



Advanced Automated Authoring with XML

Petr Nálevka <petr@nalevka.com>

University of Economics, Prague

XML Prague 2009



Introduction

This presentation shows how to automate authoring of large detailed and highly visual documents using XML technologies.

Advantages

- **Automatic generation of document fragments (XSLT, DocBook)**
- **Styles are easily pluggable and applied automatically (DocBook)**
- **Modular and reuseable data (XInclude)**
- **Professional typesetting with highly customizable output (DocBook, FO)**
- **Highly visual documents (SVG)**
- **Collaboration and versioning (Subversion)**
- **Visual authoring (XML Mind)**

Disadvantages

- Authors need to learn new tools
- Authors don't see what they get immediately
- Higher initial time to build, but gets payed off
- Not an out-of-the-box solution

Aims

- **Documentation for large networking projects**
- **Several sites, different equipment, network topology, ip plan, wireless connections**
- **Documentation supports the project through its whole life-cycle**
 - **Proposal**
 - **Design**
 - **Implementation and testing**
 - **Support**

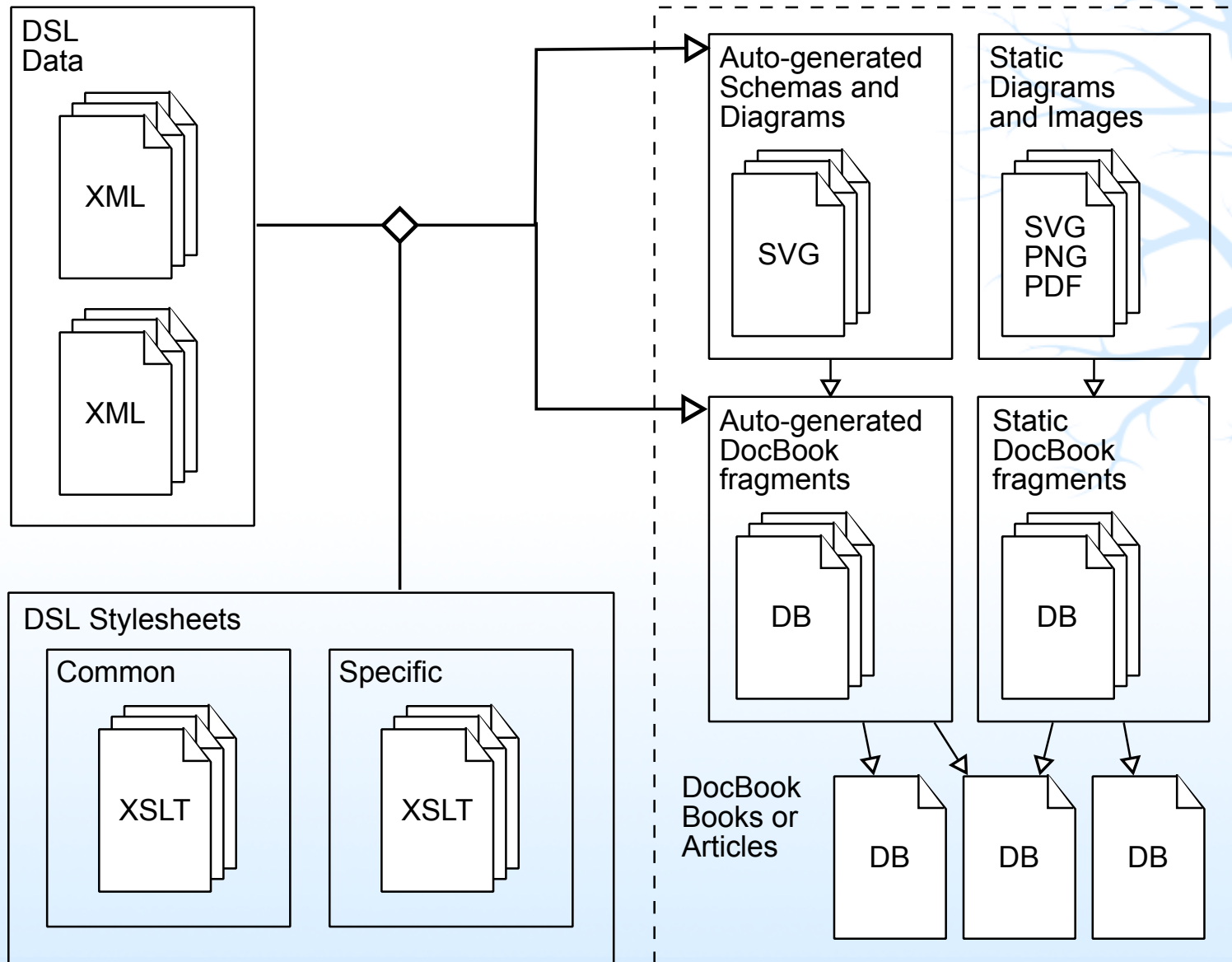


The Authoring Framework

Domain model, Architecture, Components, Publishing process...

Architecture

Generating DocBook out of DSL.

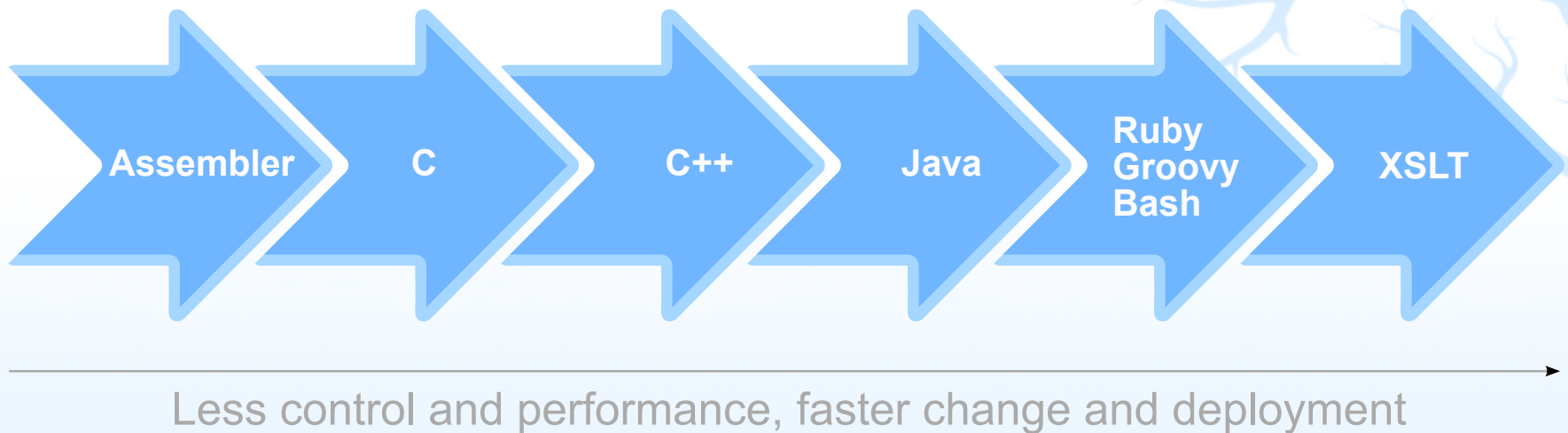


XML Domain Specific Language

- It is really a *FRAMEWORK*
- XML allows to design a nice DSL and use all the rich XML toolchain on it
- Domain specific rather than generic grammar
- Nicely designed DSL = maintainable, understandable, no redundancy, easy change, cleaner processing
- Your domain model is the most rapidly changing part of your system

XML Advantage

- More agile than scripting languages
 - One data format no mapping between formats
 - Declarative approach
 - Focused on trees



Making XML even more flexible

- No strict schema
- Define Fixpoints
- Loose stylesheets
- Loose schemas — Schematron
 - Check fixpoints
 - Check consistency





Visual Documents

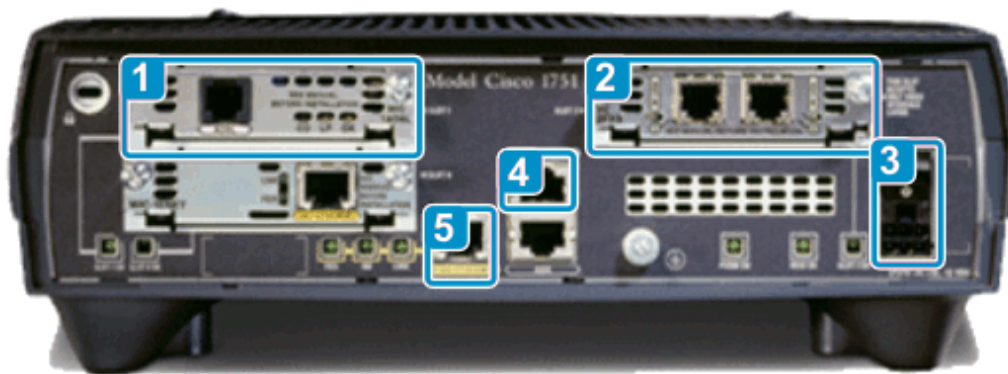
Automating visual documentation

Getting Visual

- **Visual information is much more understandable for humans than text.**
- **Make people understand and you may win a tender, sell a product or use less skilled staff.**
- **Visual information is difficult to maintain but not if you can automate it.**

Use-cases — Image callouts

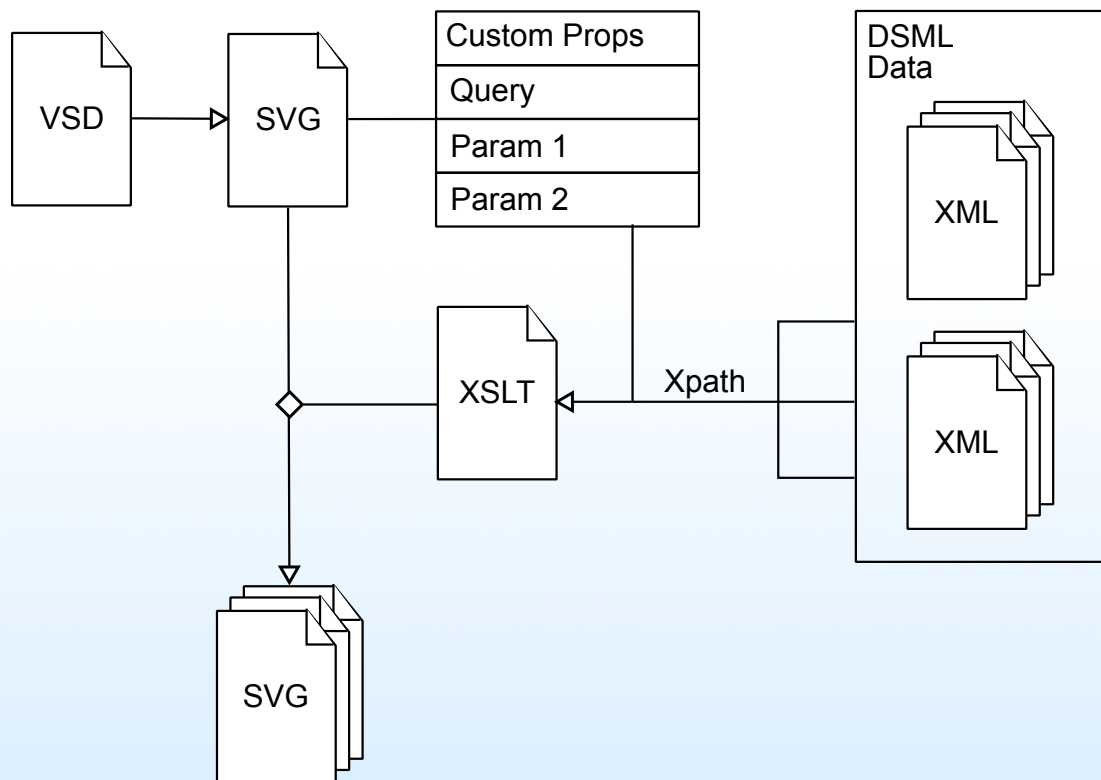
- DocBook feature, not implemented in stylesheets
- Pure XSLT/SVG implementation



- 1 WIC/VIC Slot 1
- 2 VIC Slot 2
- 3 Power switch and power socket
- 4 Console port
- 5 10/100 Ethernet port

Use-cases — Visio Diagrams

- Layout is controlled by the author
- Using Visio's custom properties
- Populating Visio diagrams with DSL data



Use-cases — Rack Layouts

- Custom diagrams from scratch using **SVG**
- **SVG** cliparts (may be exported from Visio)
- Rack devices and their position is defined in **DSL**

Use-cases — Google Earth

- Another dimension of visuality
- Keyhole Mark-up Language (KML)
- Two approaches
 - Generate interactive map out of DSL data.
 - Plan the project in GE and than transform to DSL
- Wireless coverage example

Future work

- **Charts**
- **Graphs**
- **Maps**





Visual Tools

Untrained secretaries don't understand COBOL and they also won't understand XML.

Integration with Visual Tools

- **Visio (SVG)**
- **Google Earth (KML)**
- **Excel (CSV)**



Visual XML Editor

- **Conversion from Word can hardly be fully automated**
- **Need of visual XML editor for non-technical users**
- **XMetal, Oxygen, Epic, Serna**
- **XML Mind**
 - **Well balanced shielding**
 - **Great extensibility**
 - **CSS styling**
 - **Always *VALID***
 - **Pasting tables, lists, paragraphs**
 - **Out of box DocBook support**
 - **Support for modular documents**
 - **Competition — Word but also textual XML editors**



Thank you for attention!

**For any additional questions, please, contact me at
<petr@nalevka.com>.**