Extending the Boundary of Spreadsheet Programming

Lessons Learned from Chinese Government Project

Xingliang Yu, Jing Li, Hua Zhong

Software Institute
Chinese Academy of Science
Outline

I. Chinese Style Spreadsheet
II. Challenges and Solution
III. ESL Language
IV. EUD Effectiveness on SNS
V. Lessons Leaned
Chinese Style Report

- **Country Level**
  - Approx: 200
    - Sector 1
    - Sector 2
    - Sector 3
    - Sector 4
    - Sector 5
    - Sector n

- **Province Level**
  - Total: 34
    - Avg: 89
      - Prov 1
        - Sector 1
        - Sector 2
        - Sector 3
        - Sector 4
        - Sector 5
        - Other Provinces

- **City Level**
  - Total: 333
    - Avg: City 36
      - City 1
        - Sector 1
        - Sector 2
        - Sector 3
        - Sector 4
        - Other Cities

- **County Level**
  - Total: 2862
    - Avg: County 26
      - County 1
        - Sector 1
        - Sector 2
        - Sector 3
        - Other Counties

- **Town Level**
  - Total: 43275
    - Avg: Town 5
      - Town 1
        - Sector 1
        - Sector 2
        - Other Villages

- **Village Level**
  - Total: 691510
    - Avg: Village 5
      - Village 1
        - Sector 1
        - Other Villages

Reference: People’s Daily, May 30th, 2005
Chinese Style Report

Long Tail Effect in eGovernment:
- thousands of threads from upper
- all pass through the eye of just one needle
Chinese Style Report

- Report = Spreadsheet
  - Most of reports are in Spreadsheet format
  - User are more familiar with EXCEL than other table tools

- Format
  - Rigorous
    - Anything of Tables from upper level agency, except the data, must not be modified, including:
      - text and it's format
      - line and space
      - Paper size and binding style
  - Complex
    - No table-model in research can express all kind of table we seen
  - Flexible
    - Presentation format can change depend on the content of data
Chinese Style

- **Scale**
  - Wide Business Range
  - Deep Hierarchical
  - Large number of report
  - Massive data

- **Usage**
  - Output
    - To upper level agencies
    - To upper positions
    - To self-using report
    - To lower level documents
  - Input
    - From other agencies
    - From other self generated tables
Chinese Style

- **Lifetime**
  - Instant
  - Several Days
  - Several Months
  - Several Years
  - Permanent
    - National Archives

Chinese Spreadsheet Lifetime vs. Number
Chinese Style

- Content
  - Multiple Type
    - Character
    - Number
    - Date & Time
    - Graph
  - Unequal Semantic
    - Compared with database design, semantic arrangement in table are not so cohesive, or somewhat random
    - No strict semantic design method or rules available
<table>
<thead>
<tr>
<th>School code:</th>
<th>School name:</th>
<th>Unit:</th>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preliminary school</td>
<td>Middle school</td>
<td>High school</td>
<td></td>
</tr>
<tr>
<td>In-school student</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In the in-school student</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Re-entry</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Girl</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minority</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Border</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>local registered residence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In above 6, but is educational residence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Treated as local residence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>last academic year</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fail to go up to the next grade</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Girls in 9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transient Student</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In all above</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>from Hongkong, Macao</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>from Taiwan</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foreign</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Girls</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>responder</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>auditor</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Challenges

- Expressiveness of programming language
  - End-user enabled
  - Better form of array notation and reference
  - Breaking the limitation of reference in modern Spreadsheet

- Usability
  - Not too highly abstract syntax

- Effectiveness
  - Prevent user's error while programming
  - Program need test to raise quality
Solution

- ESL Language (EUD-Enabled Spreadsheet Language)
  - Usability
  - Expressiveness
  - Presentation
  - Integration

- EUD effectiveness
  - A specific research on pre-release failure in web-plugins development on SNS
  - Release-waiting farm technology
ESL Language

- **Definition**
  - **Template**
    - $T=(ID, CellMatix, Formulae, ParmList, Context, Grid)$
  - **Sheet**
    - $S=(ID, Attr, CellMatrix, Grid)$
  - **Parser**
    - $Parser: S_{ir}=Parser(T_i, Parm_{ir})$, $i=1...n$
      - A set of Template: $Dt=\{T_1, ..., T_n\}$
      - A set of Sheet: $Ds=\{S_{11}, ...S_{nm}\}$
      - A Parameter List $Parm_{ir}$

- **3 Mode**
  - **Computation**
  - **Validation**
  - **Presentation**
ESL Language

- Rang reference
  - Expand R1C1 Style
    - [T*R*C*]
    - No current cell, no relative addressing
  - Syntax
    - `<Range> ::= IntegerLiteral..'IntegerLiteral`\n      `|IntegerLiteral`\n      `|'*'`\n      `|'..'IntegerLiteral`\n      `|'('<MultNumID>')'`\n      `|IntegerLiteral`\n      `|ID..'ID`\n      `|Macro`\n      `|<EmbeddedSQL>`
ESL Language

- Formula not bound to cell
  - `<AssignFormula>::= <Ref> '=' <Expression>`

- Include Assertion in program
  - `<Formula>::= <AssignFormula>`
  - | `<LogicFormula>`
  - | `<Formula>','<MacroDef>`

- Macro support
  - `<Range>::= Macro`
  - `<Value>::= Macro`

- Embedded SQL
  - `<Range>::= <EmbeddedSQL>`
  - `<Value>::= Macro`
ESL Language

- A sample of ESL program
  - \([T3001R1\ldots15C1\ldots16] = \text{SUM}([T1\ldots350R1\ldots15C1\ldots16], 'T')\)
  - \([T3002R1\ldots15C1\ldots6] = \text{[T3001R1\ldots15C2\ldots7]} /\text{[T3001R1\ldots15C1]}\)
  - \([T3002R1\ldots15C7\ldots10] = \text{[T3001R1\ldots15C9\ldots12]} /\text{[T3001R1\ldots15C8]}\)
  - \([T3002R1\ldots15C11\ldots13] = \text{[T3001R1\ldots15C14\ldots16]} /\text{[T3001R1\ldots15C13]}\)
Social Networking Site (SNS)

Facebook
- Growth with participation expanding at rates 20% a month[16]
- Encourage end users to develop web plug-ins and Shares these plug-ins across the whole site [6]

Xiaonei.com
- Friend Trade: a web plug-in developed by five college students, which has been installed by more than three millions of Chinese users in 42 days[7]
The difficulty of research on EUD effectiveness
- Size of programming task is small
- Distribution of End User is disperse
- Knowledge needed for communication is scars

EUD of Web Plug-ins on SNS
- No professional background knowledge
- No strict discipline
- No standard process
- Developed by one, used by many
RWF Technology (1/3)

- Release Waiting Farm

1. Requirement Analysis
2. Design
3. Construction
4. Use-it-myself Testing
5. Release

Starting Point

- Mini World
- Brainstorm
- Invited Senior
- Web Toolset

Release-Waiting Farm
RWF Technology (1/3)

- As a research on formalizing process of EUD, we present RWF technology at ICSP 2009,
  - Presented at 17th May, 2009.
  - Also an event collocated with ICSE.
After a 5-months experiment, we found:
- the farm is big enough to test plug-ins
- brainstorm and invited senior methods work effectively
- the farm costs low when compared with the main server, and it is worthy to build a farm

RWF technology reuse in ESL
- Provide RWF support in ESL environment

Formalize the development process of end user, is possible, and can raise plug-in quality
Lessons Learned

- Use existing mature spreadsheet applications.
  - more than 99% Chinese governmental projects, we found that users tend to use MS Office, or WPS (a Chinese local office product).

- Allow arbitrary editing.
  - It's not true that report is a kind of read-only, printable spreadsheet.

- build and maintain dictionaries.
  - Provide dictionaries that included professional words and definition. It is significantly important to attach a list of the words that may lead to confusion.

- Formalize programming process of EUD
  - end-user who writes programs needs to know the fundamental disciplines of software engineering.
Thank you very much!