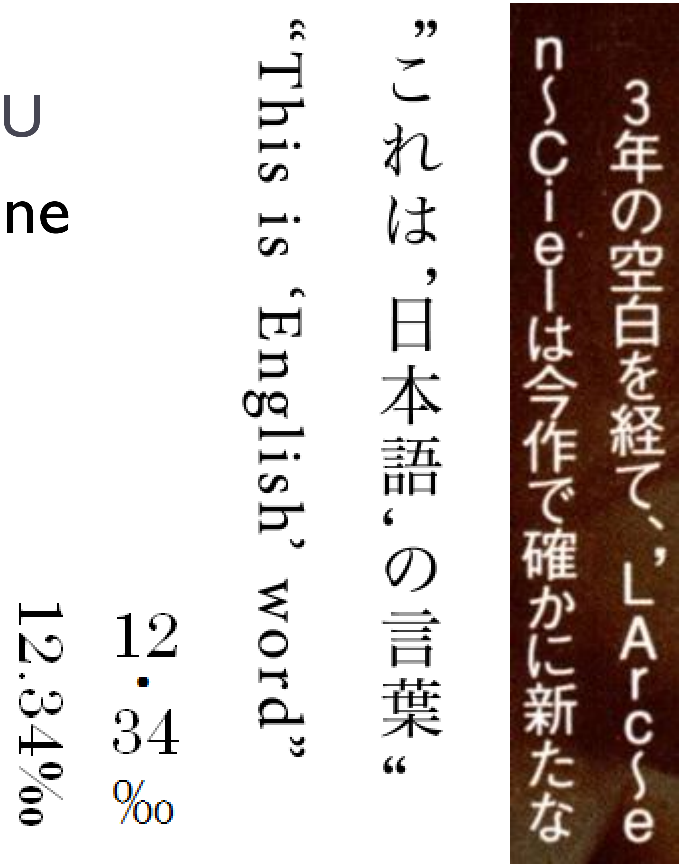
L2/12-388

UTR#50 Conflict Resolution Proposals

Koji Ishii

Why Conflict?

- Often determined by if the script is East Asian or not
  - あ → Japanese=U
  - A → Latin=R
  - Ａ (FULL WIDTH A) → Japanese(?)=U
- Unification makes hard to detremine
  - U+2019 ’
  - U+2030 ‰
    - ％ has FULL WIDTH but ‰ is unified
  - U+2113 ℓ
  - U+2126 Ω



#1: based on **the most** common use

- May requires exhaustive research
- Hard to answer to questions such as:
  - How do you determine "the most common"?
  - I believe this is more common
  - U is more common in literature, but R is more in magazines
  - This was more common until 10 years ago, so more number of books exist
  - The number of texts, or the number of readers?
  - Publications or office documents?
  - Common vertical text, or common text regardless of flows?

#2: based on **one of** common use…

- Can avoid exhaustive research
- Is more stable over time
- Easier to justify when UTC resolved by voting etc.
- How should we choose the one?
  - Helps justifying, but still the same questions apply for us to make consensus

#3: add more FULL WIDTH code points

- Can detect “if East Asian or not” solely by code point
- Takes long to add to Unicode
- How many do we need to add?

#4: control code or variations

- User doesn’t want to enter such code
  - Apps can insert automatically
- States of variations?
  - Control code like LRE/RLE/PDF/etc.
    - State is not favorable
      - □ Bi-di has states
  - Extend IVS to symbols/punctuation/letters
- Orientation code or script code?
- Takes long to add to Unicode
  - Can use existing VS?

#5: Context-based orientation

- Can orient correctly on common cases
- Can satisfy both parties
  - Probably the only way to make both win
- Cons
  - Can be complex and ambiguous; e.g., ‘98’ and ’98
  - Whether to match outer or inner?
  - Requires a lot of testing and improves
    - Can change over time; e.g., ’98 isn’t common after 2000
- Is this “the stable default”?
  - Good app feature, apps can insert tags automatically
  - If app feature, user can correct as s/he types
  - If app feature, easy to improve over time

#6: common use **in Japanese context**

- Makes most hard case resolvable by common sense
- “Character A is never used in Japanese context”
  - Can require at least one commercial use
- “There are many Latin-mixed text in the wild”
  - Higher-level protocols can tag Latin text
    - :root { text-orientation: sideways; }
    - [lang|=ja] , [lang|=ko], [lang|=zh] { text-orientation: mixed-right; }
    - Kindle Japan requires <span lang="en"> to rotate quotes
  - Apps can insert tags automatically
    - Word automatically applies the property by keyboard + lang detection
- “Common use in Latin context within Japanese” is theoretically possible option but we probably don’t want?

Two more things…

Priorities among multiple criteria

1. Full-width=U, has full-width counterpart=R
2. Common use in Japanese context
3. Common use in other East Asian context
4. Similarity to existing characters
5. Common use in Latin context within Japanese
6. Common use in Latin context within East Asian scripts
7. Unicode consistencies (block, general category, etc.)

- "Common" requires at least one commercial use

Change already-resolved data?

- Change all resolved data to match to the new scope?
  - There will be inconsistencies without doing this
  - Most vendors/publishers do not want major changes any longer
- Recommends not, due to the impact