



未踏ソフトウェア創造事業 高田PM中間報告会

採択テーマ 「組み込み向け日本語入力環境の構築」

太田一樹(東京大学 理科一類 学生)



- 自己紹介
- 本テーマの背景事情
- 基礎知識
- UNIX上の入力環境事情
- uimとは？
- プログラミング言語Schemeの概要
- 未踏前期期間での開発内容
- 未踏後期期間での開発予定

- 名前
 - 太田一樹
- 出身
 - 東大阪(瓢箪山)
- 所属
 - 東京大学理科一類
 - 理学部情報科進学 予定
 - 日本KDEユーザー会開発コーディネーター
- プログラミング歴
 - 約3 - 4年

That's Me



- 基礎知識編
- UNIX上の入力環境事情
- uimとは？
- プログラミング言語Schemeの紹介と概要



• UNIX

- OSの一種。WindowsやMacOSと同列のもの。ただし、UNIXと言う名前のOSがあるわけではなく、言うなれば「UNIXというジャンル」を指す。UNIXの一種にLinuxやBSDというものが存在する。

• GUI

- Graphical User Interfaceの略。マウス等でぽちぽちとコンピューターを操る方法一般を指す。



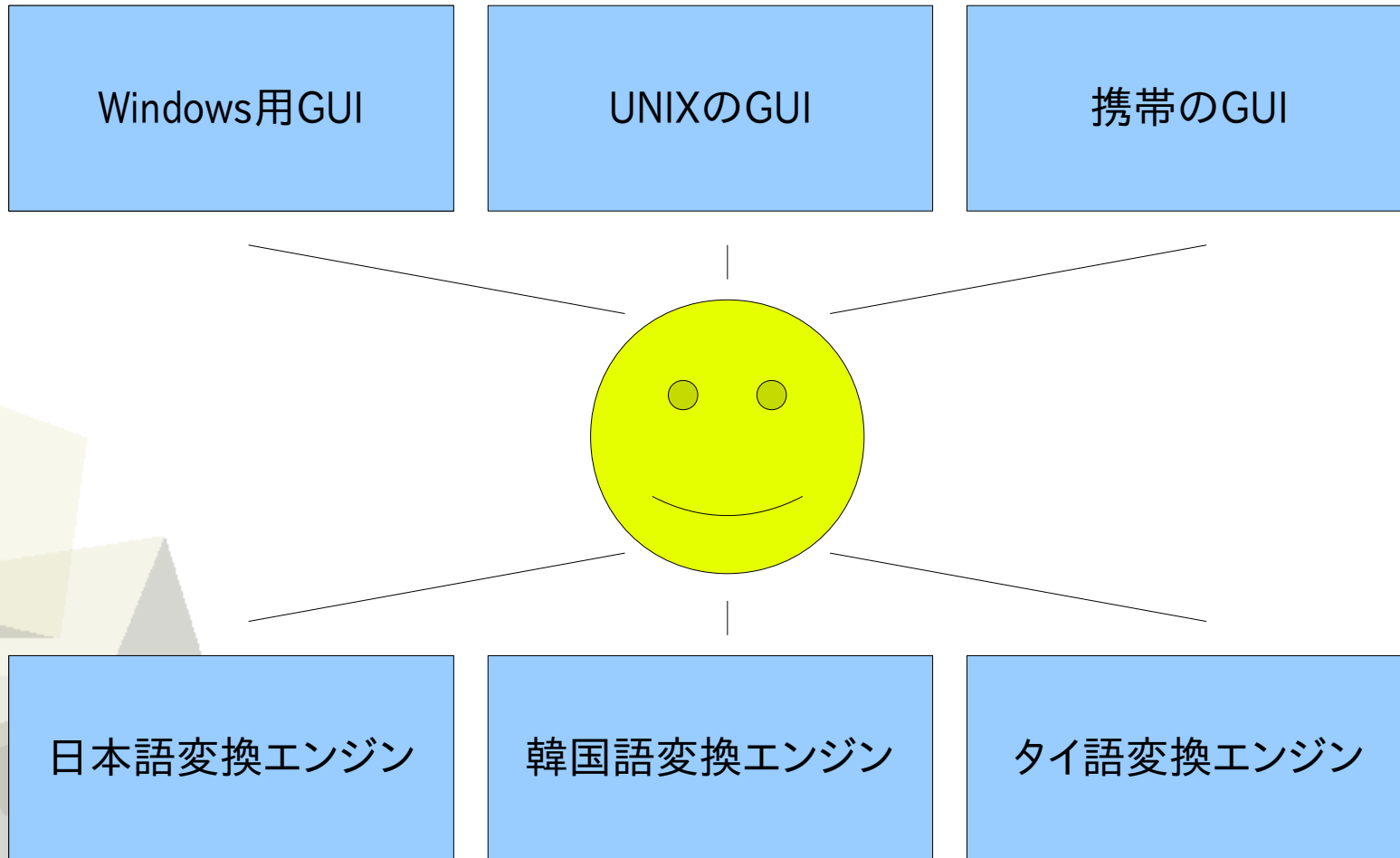
- 過去の遺産
- 乱立するGUIツールキット
- 入力フレームワークの乱立
- 様々な言語用の入力モジュール
- 新しく標準化の動き

良いモノから悪いモノまで
混沌としている状態。

That's Chaos

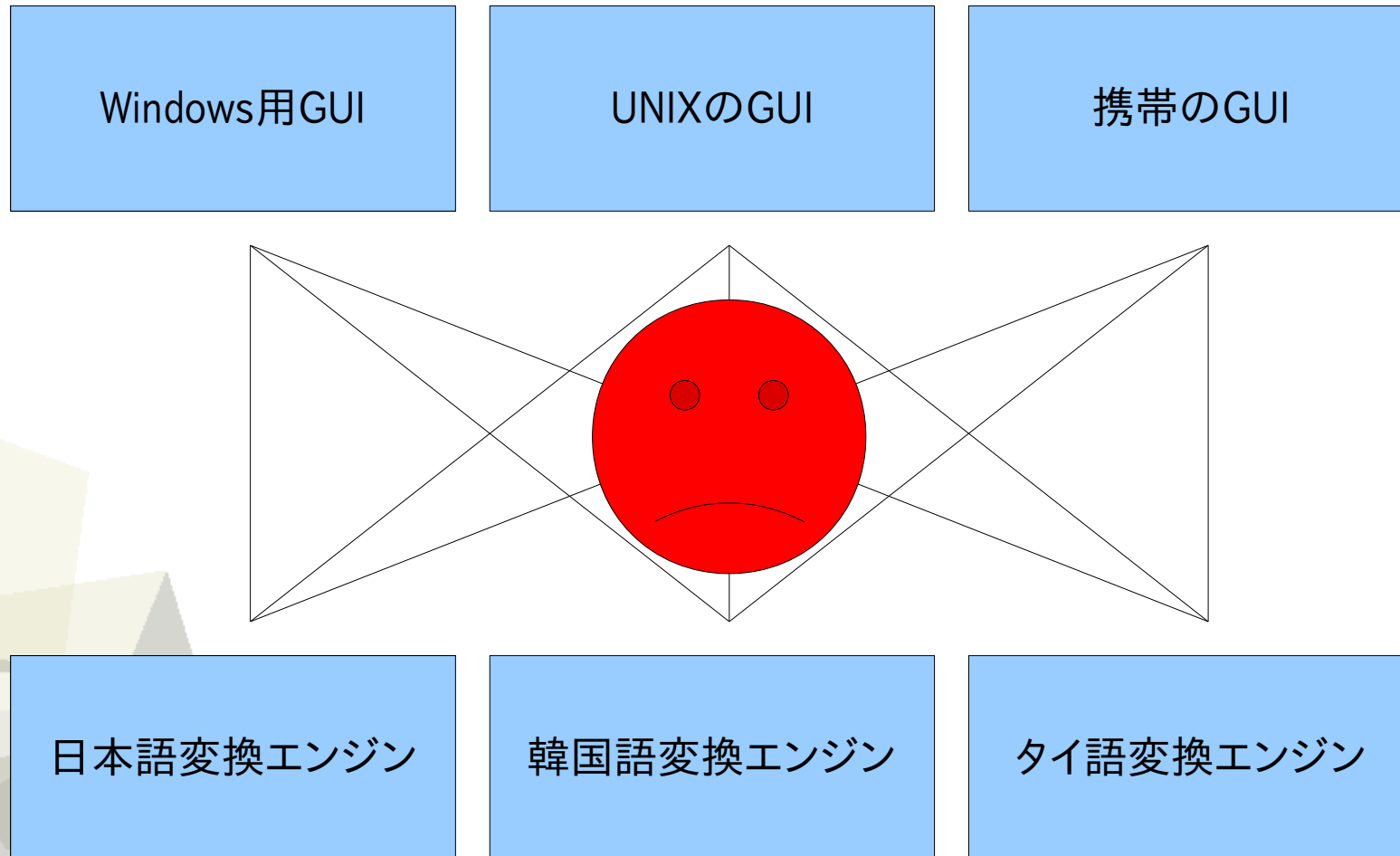


2002年より開発が始まった入力用フレームワーク。
GUIと変換エンジンの中間層に位置する。





uimが無い場合の概念図



GUIと変換エンジンの間をうまく取り持つ仕組みが無いと、変換エンジンとGUIをダイレクトに結びつける必要がある。

しかし、これではGUIがn個、変換エンジンがm個有った場合、 $n * m$ 個のブリッジプログラムを作成する必要がある。

逆に、uimが有る場合は $n+m$ 個のブリッジプログラムで済み、メンテナンスコスト等を下げる事が出来る上に再利用性が上がる。

$$\underline{N * M \geq N + M \quad (N > 2 \text{ and } M > 1)}$$

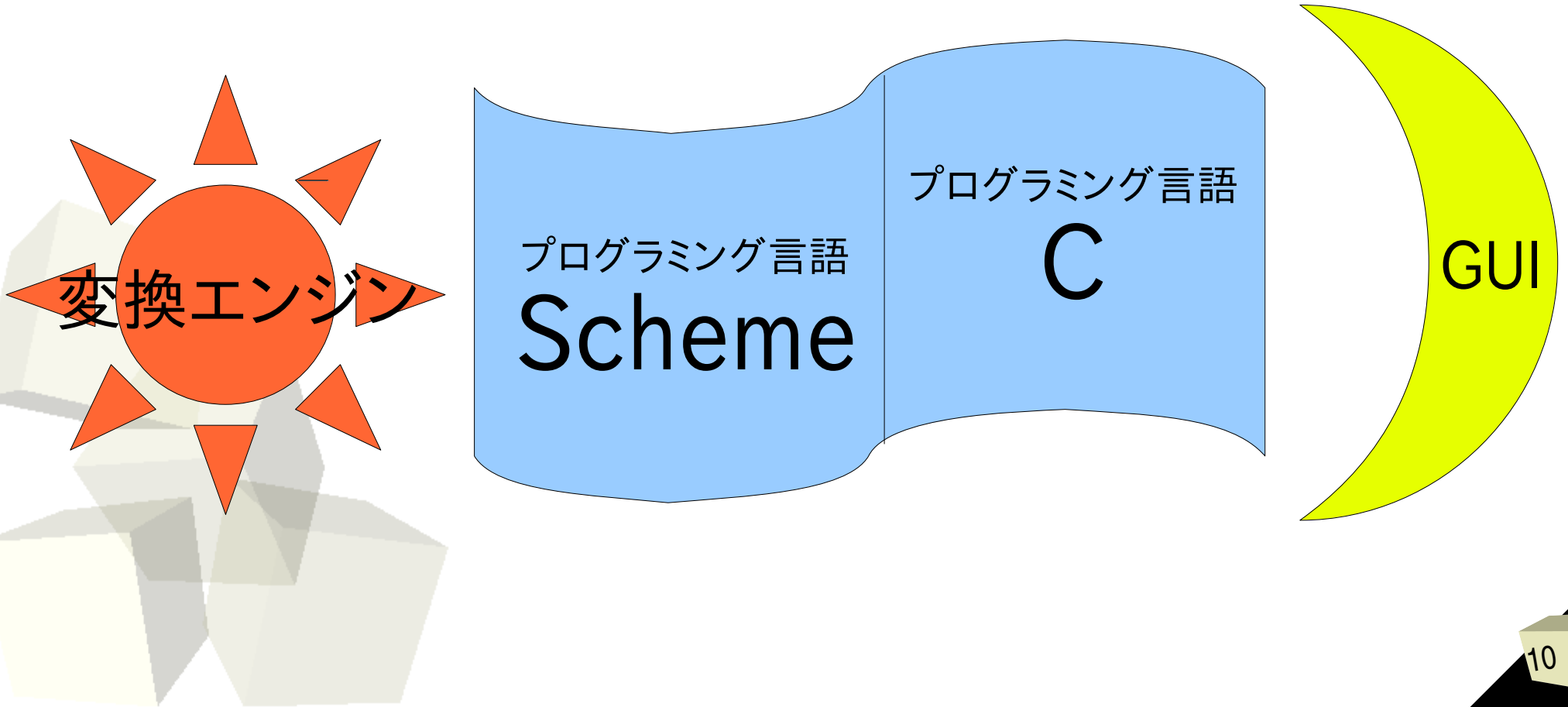
That's Bad

without uim



uimの中身

2つのプログラミング言語を用いて実装されている





- Lisp言語の方言
- MITや京大、東大等で教育用に用いられている
- コンパイルをしないので非常に生産性が高く、また言語が小さくまとまっているので覚え易い。
- 「Hackers and Painters(邦訳:ハッカーと画家)」で注目を浴びる。この本によると、Yahoo! Store等の構築にもLisp系の言語が用いられていたらしい。

-> 例を見てみましょう。



Scheme道場

; 足し算

(+ 1 2 3) => 6

; 引き算

(- 6 3 2) => 1

; nの階乗を計算する関数

(define (factorial n)

(if (= n 1)

1

(* n (factorial (- n 1))))



かっこだらけ。

(That's Scheme)





未踏期間前期での開発内容(1)

- uimを組み込みシステム向けに改良する上でボトルネックとなっているのは、内蔵のScheme処理系と判断した。
- 言語仕様に反した動作 -> 変なScheme
- メモリ効率が悪い
- ライブラリ関数が少ない

そこで、未踏期間前期ではSchemeの言語処理系を刷新し、組み込みシステム用途向けの処理系の作成・そしてuimへの取り込みに取り組んだ。



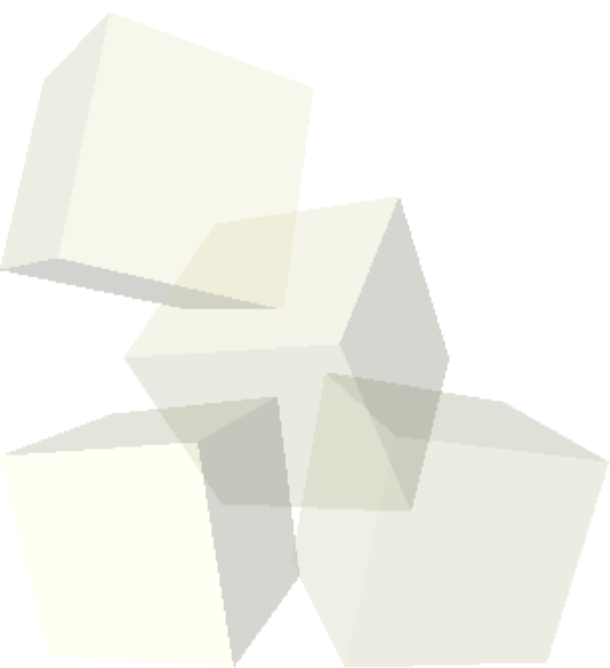
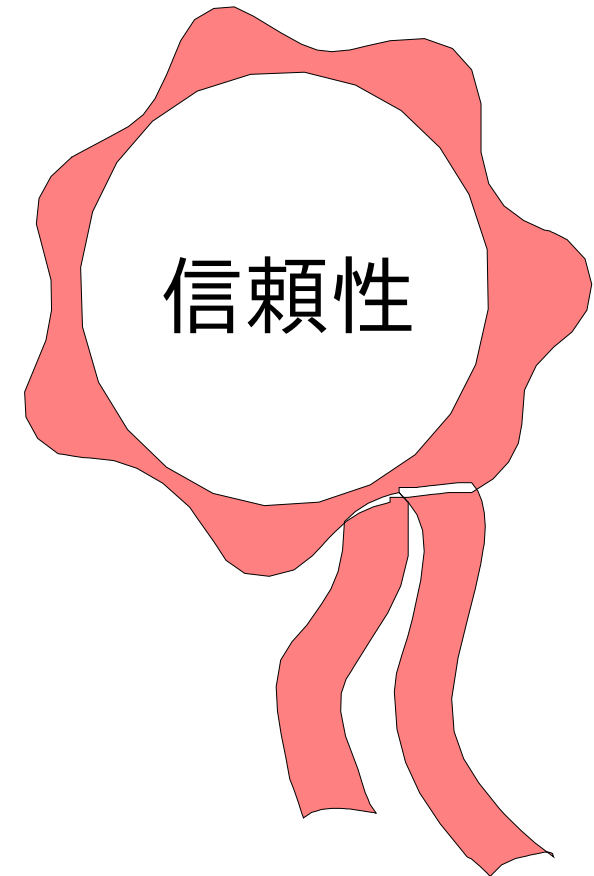
現在までの処理系の開発成果

- R5RS(Schemeの言語仕様)準拠
- 以下のSRFI(Schemeのライブラリ仕様)をサポート
 - SRFI 2(and-let*)
 - SRFI 6(basic string ports)
 - SRFI 8(receive)
 - SRFI 23(error reporting mechanism)
 - SRFI 34(exception handling for programs)
 - SRFI 38(external representation for data with shared structure)
 - SRFI 60(integer as bits)
- メモリ効率の改善
- マルチバイト文字対応(Shift-JIS, EUC-JP, ISO2022-JP, UTF-8)
- SIOD(Scheme In One Defun)互換モード



信頼性確保の為に、UnitTest機構を導入

現在までに約1000ケースのテストを書き、変更の度に動作を確認
新機能追加の度にテストケースを追加





言語として必要な一通りの機能は揃えた

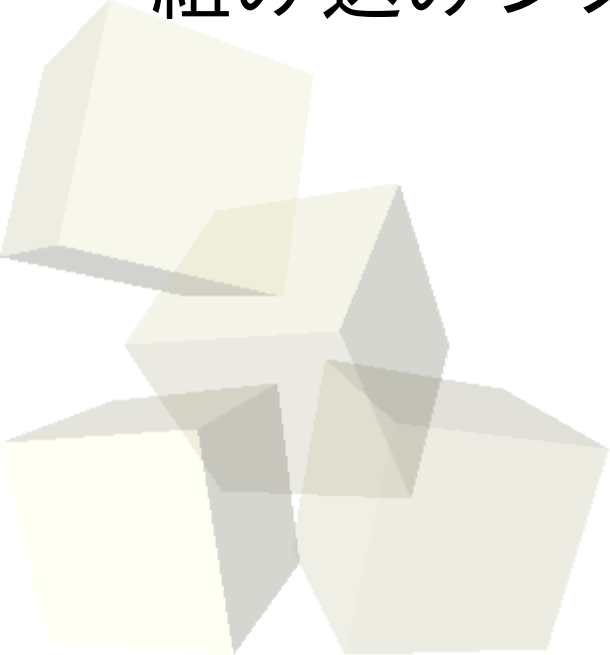
- TODO

- 細かいバグつぶし
- 処理系の高速化(現状は昔の処理系と同じぐらい)
- 他のプロジェクトへのマージを目指す
 - gimp(画像処理ソフト)
 - festival(音声発声ソフト)
 - ...

That's I've Done



- 各種プラットフォームへの移植
 - ◆ 組み込みLinux
 - ◆ Tron
 - ◆ 携帯電話?
 - SymbianOS
 - BREW
 - ◆ PSP?
- 組み込みシステムに取り組む際のノウハウの文書化



以上で僕の発表を終わります。
御清聴有難う御座いました。

That's All!

Thanks!