

Consolidated Renal Operations in a Web Enabled Network  
(CROWNWeb) Release 4.0

Document Version: XML Schema Design

Document Version 0.2

Release Version: CROWNWeb Release 4.0

May 6, 2011

# XML Schema Design

Edaptive Systems

400 Red Brook Boulevard, Suite 220

Owings Mills, MD 21117

## Table of Contents

1. Introduction.....	3
2. Referenced Documents .....	3
3. Purpose of this Document.....	3
4. Background .....	4
5. Goals .....	4
6. XML schema Design Structure.....	4
7. Influencing Factors .....	5
7.1 New elements .....	5
7.2 Resolve discrepancies.....	5
7.3 Reduce business-logic inconsistencies .....	6
7.4 Common validation logic .....	6
7.5 Improve schema structures .....	6
7.6 Align with domain ontology.....	6
7.7 Facilitate enhancement .....	6
7.8 Improve internal documentation.....	7
7.9 Integrate with infrastructure .....	7
7.10 Improve performance.....	7
8. Schema Design Principles.....	7

## 1. Introduction

The CROWNWeb application of the Centers for Medicare and Medicaid Systems (CMS) is the web application for Medicare's End-Stage Renal Disease clinical management processes. Medical providers use CROWNWeb to submit and retrieve information pertaining to the delivery of renal care, including dialysis and kidney transplants. Users interact with CROWNWeb via its graphical user interface (GUI) or via XML documents containing equivalent data.

## 2. Referenced Documents

Document Name	Document Number	Issuance Date
CROWNWeb_3.0_XML_schema_delta.docx		February 1, 2011
CROWNWeb_4.0_XML_schema_ICD.docx		May 6, 2011

## 3. Purpose of This Document

The schema design for CROWNWeb 4.0 has been enhanced significantly from that of CROWNWeb 1.2. The changes include:

- New requirements to accommodate Clinical Performance Measures (CPM)
- Lessons learned from the earlier versions
- Reduction of data discrepancies
- Elimination of business-rule inconsistencies between EDI submissions and those from the GUI
- Improvement of data quality
- Compatibility with the planned addition of submissions via the National Health Information Network (NHIN)
- Facilitated integration with business logic
- Ease of understanding by technical staff
- Simplification of maintenance
- Expandability for future requirements.

To accomplish these goals, the schemas for CROWNWeb 4.0 employ various idioms in accordance with design principles of modularity, clarity, useful internal documentation, ease of implementation in software, re-use of common structures, semantic coherence, and separation of concerns. This document explains these idioms, their rationale, and details of the changes from the CROWNWeb 1.2 schemas.

## 4. Background

CROWNWeb originally added XML processing in order to support large dialysis organizations (LDOs) that gather large amounts of patient and clinical data on behalf of affiliated facilities. Certain organizations qualify as Batch Submitting Organizations (BSOs) and submit their constituents' data in batches as XML documents via the CROWNWeb application. The structure and content of these XML documents conform to specifications encoded in XML Schema documents (XSD documents).

The original schemas were created to support a pilot of the CROWNWeb batch functionality involving a small number of BSOs and their constituent facilities. Over the course of the pilot, lessons were learned about the strengths and shortcomings of the batch submission process. It successfully permitted the BSOs to submit large collections of patient demographic and clinical data, fulfilling the primary purpose. Along the way, the community observed some discrepancies between batch-submitted data and data submitted via the GUI or the legacy systems.

## 5. Goals

Goals for CROWNWeb 4.0 XML schema redesign include:

- Resolve discrepancies between Batch (EDI) and User Interface entry.
- Incorporation of new data elements including clinical performance measures
- Preparation for national roll-out
- Incorporation of lessons learned from Batch (EDI) pilot
- Enhancement of stability and expandability of the CROWNWeb application

## 6. XML Schema Design Structure

CROWNWeb 1.2 has two types of schemas defined:

1. EDI Inbound XML Schemas (2 Types)

This is the XML file submitted by the LDOs/BSO to CROWNWeb application representing two types of data, patient demographic data and clinical data (medical test results, prescription information, treatment data, etc.).

2. EDI Outbound XML Schemas (2 types )

These are the XML documents with processing summary results for each of the two types of submission.

CROWNWeb 4.0 will use the same schema types – but common definitions have been factored out into subordinate schema documents to represent shared structures. In addition to the four schemas that represent demographic or clinical submissions and feedback, there are abstracted schemas that represent data shared by the primary schemas, such as “Address” structures or measurement units.

## 7. Influencing Factors

The influencing factors which prompted the XML schema changes in CROWNWeb 4.0 were:

- Need to add new elements to meet CROWNWeb 4.0
- Include solutions to address batch-submission discrepancies.
- Reducing the risk of inconsistencies between different XML submission types (currently comprising patient demographic information and patient clinical information).
- Enabling common validation, processing, and persistence logic for data entered whether via the GUI or via XML communication channels.
- Improve the schema structures to be more logical and maintainable.
- Bring the XML structures into closer alignment with the domain ontology.
- Anticipate and facilitate future enhancements, such as planned additional enterprise communication channels for CROWNWeb XML data.
- Improve the clarity of internal documentation and nomenclature within the schemas.
- Facilitate integration with planned enhancements to the enterprise infrastructure.
- Improve performance and throughput for XML document parsing.

These goals interact with each other, for example alignment with domain concepts achieves better organization of the schema structures and vice versa. Consequently, the schema is clearer to understand, and it is easier to add new attributes to distinguish intended results for submission data. The organization of the schema lends itself to patient-by-patient processing of batch submissions, which allows clean separation of concerns between the application-tier and data-tier logic, and easier options for scalability and throughput management in production. Taken together, all these changes help the XML processing subsystem to share validation and storage logic with the GUI subsystem, and to resolve and prevent discrepancies between CROWNWeb data and field results.

### 7.1 New Elements

The requirements for CROWNWeb 4.0 mandate approximately 150 new or changed elements within the XML submissions, i.e., to be declared in the XML schemas. Most of these are included in the clinical submission schema; some are included in the patient demographic schema. Because of the large number of new elements, the schema changes needed for CROWNWeb 4.0 were fairly significant regardless of any other alterations.

### 7.2 Resolve Discrepancies

CROWNWeb 4.0 adds “Action” attributes to key structures to permit the business logic to distinguish between certain intended results, for example whether to add data as new information or to update information already stored. Version 4.0 also removes certain schema-level validations from XML parsing. Between those changes and the

overall schema reorganization, the XML handlers within CROWNWeb will be able to distinguish intended actions, for example to add a new record or update an existing one. In combination with the improvements in business-logic inconsistency and changes in the application logic, this will reduce discrepancies between CROWNWeb data and actual clinical history.

### **7.3 Reduce Business-Logic Inconsistencies**

Some schema validations were removed for CROWNWeb 4.0. XML schema processors can verify certain requirements on XML data, for example whether an element is mandatory, must appear with other elements, or must be of a certain data type such as a date, numeric, or Boolean (true/false) value. Repeated data types, or custom types used in place of equivalent standard schema types, were identified and redefined as common or standard types, respectively. Internal element and attribute names were expanded for clarity. Taken together, along with middleware changes to share validation and persistence logic, these changes reduce the risk of inconsistent application of business logic.

### **7.4 Common Validation Logic**

Some schema-level validations were eliminated for CROWNWeb 4.0. The middleware logic now performs these checks, using the same code that validates data entered via the GUI.

### **7.5 Improve Schema Structures**

Taken together, the use of common schema structures in centralized locations, the renaming of schema elements, attributes and types, and the organization of schema elements within enclosing elements and other hierarchy alterations make the schemas simpler, more robust, more self-explanatory and better able to handle future expansion.

### **7.6 Align with Domain Ontology**

The new structures and renamed components of the schemas follow the domain terminology more closely. Elements were analyzed for interdependency and allocated to structures accordingly. Abbreviations are less cryptic and internal documentation (using “annotation” and “documentation” schema elements) is much more complete. This makes both the schema documents and conforming XML submission documents easier to understand and to handle programmatically. Over time, this will reduce the cost of maintaining XML-submitting systems for the networks and facilities, and ease troubleshooting when it is needed.

### **7.7 Facilitate Enhancement**

With more logical structures, single points of change for common elements, closer alignment of nomenclature with domain terminology, use of included (“imported”) common schemas and improved internal documentation, it will be much easier to incorporate new or changed schema requirements in future CROWNWeb versions.

## 7.8 Improve Internal Documentation

Clearer documentation and explanation of the relationship between schema structures and domain concepts makes it easier to use, program to, and deploy systems based on the schemas.

## 7.9 Integrate with Infrastructure

Between the schema redesign and improvements to the XML-handling software components, it is possible to bring the batch-handling deployment of CROWNWeb into compliance with the CMS three-tier architecture.

## 7.10 Improve Performance

The reorganization of the schemas and the business logic has eliminated certain known bottlenecks in CROWNWeb 1.2's XML processing. In conjunction with the three-tier deployment, it allows more flexibility in configuration to scale production throughput.

## 8. Schema Design Principles

To support these goals, the XML schemas use various idioms and common patterns, along with a strategy for document organization. The design principles are:

- Use standard schema types where available, rather than custom types. “Don’t reinvent the wheel.” This makes it easier for the BSOs to program the software to use the schemas.
- Reuse types wherever sensible – don’t create multiple identical or nearly identical versions of the same thing. A single point of maintenance avoids the risk of divergence between multiple points.
- Prefer named schema types to anonymous “one-off” types. This facilitates re-use and refactoring.
- Schema names and enumeration values should be descriptive and meaningful, not overly abbreviated or obscure. They should be short enough to be useful, but long enough for ease of understanding.
- Element names should begin with a lower-case letter. Type names and enumeration values should begin with an upper-case letter. Internally, names should use “camel case” – capitalization of the first letter of each compound word part. For example, an element name might be “lastName”.
- Types used by more than one schema should be defined in a subordinate schema for importation into the primary schema. That way, more than one schema can import the same common set of types.
- Use namespaces.
- Where a common type is not possible, but more than one type is similar, use parallel structures. For example, in the CROWNWeb 4.0 schemas, there are many types that list specific choices with a final “Other” option, accompanied by an optional free-form text element to hold the details of the “other” choice. All these types are implemented as a “ListedX” type with an enumeration of valid values for “X”, with

an “X” type that comprises a “ListedX” enumeration-type element named “listed” and a “string”-type free-form element named “other”. The regularity of the pattern supports maintenance and software development for the schemas.

- Perform only surface validations within the schema itself; perform the rest in code. For example, the schema can enforce whether an element is unconditionally mandatory or only if a certain element is present. This constitutes a surface validation similar to how the GUI might require a value for a particular screen entry. Any deeper validation, such as enforcement of a value’s uniqueness, was removed from the schemas and pushed back to the middleware for programmatic verification. Simply put, the schema enforces the structure and presence of content, the middleware enforces its correctness.
- Fill in the XSD “annotation” and “documentation” elements. These are the internal comments of the schema document, and contain copy-and-pasted text from requirements and analysis documents.
- Keep the schemas as simple as possible, but no simpler.

These design principles make the schemas simpler to understand, maintain, and implement.