Challenges in electronic documents for Parliaments

Prof. Fabio Vitali
University of Bologna

Purpose of this talk

- Demonstrate that parliamentary processes and citizens' access to parliamentary records and documents can be improved by:
  - Adopting the best technologies for document management (mainly, XML and related standards)
  - Adopting a common, multi-linguistic, multi-national standard format for electronic representation of documents (mainly, Akoma Ntoso)
  - Foster the creation and adoption of many different hardware and software architectures that are made available to support these standards (mainly, Bungeni).

Parliamentary activities

- A complex production system that generates documents with different legal status:
  - Bills and acts to become the law of a country
  - Minutes (or hansards) to become a lasting log of the activities of the parliament
  - Daily and weekly announcements, tablings and reports to organize and master the internal logistics
- Each document of the uttermost importance and to be printed in large quantities and/or made available to a wide part of the population daily and in a very limited amount of time

Next: Computer support for parliamentary activities

Computer support for parliamentary activities

- Support for generating documents
  - Drafting activities, record keeping, translation into national languages, etc.
- Support for workflow
  - Management of documents across lifecycle, storage, security, timely involvement of relevant individuals and offices
- Support for citizens' access
  - Multi-channel publication (on paper and on the web), search, classification, identification
- Further activities
  - Consolidation, comparison, language synchronization, etc.

How the Web can help

- Born as a publishing medium
- HTML helped make it a big success
- HTML constraining by its own simplicity
  - Excessive reliance on typographic rather than semantic description
  - Few rules not even strongly imposed
- A new language was invented, called XML, that could solve that
  - Clear differentiation between aspect and meaning
  - Strong syntactic rules heavily imposed to warrant uniformity, homogeneity, sophisticated applications

Norms and XML

- Norms are ideal for being expressed in XML:
  - They have a well-defined structure, which is systematic and standardized
  - There are required and optional parts according to rules and tradition
  - There are containment constraints that determine the global correctness of the document
  - There are references to other texts (attachments, other norms, etc.) that can fruitfully be used to create a hypertext network.

Schema languages

- Complete vocabulary and construction rules for verification of the correctness of documents
  - Inspiring homogeneity
    - A source of inspiration for armies of authors and editors
    - Identifies a style for document, standardizing the terms used
  - Driving editing activities
    - A help for law editing
    - Provides ready-to-use structures to fill in with text, and methods to mark the most common occurrences of law structures.
  - Providing validation
    - A way to guarantee adherence to rules for texts edited or marked by any author or editor.
    - Provides checklists and verification mechanism to check whether the document is homogeneous to the collection already being used.

The wrong way to do it (1)

- Hire a software company and ask for a software/hardware solution that handles document management, storage, workflow
- Software company
  - Pulls users' requirements before coding
  - codes the new software,
  - identifies required hardware, and
  - generates data formats according to software
  - Delivers a complete solution according to user requirements

The wrong way to do it (2)

- Pros:
  - One decision to make
  - One organization to blame for failure
  - Everything tightly integrated by design
- Cons:
  - Total dependence on such software company, now and in the future
  - Documents tightly dependent on built software, with no real chance of letting them out towards a new and improved solution
  - Little flexibility towards new functionalities, little exceptions, and long-term survival of documents

A good way to do it

- Separate document design from software architecture
- Have a document standard generated that satisfies long term goals for storage, readability, semantic sophistication and future applications
- Hire a software firm that implements software based on such document specification

The *best* way to do it

- Buy into an existing, widespread initiative that identifies data formats and document formats for different countries.
- Ask for, if needed, minor customization in order to have local standards conform to international initiative
- Contribute to generate market for software companies and open source initiatives to provide required software and necessary customizations.

Next: Key points to remember

Key points to remember

- Document formats need to be defined independently of actual software, and never relinquished IP on them to private companies
- Non-proprietary solutions ***necessary***
- Document formats to be descriptive, rich, designed for multiple delivery channels, flexible.
- Software built on top of document formats, according to independent specification
- Customizing international solutions better than inventing one's own.