



やめといたほうが良い話



SHIORIを作ろうと思ったときに
ぶち当たる壁のあれこれ
ルールエンジンの紹介と活用方法

講師：どっとステーション駅長

※注意※

この資料は印刷しないほうがいいよ！

うかべん

同的ソフトウェア勉強会関西支部



結論と概要



- SHIORI作るのには 理系言語系マゾ以外はやめとけ！
 - 以下、一般人には笑い話、一部の人には技術的な話となります。
- 普通にしゃべると地味な題材で、どれだけハツタリできるか、実践例を交えてだらだらとトーク。
- ルールエンジンでどこいつAIを作ってみるかもしれない。

うかべん

同的ソフトウェア勉強会関西支部



SHIORIとは



- うかがかプラットフォームにおいて、ゴーストの会話を制御するエンジン。
- プラットフォームからの要求(SHIORI APIへのアクセス)に応じて適切な応答（主に会話）を返す。
- 「SHIORIを作る」とは、「SHIORI API」を実装するDLLを作成するということ。
- ふつーは「使う」ものであって、「作る」ものではありません。

うかがべん

同的ソフトウェア勉強会関西支部



魂の叫び



- **でもそんなの関係ねえ！
俺は作るんだ！**

- (ここから1時間ほど熱く語るが24ページまで省略)

この辺のウダウダ話とかはこの資料を後でダウンロードして生暖かく笑ってください。

うかべん

同的ソフトウェア勉強会関西支部



SHIORIを作りたい「動機」



- 現状のSHIORIに満足できない
 - SAORIで実現できる場合もあり。
 - SHIORIプラットフォームとして作るなら自作。
 - 汎用的なSHIORIを作りたい人とか。
- 面白そうな理論を実践してみたい
 - AI そのものを実装するなら自作しかない。
 - 自作ゴースト特化型SHIORI。

うかべん

同的ソフトウェア勉強会関西支部



SHIORIを作りたい「動機」 その2



- や、なんとなく
 - お好きにどうぞ
- 何れにせよ、任意のプログラム言語が使えるため、既存SHIORIを使う場合に比べて自分のやりたいことに関しての制約が非常に低くなる。
但し、当然ながら必要とされる開発技術も高度なものが要求される。

うかべん

同的ソフトウェア勉強会関西支部



その言語、使えますか？



- 開発のベースとなるプログラム言語や処理エンジンを選定するときの最重要課題。
- 第1目的が実現できるという理由だけで選んだ時に、後ではまる要素。

うかべん

同的ソフトウェア勉強会関西支部



日本語、通りますか？



- 世の中の多くのプログラム言語は海外で作られている
- アルファベットが使えるれば良い人たちは、ワイドコード圏の都合をスルーしがち。
- 特に技術的に先進的なアプローチを採用している言語（そしてSHIORI作りたい人が採用したい言語）ほど、日本語で泣くことが多かったりする。
- **日本語問題を侮るな！ここで調査に手を抜くと越えられない壁が発生するぞ！**

うかべん

同的ソフトウェア勉強会関西支部



(補講)

もじもじくんの相性問題



- 文字型は1byteオンリー
 - 何も考えず1byte。
 - 更に0x80以上のコードで落ちる→論外
 - OSのANSIコードに任せる。
→出来上がったSHIORIが日本語環境専用。

うかべん

同的ソフトウェア勉強会関西支部



もじもじくんの相性問題 その2



- UTF-8に関して考慮した関数がある。但し文字列としてはbyte列で扱う。
 - 自作ゴースト専用であれば大丈夫かも。
 - 汎用に作ろうとすると超えられない壁が発生する。
- I/O処理などを行わせようとしたときにややこしい事態が発生することが多い。
 - ディレクトリ名に日本語が入っていると落ちたり。

うかべん

同的ソフトウェア勉強会関西支部



もじもじくんの相性問題 その3



- 文字型は1byteとunicode両方サポート
 - 一見、一番いいように思うが……。
 - 2種類サポートすることで言語設計に無理が生じていることが多い。
 - I/O処理や周辺ライブラリなどでunicode文字の問題が発生したりする。

うかべん

同的ソフトウェア勉強会関西支部



もじもじくんの相性問題 その4



- 文字型はunicodeオンリー
 - 通常最もトラブルが少ない。
 - I/O処理などに関しても通常はunicode APIを使うため問題になることが少ない。
- 汎用SHIORIなどで辞書の構文解析を行う必要があるなら、極力このタイプを選択すべき。

うかべん

同的ソフトウェア勉強会関西支部



構文解析と字句解析



- 構文解析とは
 - 一定のルールに従った文章を、プログラムが解釈できるデータ構造に変換する処理。
 - 構文解析が終わったデータ構造に対してプログラムは処理を実行する。
- 文章の構文解析を行うために
 - 文章を直接構文解析するのは難しい
 - 扱いやすいように前処理を行う→字句解析

うかべん

同的ソフトウェア勉強会関西支部



構文解析と字句解析

その2



- 字句解析とは
 - 構文解析において、最初に行う作業。
 - 文字列を構文解析しやすいように、小さな基本要素（トークン）に切り分けること。
 - 日本語なら、文章を「動詞」「形容詞」「句読点」などに分解する行為。
- 構文解析と字句解析を区別せずに一緒に扱う考え方もある。

うかべん

同的ソフトウェア勉強会関西支部



構文解析/字句解析と 開発ツール



- 構文解析や字句解析を行うためには、文字列を処理するプログラムを作成する必要がある。
 - 手組みで作ることも可能。
- 構文解析/字句解析のルールを与えることで、プログラムを自動生成してくれる開発ツールがあります。
- 自動生成は理系マゾのロマン！
- プログラム言語によってはツールが充実していたりする。言語選定のときに考慮しよう。

うかべん

同的ソフトウェア勉強会関西支部



字句解析にお勧め、「re2c」



- 「日本語文字列/unicode」を強く意識する必要があるのは字句解析のみ。
 - 適切な字句解析を行えば、構文解析以降の処理で日本語を意識する必要は薄れます。
- 字句解析処理は結構面倒なわりに報われない。
- あれこれ言いません。unicodeの字句解析するなら「re2c」使いなさい！

うかべん

同的ソフトウェア勉強会関西支部



re2cの特徴



- 登録した正規表現のうち、どれが一致するのかを調べるプログラム（字句解析機）を自動生成します。
 - 16bit文字/32bit文字にも対応します。
 - 生成結果はC言語の、ひたすら文字列をif・switch・gotoで比較しまくる1関数。
- 生成された関数は非常に高速。
- 生成コードが単純なため、文字列置換でC#で使えるコードに出来る。

うかべん

同的ソフトウェア勉強会関西支部



re2cの特徴 その2



- 他のプログラム言語を使う場合でも、ここだけC言語のライブラリとして作るのがお勧め。
- 小さな入力に対して妙に大きいプログラムが出来上がるので、工数稼ぎたい社会人にもお勧め！

うかべん

同的ソフトウェア勉強会関西支部



構文解析にお勧め、「caper」



- 字句解析ツールで日本語の迷宮に迷い込むことが多い割りに、構文解析用ツールは世の中に結構多い。
- 今回は「caper」というツールを紹介してみます。
 - C++/Java/C#/Javascriptで使えます。
 - 開発が日本人なのでマニュアルが日本語(^^;
 - 素のBNF限定です。
 - これも工数稼ぎに使えます。自作と言い張れます。

うかべん

同的ソフトウェア勉強会関西支部



(私見なので補講) 今風のSHIORI設計



- 文法の設計方針
 - 会話記述は簡便に → 里々にインスパイヤ？
 - 単語辞書・会話辞書で文法を変える
 - 会話辞書は会話に特化した文法を検討すべき。
 - 単語辞書も然り。
 - 辞書設計にプログラムは含めない。
 - HTMLとスクリプトの関係。

うかべん

同的ソフトウェア勉強会関西支部



(補講)

今風のSHIORI設計



- SHIORIエンジン実装
 - エンジン実装は要、開発者の趣味に走れ！
- スクリプト言語は借り物で
 - 今時世の中溢れてる → 1から作る必要無し
 - 小さな実装のものが望ましい。
- 独自辞書言語 + 辞書エンジン + 借り物スクリプト
これが多分今風現実解。

うかべん

同的ソフトウェア勉強会関西支部



(さらに補講) SHIORIに潜む罠



- SHIORI APIではまるどころ
 - load(loaddir)で渡される文字列はANSIエンコード
 - 文字列がHGLOBALなので解放忘れに注意
- その他のはまりどころ
 - カレントディレクトリは絶対さわるな！
 - カレントディレクトリは「プロセス」で共有されるため、SHIORI側で弄ると大変なことに
 - その他、環境系変数は他で弄られる事を考慮する事

うかべん

同的ソフトウェア勉強会関西支部



越えられない壁



- ここでテーマを変えます。
- さりげなく、開発フェーズにおける越えられない壁をスルーしますが気にしない方針。

うかべん

同的ソフトウェア勉強会関西支部



どこいつ辞書、 どうやって管理しますか？



- SHIORIのキモは、辞書管理方法にある
 - 辞書からどのように会話を組み立てるか
 - 実用的な速度域で会話の意外性を発揮できるか
- どこいつA Iの実現手法としての「ルールエンジン」を検証します

うかべん

同的ソフトウェア勉強会関西支部



神のみぞ知るセカイ



- 世の中の出来事の順番は決まっていない。
 - ある事柄が起こったときにやるべきこと、続けて起こる事柄は、それぞれ個別に存在する。
 - フラグの発生が別のフラグを引き起こす連鎖反応を1本のプログラムで記述するのは至難の業。
- 神様は、フラグとイベントで世界を知る！
 - 個別の「フラグ」と、「フラグによって引き起こされること」だけを列挙しよう。→**ルールの作成**

うかべん

同的ソフトウェア勉強会関西支部



軽く用語説明



- ルールって？
- ルールをまとめた物は、なんですか？
- →ルールに存在する、2つの要素

うかべん

同的ソフトウェア勉強会関西支部

Left Hand Side



条件。フラグ。

例)

HANDに、触られる。

Right Hand Side

行動。

イベント。



例)

叫び声をあげる。

Left Hand Side

Right Hand Side



条件。フラグ。

例)
HANDに、触られる。

行動。
イベント。



例)
叫び声をあげる。

Rule 1 触られた！

Rule

Rule

Rule

Left Hand Side

Right Hand Side



条件。フラグ。

例)
HANDに、触られる。



行動。
イベント。

例)
叫び声をあげる。

Rule 1 触られた!

Rule

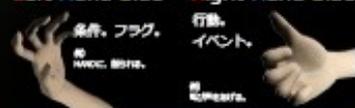
Rule

Rule

Rule

Rule

Left Hand Side Right Hand Side



Rule 1 触られた!

Knowledge base

(知識ベース)

すべてのルール^の集合。
知りうる世界のすべて。



Knowledge base

(知識ベース)

すべてのルールの集合。
知りうる世界のすべて。



軽く用語説明



- 左手側(Left Hand Side、LHS) →条件、フラグ
- 右手側(Right Hand Side、RHS) →行動、イベント
- ルール(rule)
 - ある事象が成り立つときに(LHS)、その事象に連動して発生する事象(RHS)を示したもの
- 知識ベース(knowledge base・ナレッジベース)
 - 全てのルールの集合。知りうる世界の全て。

うかべん

同的ソフトウェア勉強会関西支部



用語説明 その2

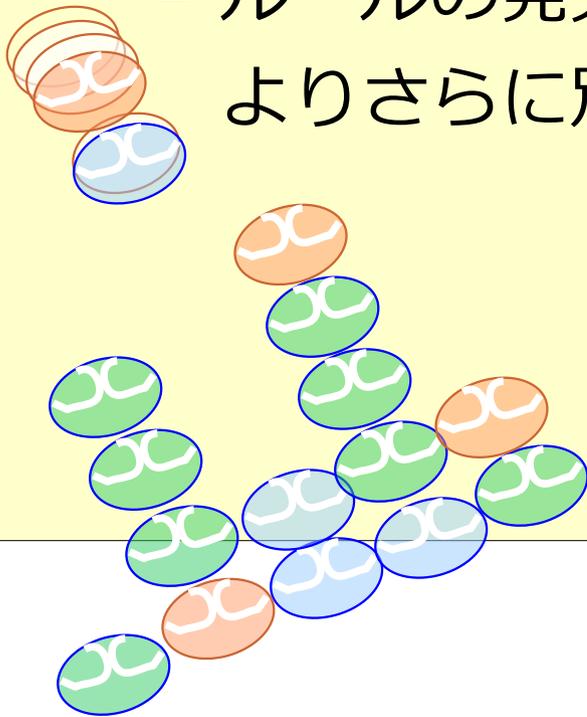


- 発火

- ルールのフラグが成立し、イベントが発動すること

- 連鎖

- ルールの発火により新たなフラグが設定され、それによりさらに別のルールが発火すること



うかべん

同的ソフトウェア勉強会関西支部



ルールを記述してみる (味付けはlisp風味)



- Rule1 ボールを投げたらカーブがすっぽぬけ

```
(defrule 甘めのストレート  
  (ピッチャーが 甘めのストレートを投げる)
```

=>

```
(assert (バッターが ライトにヒットを打つ)) )
```

- Rule2 ランナー走る

```
(defrule ランナー走る  
  (and (バッターが ライトにヒットを打つ)  
        (ランナーが 1 塁に居る))
```

=>

```
(assert (ランナーが 3 塁を狙う)) )
```

うかべん

同的ソフトウェア勉強会関西支部



ルール記述の続き



- Rule3 一方守備はイチローだった

```
(defrule イチロー投げる
  (and (バッターが ライトにヒットを打つ)
        (ランナーが 3塁を狙う)
        (イチローが ライトを守っている) )
```

=>

```
(assert (イチローが 3塁に投げる)) )
```

- Rule4 結論！

```
(defrule 結論
  (and (ランナーが 3塁を狙う)
        (イチローが 3塁に投げる) )
```

=>

```
(assert (レーザービームで 人類滅亡!)) )
```

うかべん

同的ソフトウェア勉強会関西支部



世界のフラグは大混乱



- この知識ベースで地球が滅亡するフラグ(**assert**)
 - (assert (ランナーが 1 塁に居る))
 - (assert (イチローが ライトを守っている))
 - (assert (ピッチャーが 甘めのストレートを投げる))
- このまま情報を管理できたら、楽? でも……
 - 普通にIF THENプログラム組むのはとっても大変!
 - 後でフラグが増えたりしたら、プログラム大破綻!
- これをそのまま実行できるエンジンがあります

うかべん

同的ソフトウェア勉強会関西支部



ルールエンジンの登場



- 知識ベースに含まれる全てのLHS(発動条件)を整理し、高速に連鎖をたどる仕組みが求められた。
- ⇒推論エンジンの発明。
 - ルールエンジンともいう。推論エンジンというよりカッコいいので最近流行の呼び名。
- 仕組み自体は1960年代から研究されている由緒正しい分野。商用・フリー含め色々あります。
- ここでは、**CLIPS**を紹介します！

うかべん

同的ソフトウェア勉強会関西支部



NASAけない話



- C言語統合型プロダクションシステム
 - **C** Language **I**ntegrated **P**roduction **S**ystem
 - 1984年、NASAジョンソン宇宙センターにて開発。90年になり予算が付かなくなったため、以後パブリックドメインとして開発が続く。
 - 文法は、ほぼLISP。オブジェクト拡張あり。
 - 必要な関数はC言語で組む。組み込み開発向け。
 - 連鎖中枢「Rete」アルゴリズムの元祖。



うかべん

同的ソフトウェア勉強会関西支部



CLIPSの特徴 2



- 前向き連鎖（フラグが立ってイベント発火）
 - ルール競合時の発火順選択ルールが色々
 - 先に成立/複雑な条件/ランダムに選ぶ etc...
 - 同じ条件でも違うフラグで再発火したりする
- 継続的な開発によるサポート
 - この業界では老舗でリファレンス的な存在
 - 今年になってUTF-8対応版がリリース
 - リテラル文字列に日本語が使える！

うかべん

同的ソフトウェア勉強会関西支部



実際に使ってみた



- コマンドプロンプトなデモ

- 栞なデモ

- ……栞なデモは、越えられない壁の説明が長くなり過ぎるため、起動するのを見せるだけ(--;
- 需要があれば後でリアルタイムイベント追加……予定

うかべん

同的ソフトウェア勉強会関西支部



用語説明 その3



- 前向き連鎖：事実駆動(facts driven)
 - 事実により成立するルールを探し、そのルールが発火する新しい事実をたどる。
- 後向き連鎖：仮説駆動(hypothesis driven)
 - 現在の事実が成立するための事象を発火するルールを探し、最終的にルールが発動する条件を探す。

うかべん

同的ソフトウェア勉強会関西支部



前向き推論がうれしい分野



- とりあえず行動することが必要な、即断即決 A I 処理系に向く。
 - 即応が必要なゲームの A I。FF12のガンビットはルールエンジンの考え方そのもの。
 - イベント駆動の U I 処理系一般。
 - イベント駆動のどこいつ A I。触られ反応、連鎖トークなど。
- 「世界を動かす」ための仕組み

うかべん

同的ソフトウェア勉強会関西支部



後ろ向き推論がうれしい分野



- 事象の原因を推論したり、検証が成立するかどうかを見極める A I 処理系に向く。
 - 長考が必要なゲームの A I。詰め将棋など。
 - 予定表の管理。予定が連動している場合の行動調整。
 - オチを決めてからネタを考えるどこいつ A I。
- 「世界を知る」ための仕組み

うかべん

同的ソフトウェア勉強会関西支部



あらためて結論



ルールエンジンは槩向きか？

- 前向き推論が参照する知識ベースの構造は、条件判断フラグ付きのどこいつ会話辞書と見なせる。
 - 既成の考え方の拡張で取り組みが可能
 - 会話分岐用と単語選択用の知識ベースは別物かな？
- 後ろ向き推論に関しては要研究。
- CLIPSは、C言語で拡張書けばきっと化けるぜ！
 - 素のCLIPSは外の世界への干渉手段が無い
 - 拡張書くのが越えられない壁なのはヒミツ

うかべん

同的ソフトウェア勉強会関西支部



(補講)

ルールに向かない処理



- ルールエンジンは万能の道具ではない
 - 「順番」を排除できない⇒ルールの外で処理すべき
 - スタック・階層が必要な処理は不向きかも
GOTO⇒○、GOSUB⇒×
- If文はTHENの中で使うべきではない
 - 処理分岐はルールエンジンに任せる
 - 排除できない場合はルールに階層がある

うかべん

同的ソフトウェア勉強会関西支部



(補講)



「心」を実現するAI

- せめぎあう3つの世界
 - 自分が望むことを決める (欲望)
 - 現状から、知りうる世界を動かす (前向き駆動)
 - 欲望を満たす条件を探す (後ろ向き駆動)
- 可能な限り正確な未来を予測し、可能な限り欲望を満たす選択を探し続けるのが「心」の果たすべき役割。
- 適度に現実を裏切る「世界」が面白さを生む？

うかべん

同的ソフトウェア勉強会関西支部



(補講)



ルールエンジンとビジネス

- 推論エンジンはこれまで人工知能の研究分野だった。しかし最近、ビジネス分野で急速に注目を集めている
 - 昔：ルールをアプリケーションに実装。
 - 問題：ルールが増え、複雑に。変更も多発。
 - 今：アプリからルールを分離。
- 覚えたら飯が食える技術デスヨ。
- でも、うかがかに使う勇気！

うかべん

同的ソフトウェア勉強会関西支部



(補講)

踊る開発戦線



- それは「便利」な道具ですか？
 - 「工夫すれば使える」は落とし穴
 - 使える道具ほど、工夫は不要です
- 工夫しないほうがいいのか？
 - 工夫が思いつける技術者は土壇場に強い
 - 工夫しすぎる技術者はルーチンワークが遅い
 - 能ある鷹は爪を隠そう

うかべん

同的ソフトウェア勉強会関西支部



最後に



- 発表3日前に説明するルールエンジンをCLIPSに戻したので、資料が.NETよりだったりします。
- .netならWF付属のエンジン使うのが良いかも。
 - WFルールエンジンは機能限定・高速動作が特徴
- 今回、つかみの話に徹しました。実践してみても越えられない壁が多発しても泣かない事。
- 俺達の未来は始まったばかりだ！

うかべん

同的ソフトウェア勉強会関西支部



資料等



- 素材引用

- 手の素材 [FREE-CG.COM](http://free-cg.com) <http://free-cg.com/index.html>

- プログラム・参考文献

- CLIPS [A Tool for Building Expert Systems](http://clipsrules.sourceforge.net/) <http://clipsrules.sourceforge.net/>
- Rule記事 [ビジネスルールの館](http://www.iluminado.jp/index.php) <http://www.iluminado.jp/index.php>
- Rule記事 [現実世界におけるルールエンジン](http://www.infoq.com/jp/articles/Rule-Engines) <http://www.infoq.com/jp/articles/Rule-Engines>
- .net WF [WFルールエンジンの紹介](http://msdn.microsoft.com/ja-jp/library/aa480193.aspx) <http://msdn.microsoft.com/ja-jp/library/aa480193.aspx>
-

- ゴースト「くりっぷす！」関係

- 開発 [CLIPS棊\[CLIPSHIO\]開発ページ](http://code.google.com/p/ghost-clips/) <http://code.google.com/p/ghost-clips/>
- シェル¥0 [フリーシェル「自由なみかず」](http://www3.to/two-gy/) <http://www3.to/two-gy/>
- シェル¥1 [フリーシェル「insemination」](http://www24.big.or.jp/~tir/) <http://www24.big.or.jp/~tir/>
- バルーン [草葉犬死朗さんがうかべんで使った物](http://homepage3.nifty.com/deaddog/VB6.NAR) <http://homepage3.nifty.com/deaddog/VB6.NAR>

ハル



トメさん

やめといたほうが良い話

COPY RIGHT 2008 DOT-STATION. ALL RIGHT RESERVED.

うかべん

同的ソフトウェア勉強会関西支部