

One project, four schema languages; medley or melee?

Family

Given

MURATA Makoto

International Univ. of Japan

Tags and Schemas (2004/10-2005/4)

- 1947 tag names
- 119 schema modules
 - RELAX NG
 - W3C XML Schema
 - DTD
- 128 schemas in Schematron

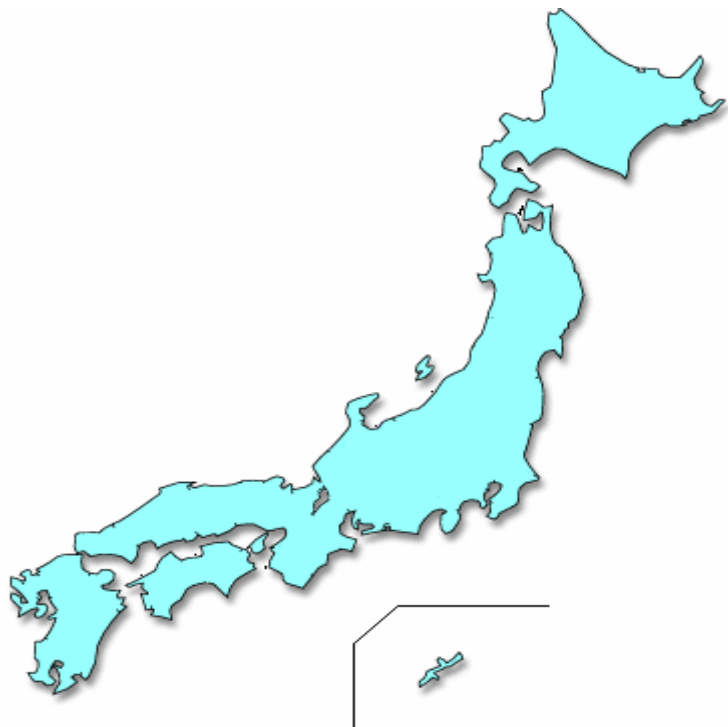
Outline

- Local governments in Japan
- Towards e-local governments
- Document analysis
- Document design
- Schema development environment
- Schema documentation
- Using Japanese characters



Local governments in Japan

Japan



47 Prefectures



200 municipalities in Hokkaido



[Abashiri](#) | [Akabira](#) |
[Asahikawa](#) | [Ashibetsu](#) |
[Bibai](#) | [Chitose](#) | [Date](#) |
[Ebetsu](#) | [Eniwa](#) | [Fukagawa](#) |
[Furano](#) | [Hakodate](#) | [Ishikari](#)
| [Iwamizawa](#) | [Kitahiroshima](#)
| [Kitami](#) | [Kushiro](#) | [Mikasa](#) |
[Monbetsu](#) | [Muroran](#) |
[Nayoro](#) | [Nemuro](#) |
[Noboribetsu](#) | [Obihiro](#) | [Otaru](#)
| [Rumoi](#) | [Sapporo](#) | [Shibetsu](#)
| [Sunagawa](#) | [Takikawa](#) |
[Tomakomai](#) | [Utashinai](#) |
[Wakkanai](#) | [Yubari](#) ...

Hokkaido yesterday



Local governments are not rich

- Local tax is not enough.
 - Hokkaido: Only 20% from local tax
- Local governments depend on fund from the central government and local government bond.

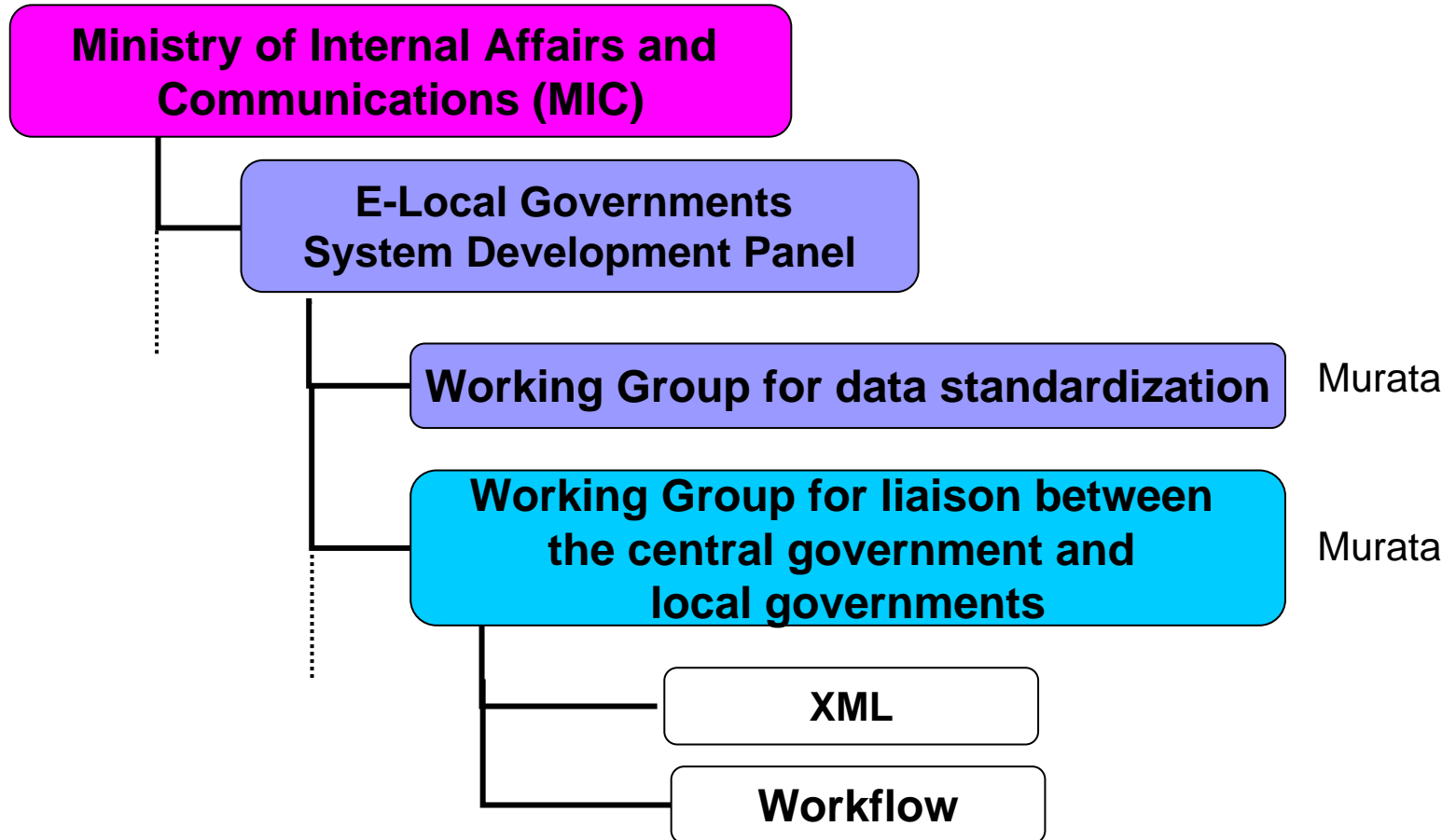
Tight control from the government

- Local governments have to report almost everything to ministries.
- More than 700 periodical reports
 - For example, the number of employees 50 years old and over.
- FAX, snail mail, floppy disk, ...
- Local governments are *tired*.



Towards e-local governments

Organizational framework



Scope

■ Target documents

- Initially, periodical reports from local governments to MIC
- In the future, every information interchange between local governments and ministries.

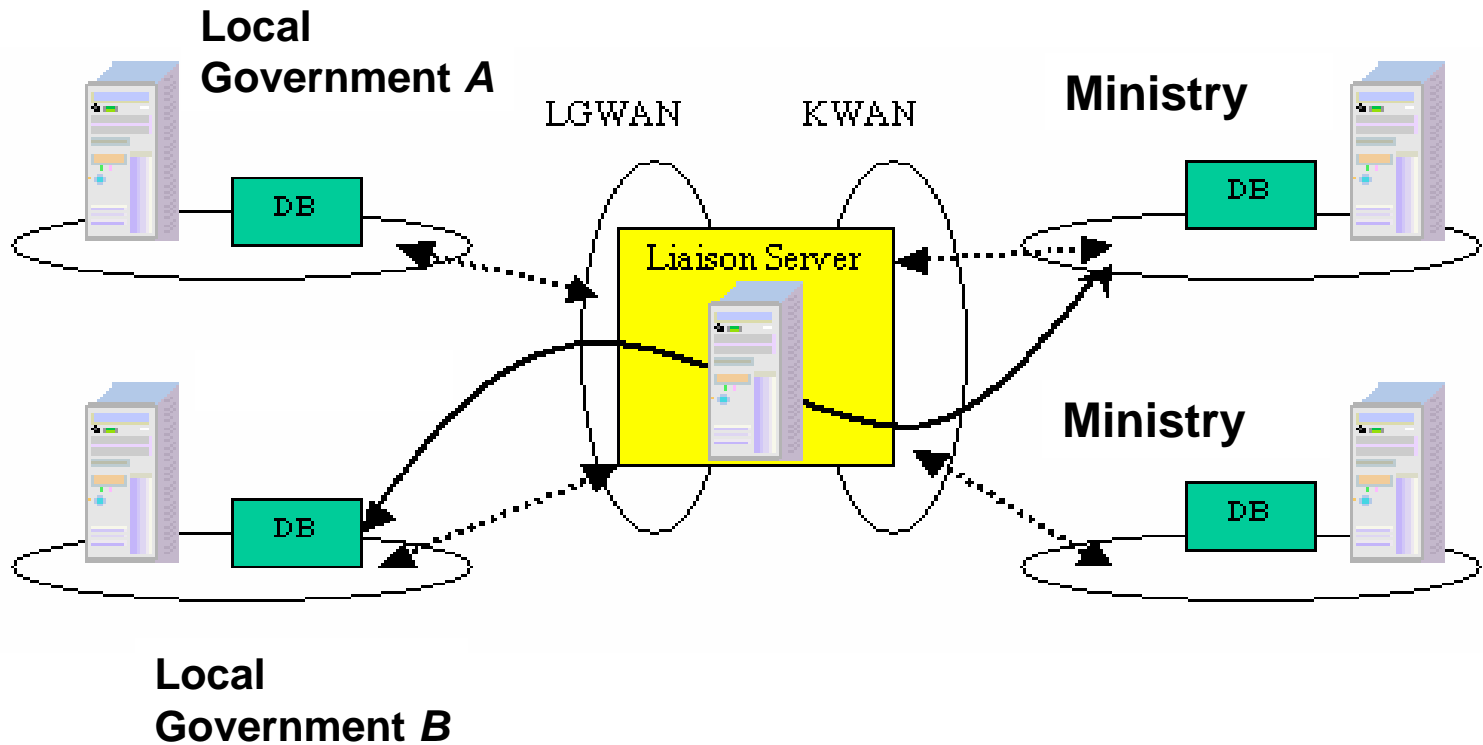
■ Year 2004 (2004/10 – 2005/3)

- Design and analysis of periodical reports and workflow

■ Year 2005 (2005/5 – 2006/3)

- A pilot system as well as further design and analysis

Pilot system (plan)

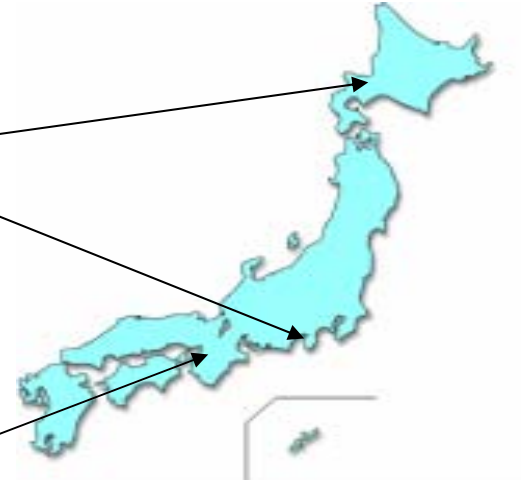


Why use XML?

- The government does not want to use proprietary formats.
- Information reuse
 - At present, local governments input the same information repeatedly for different reports.
 - Local governments input the same information that their other software already know.

Technical members in the XML team

- Prof. Asami and me (Kanagawa)
- IT companies (Hokkaido)
 - Fusion
 - Mediarium
 - Keiwa business
 - Shoubunsha
 - NTT comware Hokkaido
- NEC (Osaka)



Only some of them are XML experts.
No experts of W3C XML Schema



Document analysis

Periodical Reports

- 293 periodical reports for the Ministry of Internal Affairs and Communications.
 - One variation for prefectures
 - Another for municipalities
- Topics of periodical reports
 - Financial information
 - Local government bond
 - Salary of employees
 -

Reports from municipalities and those from prefectures

- Similar structures and similar data
- Some differences are inherent.
 - We keep them.
- Other differences are caused by mistakes or layout constraints.
 - We remove them.

Two reports we chose

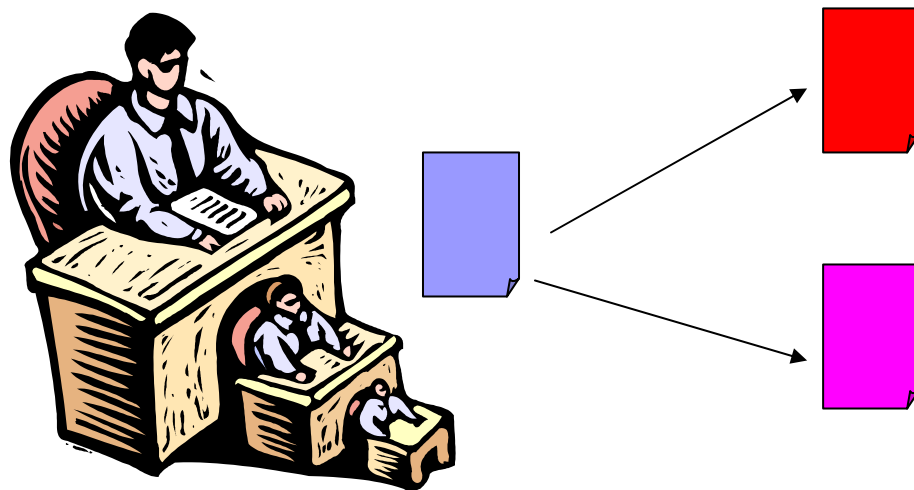
- Dominated by numbers
 - Other periodical reports contain prose as well as numbers.
- Tabular structures
- As usual, we eliminate layout information and focus on structures or semantics.

Duplication of information

- One page for the total
- One page for the total in each area
- Several pages for all local-government-projects

- It does not make sense to create them separately.

Reorganization for avoiding re-keying (plan)



Common components

- Some pieces of information (e.g., metadata) repeatedly appear in several periodical reports.



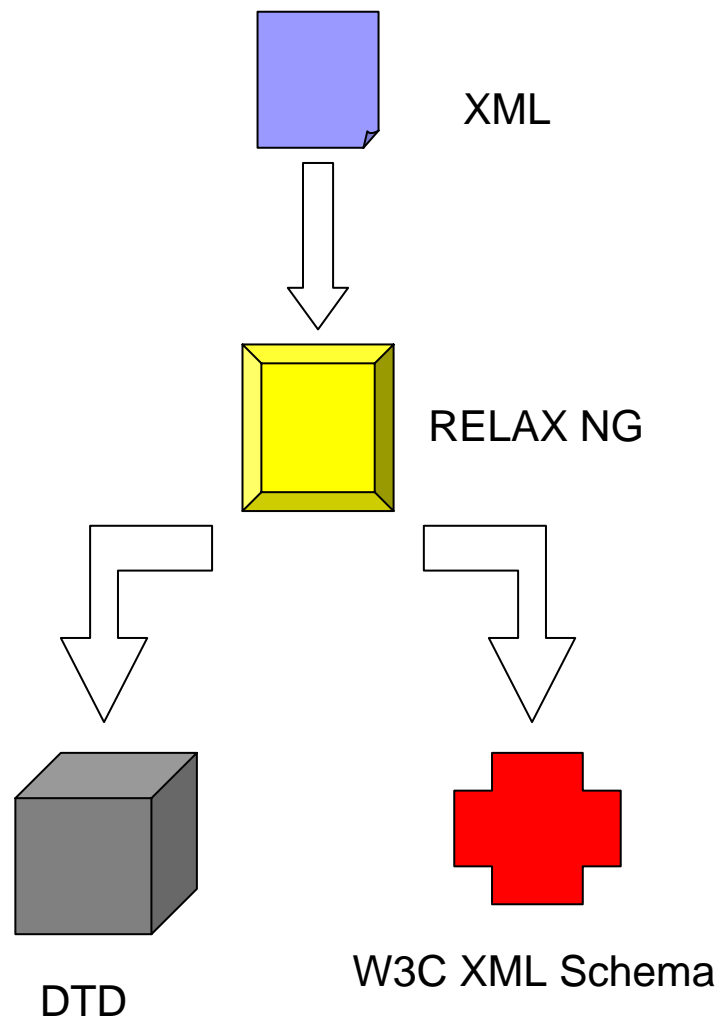
Document design

Schema languages

- DTD
 - A lot of people understand DTDs.
 - W3C XML Schema
 - Supported by Microsoft, IBM, and so forth.
 - RELAX NG (ISO/IEC and JIS)
 - Simple and powerful
 - Japan contributed to its design and implementation.
- We do not want to lose any of them.

Schema authoring

- Step 1: Handcraft XML instances
- Step 2: Generate RELAX NG schemas from XML instances by trang and hand-tune them.
- Step 3: Convert RELAX NG schemas to DTD and W3C XML Schema by trang and then hand-tune them.
 - We also use an in-house tool for slightly changing DTDs.



Lessons learned

- RELAX NG and trang allow inexperienced developers to create WXS schemas easily.
- Trang
 - Good at generating RELAX NG schemas from XML
 - Good at generating WXS schemas from RNG schemas
 - Not so good at generating DTDs from RNG schemas (due to namespaces)

Example

```
element 事業内容 {  
    element 市町村名 { xsd:NCName },  
    element 事業区分 { local_事業区分 },  
    element 制度 { local_制度 }?,  
    element 事業名 { xsd:NCName },  
    ## 要素「当初-追加の別」は、元データ「当初追加の別」のデータを格納する。  
    element 当初-追加の別 { local_当初-追加の別 },  
    element 事業期間 {  
        element 事業開始年度 { 年度-和暦 },  
        element 事業終了年度 { 年度-和暦 }  
    },  
}
```

Common components

メタデータ = `element meta:メタデータ { メタデータ.内容 }`

メタデータ.内容 =

`element meta:西暦年度 { 年度-西暦 },`

`element meta:都道府県名 { 都道府県名.内容 },`

`[doc:documentation [`

`doc:summary ["追加予定"]`

`doc:description ["一部事務組合等コードを追加する。来年度には、消防本部のコードを追加予定。"]]`

`]`

`element meta:地方公共団体コード { 地方公共団体コード },`

メタデータ.内容拡張用

Datatype library

都道府県名を表す文字列 =

“北海道” | “青森県” | “岩手県” | ... | “沖縄県”

地方公共団体コード = `xsd:long` { `minInclusive` = "010000" `maxInclusive` = "479999" }

単位を百万円とし小数点以下一桁までを表す有理数 = `xsd:decimal` { `fractionDigits` = "1" }

単位を百万円とし小数点以下一桁までを表す非負有理数 = `xsd:decimal` { `minInclusive` = "0" `fractionDigits` = "1" }

単位を千円とし小数点以下一桁までを表す有理数 = `xsd:decimal` { `fractionDigits` = "1" }

単位を千円とし小数点以下一桁までを表す非負有理数 = `xsd:decimal` { `minInclusive` = "0" `fractionDigits` = "1" }

...

Schematron

- Integrity constraints
 - Among different pieces of information in a single report
 - E.g., $//a/b + //c/d = //e/f$
 - Among different reports
- It is only Schematron that can capture such integrity constraints.

Example

```
<sch:schema xmlns:sch="http://www.ascc.net/xml/schematron">
  <sch:ns uri="urn:go.jp:xmlns:01234567890:AGNHY202:200503"
  prefix="go"/>
  <sch:pattern name="検証 No.1">
    <sch:rule context="/go:一部事務組合への加入等の状況/go:検算_総合計/go:
    組合加入状況">
      <sch:assert test=". = sum(../*[not(name() = '検算_総合計')//go:組合加入
      状況)">[検証 No.1]は正しくありません。メッセージ: 検算は議会議員公務災害補償
      から農業共済事業の合計でなければならない。 </sch:assert>
    </sch:rule>
  </sch:pattern>
</sch:schema>
```



Schema development environment

Development Environment

- SourceForge
 - CVS server
 - Mgmt of action items
- Eclipse
 - CVS client
- oXygen (XML editor) as an Eclipse plugin
 - trang
- Cygwin (Linux-like environment for Windows)

oXygen

- It is not very expensive.
- It supports RELAX NG, W3C XML Schema, and the DTD.
- It works on Windows, Unix, Mac, and so forth.
- The company responds to my requests in a very timely manner.

In-house commands

- Shell scripts under Cygwin
- Commands for schema documentation
- Commands for in-house schema conventions



Demo



Schema documentation

Annotations in RELAX NG schemas

■ Schema

```
local_事業区分 = “循環型社会形成” | “少子・高齢化対策” | ...[  
  doc:documentation [  
    doc:summary [ "値を指定するデータ型" ]  
    doc:description [  
      “事業区分の小分類の表記を設定する...”  
    ]  
  ]  
]
```

In-house Schema for RELAX NG schemas

- Derived from the standard schema “relaxng.rnc” .
- Mandate an in-house convention for annotations

```
include "relaxng.rnc" inherit = rng {  
  other =  
    (external "relaxng-annotation.rnc" inherit = rng  
    | external "documentation.rnc" inherit = rng)?  
}
```

Automatically-generated HTML documents



Using Japanese characters

Use of Japanese is a must

- Translation to English is often unreadable or impossible.
 - 前年度末公共用地先行取得用地保有面積

Theory and Practice

- The XML recommendation supports I18N very well.
- However, do implementations support I18N really?
 - Non-ASCII data
 - Non-ASCII tag or attribute names
 - Non-ASCII file names

Files containing Japanese characters

- UTF-8

- SourceForge (needs configuration)
- Eclipse CVS (needs configuration)
- oXygen (needs configuration)

- Work very well.

Tag names and attributes names including Japanese characters

- XML parsers
 - Validators
 - oXygen
- Work very well.

Namespace URIs including Japanese characters

- Regretfully, the first W3C recommendation is unclear about non-ASCII URIs.
- Xerces-J (version 2) does *not* support non-ASCII namespace URIs, but MSXML does.
 - We gave up and switched back to US-ASCII URIs.

File names containing Japanese characters

- UTF-8
- We reinstalled the server OS (Linux) for SourceForge using UTF-8 file names.
 - Work well.
 - But Cygwin does not always work.

Lessons learned

- With the exception of non-ASCII namespace names, we can use Japanese safely.
- Configuration is not a simple task.

Conclusion

- RELAX NG allows you to create W3C XML Schema schemas without tears.
- Japanese tag names work.
- Schematron helps.
- The combination of SourceForge, Eclipse, and oXygen is great.

Future works

- User interface for document authoring
 - Open Office
 - Excel
 - XForms
 - Acrobat
 - Xfy
- XML guidelines