCPOLICY, %PCLINEBUSINESS
- Version 2.0 uses Type Hierarchies extensively
 - Base Types contain the common data
 - Derived Types contain specialized data



Why Do I Care About Type Hierarchies?

- Easier implementation through more efficient use of tools
- Handled with Entities in 1.x which are not code friendly
 - Processing code is not “Aware” that this is a message
- Supports Better Code Generation
 - The type hierarchy in the schema gets turned into a type hierarchy in the programming language
 - This gives the programmer more control with less code
- Supported natively by emerging Horizontal Standards such as XQuery and XSLT/XPath 2.0
 - Can make processing decisions based on types
- Can process messages, parties, addresses, etc. generically
 - Even though each instance has a different <tagname>



Background on Version 1.x Message Design

- Message Design actually began in 1999
- Limited Scope
 - Personal Auto, Home, and BOP New Business and Quotes
 - As New LOB and Transactions were Added, The Number of Messages Grew Significantly
- After 1.1, it was determined that a fix would break backward compatibility so It was decided to wait until 2.0
- Service Provider Initiated Transactions (i.e. Download) were added in 1.2 which caused the count of policy messages to grow exponentially
- There are now 832 Request and Response messages (416 pairs)



Version 2 Message Design Objectives

- Reduce the number of messages required by V1.x design
- Retain the best features of V1.x
 - Stronger validation than AL3 or ACORD XML for Life
 - Strong error reporting and change status
- Enable implementers to leverage tools and have better reuse of common message types
- Continue explicit message design model partitioned by:
 - Line Of Business (Personal Vehicle, General Liability, etc.)
 - Transaction Type (New Business, Cancel, etc.)
 - Object (Policy, Claim, etc.)
- Improve validation by use of specific tags rather than codes (e.g. CoverageCd, RoleCd)
- Ease Extensibility - Allow users to create new message types based on existing messages



Version 2 Message Design Solution

- **Considered All-Inclusive approach per AL3/Life**
 - Fell short of validation requirements
 - Would result in decreased ability to leverage XML to process messages in meaningful ways
- **Similar Naming Convention**
 - Line Of Business + Object + Action + Direction
- **Slight Reduction in Message Count**
 - Ongoing work in V2 concluded granular messages were still needed to meet other design goals
- **Improved Validation**



V1.x Insurance – Line Of Business Messages

- **Typical LOB Message Sets (e.g. Homeowners)**
 - Submission Message Sets
 - ▶ Quote, Add, Mod, Reinstatement, Reissue, Renew (6)
 - Notification Message Sets
 - ▶ Update Message Sets
 - Add, Mod, Reissue, Renew, Reinstatement (5)
 - ▶ Replace Message Sets
 - Add, Mod, Reissue, Renew, Reinstatement (5)
 - Combined Per LOB: $6+5+5=16$ Transactions
 - All transactions are a Request/Response Pair (2)
 - $16 \times 2 = 32$ messages per line of business
 - 23 Lines of Business \times 32 messages = 736 msgs.
 - Plus 8 partial image cancel/reins. = **744** total msgs.

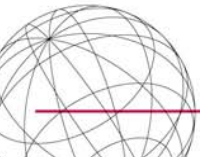


V2 Insurance – Line Of Business Messages

- **Typical LOB Message Sets (e.g. Homeowners)**
 - Submission Message Sets
 - ▶ Quote, Add, Mod, Cancel, Reinstate, Reissue, Renew (7)
 - ▶ Each occurs as a Request/Result Pair (2)
 - ▶ $7 \times 2 = 14$ messages
 - Notification Message Sets
 - ▶ Quote, Add, Mod, Cancel, Reinstate, Reissue, Renew (7)
 - ▶ Each occurs as a single Result (1)
 - ▶ $7 \times 1 = 7$ messages
 - Combined per LOB: $14 + 7 = 21$
 - 23 Lines of Business x 21 messages = **483** total msgs.
 - 744 reduced to 460* = 38% reduction in msg count
 - ▶ *Note – V1 did not include Quote Notify

Code Lists

- All Code Lists Reviewed and Updated
- Naming Conventions Are Now Consistent
 - Requires the Use of Alpha Mixed Case Codes
 - Improved Readability
 - ▶ V1 = <GenderCd>M</GenderCd>
 - ▶ V2 = <GenderCode>Male</GenderCode>
 - No Numerics in First Character Allows us to Take Advantage of an XML Extension Technique Called QName
 - ▶ More on This Later



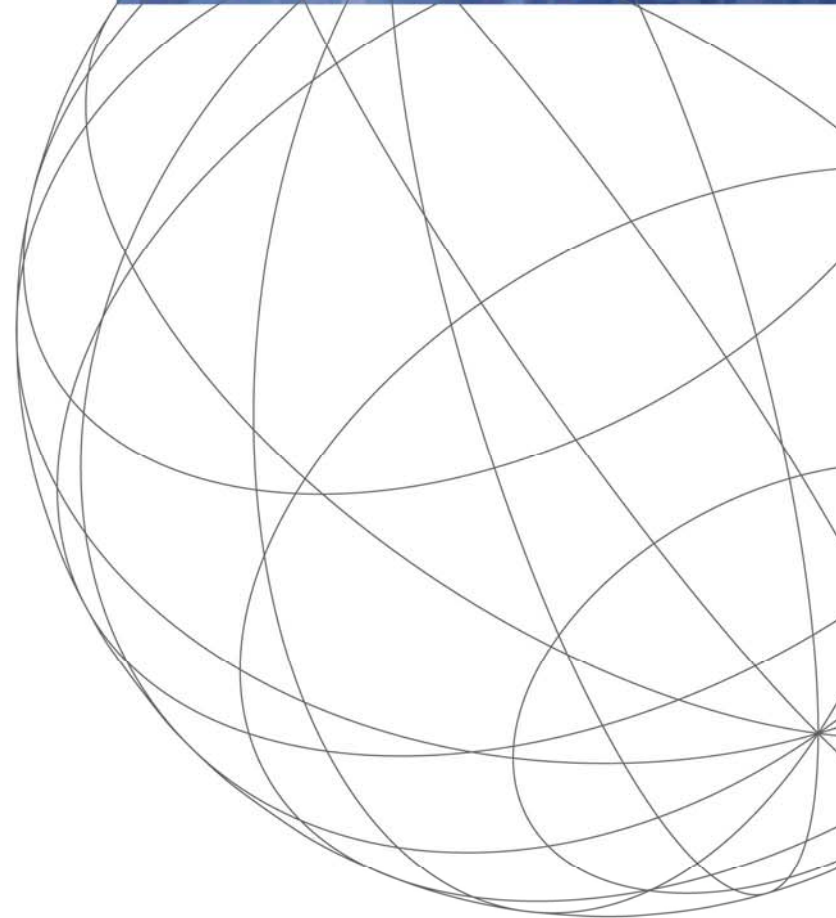
Extensibility Model

- What if the ACORD PCS design has not accounted for all of your data and messaging requirements?
 - Standard Extensions of the ACORD XML Schema
 - Use the Namespace for Guaranteed Uniqueness
- Create Company-Unique:
 - Data Types
 - Data Elements
 - Code Values
 - Aggregates
 - Messages





1.3 Common Business Entities



Common Business Entities Overview

- Party
- Address
- Coverages
- Sample Data Stream



V1.x Issues - Party

- Some roles were defined individually
 - <Producer>, <InsuredOrPrincipal>, <AdditionalInterest>
- Most roles were generic, <MiscParty>, with a RoleType code
 - 104 role type codes, 16 sub role type codes
- Inability to validate data related to a specific role
- No clear distinction between a person and an organization



V2.0 Resolution - Party

- Use Type Hierarchies
- Clear distinction between <Person> & <Organization>
- All parties to the transaction listed at top of message
- Eliminated the use of party role codes by creating individual tags for Policy transactions
- This design provides for tighter validation



V1.x Issues – Address

- Inconsistent use of address versus location
- Required individual processing
- Cannot validate the distinct data requirements by line of business
- Often had to repeat data when one location served multiple roles, e.g. insured location and garaging address



V2.0 Resolution – Address

- Use Type Hierarchy
- All addresses listed at top of message
- Eliminated address type codes creating individual tags
- Segregated Address from Location allowing for idref to associate the two
- This design provides for tighter validation



V1.x Issues – Coverage

- **Unclear distinction between “coverages” and “adjustments”**
 - Coverages / Credits / Surcharges / Taxes / Fees
 - These are all contained in the same code list
- **Inconsistent use of the above**
- **Inability to identify “adjustments” outside of “coverage”**
- **No validation of coverages by line of business**
 - e.g. Could place auto coverage in work comp



V2.0 Coverage – Resolution

- Modeled Types specific to “premium generation”
 - Coverages
 - Credits Or Adjustments
 - Taxes Fees Or Assessments
- Separate code lists for:
 - Coverages/Credits/Adjustments/Fees
- Modeled coverages & coverage options by line of business
 - PersonalVehicleCoverage, BOPCoverage, etc.
 - PersonalVehicleCoverageOption, BOPCoverageOption, etc.
- Sample data stream



Wrap-Up - Questions

- **Call for Participation**
 - Register for V2 WG
 - <http://www.acord.org/Standards/workinggroups.aspx>
- **ACORD Teams**
- **Teleconference schedule**
- **Q&A**

