

ゴースト・と・で・の・お話

「コミュニケーション」をキーワードとした
ゴースト仕様の俯瞰

2007.08.05 うかべん@横浜#1

by 浮子屋

■ 講演者紹介

- ハンドル名：浮子屋（うきや）
- サイト：<http://ukiya.sakura.ne.jp/>
- 主な作品
Ghost The Hand, Dokudenpa
SAORI Osuwari, Handutil, Kenonoke
その他 SARS, SPAM, とか色々

■ 題目

「コミュニケーション」をキーワードに…

- ゴーストのしくみ
- ゴースト間コミュ
- ゴーストとユーザのコミュ
- その他、これからのコミュ（雑談）

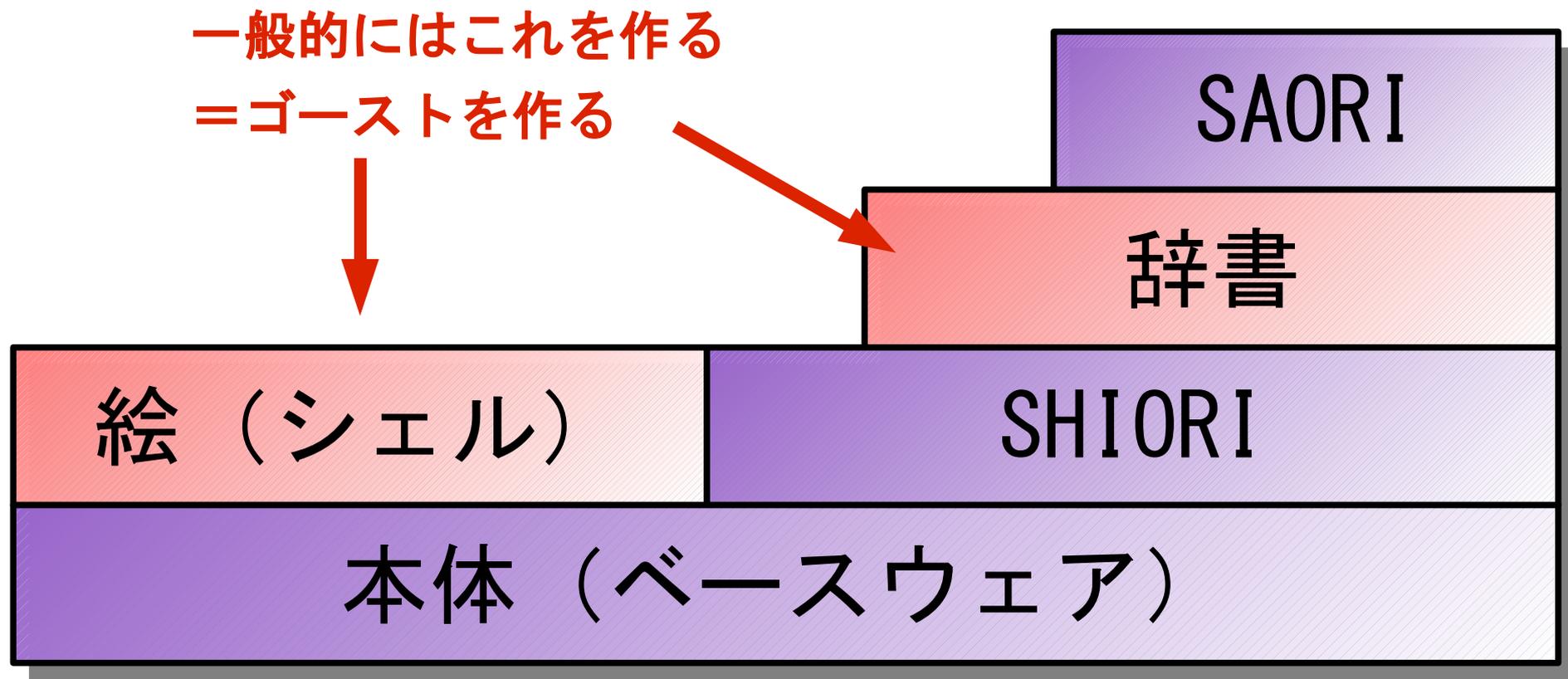
駆け足になりますが宜しくお付き合い下さい

■ ゴーストのしくみ

- ゴーストの構造
- SHIORI インタフェース

■ ゴーストの構造（1）

- ゴースト = 本体 + 絵 + トーク
- トーク = SHIORI + 辞書 (+ SAORI)



■ ゴーストの構造（2）

- 本体は「イベント」を SHIORI に通知
- SHIORI は辞書を読んだりして
- さくらスクリプトを返す

これ以外の動作はありません！



■ SHIORI インタフェース (1)

● Emily/Phase4 で胸をつついた場合

本体 → SHIORI (要求)

```
•GET SHIORI/3.0
Sender: SSP
Charset: UTF-8
SecurityLevel: local
ID: OnMouseDoubleClick
Reference0: 117 イベント
Reference1: 213
Reference2: 0
Reference3: 0
Reference4: Bust
Reference5: 0
```

SHIORI → 本体 (応答)

```
SHIORI/3.0 200 OK
Sender: AYA
Charset: UTF-8
Value: ¥u¥s[10]¥h¥s[21]
む〜〜 ¥w5…¥w5…。
¥w9¥w9¥n
いつか仕返ししてやるんだから。
¥w9¥w9
¥u¥s[11] 何するつもりなの
¥w5…¥w5…。 ¥e
```

さくらスクリプト

■SHIORI インタフェース (2)

- イベントにはたくさん種類があります

<http://crow.aqrs.jp/reference/all/Shiori/index.html>

- どんな SHIORI でも

イベント → さくらスクリプト

リクエスト → レスポンス

の流れは変わりません

- 自発的に喋っているように見えるのは、OnSecondChange イベントなどをキッカケにしています
- SSP だと、開発者パレット (Ctrl+Shift+D) の「SHIORI ログを取る」で見ることができます

(ghost/master フォルダの ssp_shiori_log.txt)

■ ゴースト間コミュニケーション

- OnCommunicate
- 通常コミュ
- OnCommunicate 応用例
→ 台本、大根、連鎖、台本連鎖
- RaiseOther
→ OnRequestProperty
- DirectSSTP
- 切り替え反応

■ OnCommunicate

- OnCommunicate は、「他のゴーストから話しかけられたよ」というイベント
- リクエスト（本体からゴーストへイベントを通知）
→ レスポンス（ゴーストから応答を本体に返す）
の流れは変わりません！
- ではどうやって他のゴーストとお話してるのでしょうか？

■ 通常コミュ（１）

- レスポンスに Reference0 が含まれると、
コミュが発生します
- Reference0= 相手ゴースト名
- 本体は、レスポンスの Value（さくらスク
リプト）を喋らせた後、相手ゴーストに、
OnCommunicate イベントを通知
- OnCommunicate の Ref. 0 は発話元ゴースト
名、Ref. 1 は発話内容です

■ 通常コミュ (2)

- 具体例：ゴースト A とゴースト B で往復

ゴースト A → 本体 (応答)

```
SHIORI/3.0 200 OK  
Value: ¥h こんにちは。 ¥e  
Reference0: ゴースト B
```

本体 → ゴースト B (要求)

```
GET SHIORI/3.0  
ID: OnCommunicate  
Reference0: ゴースト A  
Reference1: ¥h こんにちは。 ¥e
```

本体 → ゴースト A (要求)

```
GET SHIORI/3.0  
ID: OnCommunicate  
Reference0: ゴースト B  
Reference1: ¥h さようなら。 ¥e
```

ゴースト B → 本体 (応答)

```
SHIORI/3.0 200 OK  
Value: ¥h さようなら。 ¥e  
Reference0: ゴースト A
```

■ 通常コミュ (3)

- Reference0 をレスポンスに入れる方法
- SHIORI によって違います

里々	\$Value0 相手ゴースト名
華和梨	\$(setstr System.Response.Reference0 相手ゴースト名)
OpenKEEPS	\$(SendTo 相手ゴースト名)
文	res_reference0= 相手ゴースト名

■ 通常コミュ（４）

- OnCommunicateで発話元ゴースト名（Ref. 0）や発話内容（Ref. 1）を知る方法
- またまたSHIORIによって違います

里々

通常、里々内で以下のエントリに変換

*ゴースト名「発話内容

（R 0）（R 1）で取得も可能

華和梨

System. Callback. OnGET エントリで

System. Request. ID エントリが「OnCommunicate」

の時にイベント処理、System. Request. Reference0

で取得（Ref. 1も同様）

OpenKEEPS

TalkComm.<ゴースト名>.<識別子>.cond

エントリ中の単語が発話内容に含まれた時、

TalkComm.<ゴースト名>.<識別子>エントリで

イベント処理

文

OnCommunicate 関数でイベント処理

reference0 変数で取得（Ref. 1も同様）

■ 通常コミュ (5)

- 一緒に立ってるゴースト名を知る方法
- やっぱり SHIORI によって違います

里々 (隣で起動しているゴースト)
華和梨 System. Callback. OnNOTIFY エントリで
System. Request. ID エントリが
「otherghostname」の時に Ref. をリスト化
OpenKEEPS System. OtherGhost
文 ghostexlist

でも正確じゃないかも。色々ややこしい話は

<http://d.hatena.ne.jp/ukiya/20060531/1149088028>

■ OnCommunicate 応用ルール

- レスポンスに Ref. 1 以降が含まれると
- OnCommunicate の Ref. 2 以降として
- そのまま相手に通知されます

ゴースト A → 本体 (応答)

```
SHIORI/3.0 200 OK  
Value: ¥h こんにちは。 ¥e  
Reference0: ゴースト B  
Reference1: ぽにゃ  
Reference2: さとー
```

本体 → ゴースト B (要求)

```
GET SHIORI/3.0  
ID: OnCommunicate  
Reference0: ゴースト A  
Reference1: ¥h こんにちは。 ¥e  
Reference2: ぽにゃ  
Reference3: さとー
```

このとき、番号が1つ増えることに注意！

■ 応用例（１）：台本コミュ

- Ref. 1 に「ECHO/1.0」 Ref. 2 に内容
- それを受けたゴーストはその「内容」をそのまま喋るという「取り決め」

ゴースト A → 本体（応答）

```
SHIORI/3.0 200 OK  
Value: ¥h 好きなものなに? ¥e  
Reference0: ゴースト B  
Reference1: ECHO/1.0  
Reference2: ¥h ケーキです。 ¥e
```

本体 → ゴースト B（要求）

```
GET SHIORI/3.0  
ID: OnCommunicate  
Reference0: ゴースト A  
Reference1: ¥h 好きなものなに  
Reference2: ECHO/1.0  
Reference3: ¥h ケーキです。 ¥e
```

相手ゴーストが対応してないと動作しません。

■ 応用例（２）：大根コミュ

- Ref. 1 に「JPRadish」 Ref. 2 以降に変数
- 変数を見て適当に処理してねというもの
- LR 氏「ヒュールフリスと猫な杖」
- 他のゴーストがヒュールから買い物をしたたりするのを利用
- 何番目の変数が何の意味か、ゴーストの間で取り決めしておく必要があります
- http://lre.s165.xrea.com/uka/dl_jpr.htm

■ 応用例（3）：連鎖コミュ

- $A \rightarrow B \rightarrow A \rightarrow B \dots$ とコミュを連鎖させます
- 予め両方のゴーストに自分の喋る台本を仕込んでおきます
- Ref. 2 で台詞番号を送ることで、相手が何番目の台詞を喋るか制御します
- 異なる作者のゴースト間コミュに向きます
- 実装例：ヴィイと Hand
浮子屋本舗「自作ゴースト/Hand/コミュニケーション連鎖」

■ 応用例（3）：連鎖コミュ

- 具体例：ヴィイとHandで往復

Hand→本体

Value: なんだよヴィイ…
Reference0: ヴィイ
Reference1: viiandhand-1
Reference2: 1

本体→ヴィイ

ID: OnCommunicate
Reference0: Hand
Reference1: なんだよヴィイ…
Reference2: viiandhand-1
Reference3: 1

本体→Hand

ID: OnCommunicate
Reference0: ヴィイ
Reference1: そんないやらしい手
Reference2: viiandhand-1
Reference3: 2

ヴィイ→本体

Value: そんないやらしい手…
Reference0: Hand
Reference1: viiandhand-1
Reference2: 2

■ 応用例（４）：台本連鎖コミュ

- $A \rightarrow B \rightarrow A \rightarrow B \dots$ とコミュを連鎖させます
- 予め片方のゴーストに両者が喋る台本を仕込んでおきます
- リクエストのRef. 1は ECHO/1.0、Ref. 2以降に台本をずらずら並べます
- ゴーストAに台本を、ゴーストA, Bに ECHO/1.0を受けた際の処理を実装します
- 同じ作者のゴースト間コミュに向きます

■ 応用例（４）：台本連鎖コミュ

- 具体例：リリーとビオラ（黒羽龍矢氏）

リリー→本体

本体→ビオラ

ビオラ→本体

Value: リリー台詞 1
Ref0: ビオラ
Ref1: ECHO/1.0
Ref2: ビオラ台詞 1
Ref3: リリー台詞 2
Ref4: ビオラ台詞 2
Ref5: リリー台詞 3

ID: OnCommunicate
Ref0: リリー
Ref1: リリー台詞 1
Ref2: ECHO/1.0
Ref3: ビオラ台詞 1
Ref4: リリー台詞 2
Ref5: ビオラ台詞 2
Ref6: リリー台詞 3

Value: ビオラ台詞 1
Ref0: リリー
Ref1: ECHO/1.0
Ref2: リリー台詞 2
Ref3: ビオラ台詞 2
Ref4: リリー台詞 3

リリー台詞 1

Ref. 3 を喋って Ref. 4 を
Ref. 2 に入れる

ビオラ台詞 1

■ RaiseOther (1)

- 他のゴーストにイベントを起こさせる
- さくらスクリプトのタグ (SSPのみ)
- ¥! [raiseother, ゴースト名, イベント名, r0, r1, r2...]
- 他のゴーストに、「ユーザの操作」や「本体からのイベント」と錯覚させることができます
- もちろん見破ることもできます
- オリジナルのイベントを送ることもできます
(相手が対応していないと意味はありません)

■ RaiseOther (2)

- 胸をつついたらと錯覚させる

ゴースト A → 本体

```
SHIORI/3.0 200 OK  
Sender: AYA  
Charset: UTF-8  
Value: 胸を触ってやる!  
¥! [raiseother, ゴースト B,  
OnMouseDoubleClick,  
0, 0, 0, 0, Bust, 0]¥e
```

バルーンにマーカーが出ます



本体 → ゴースト B

```
GET SHIORI/3.0  
Sender: ゴースト A  
Charset: UTF-8  
SecurityLevel: local  
ID: OnMouseDoubleClick  
Reference0: 0  
Reference1: 0  
Reference2: 0  
Reference3: 0  
Reference4: Bust  
Reference5: 0
```

■ 応用例 : OnRequestProperty

- RaiseOther を利用してオリジナルイベントをやりとりすることで、ゴースト間で変数を送りあいます
- ゴースト間で、「あなたの性別は？」 「男です」と質疑応答するイメージです
- ゴーストAは OnRequestProperty イベントで、欲しい変数を指定します（質問）
- ゴーストBはそれを受けたら、 OnGetProperty イベントで、変数の値を返します（応答）

■ 応用例 : OnRequestProperty

- もちろん送る相手が OnRequestProperty に対応していないと意味はありません
- LR 氏「ヒュールfrisと猫な杖」
Don 氏「ゴーストじてん」 ほか
- 詳しくは . . .

LR 氏雑記

<http://ire.s165.xrea.com/uka/wps/page/6>

ゴーストじてん Readme

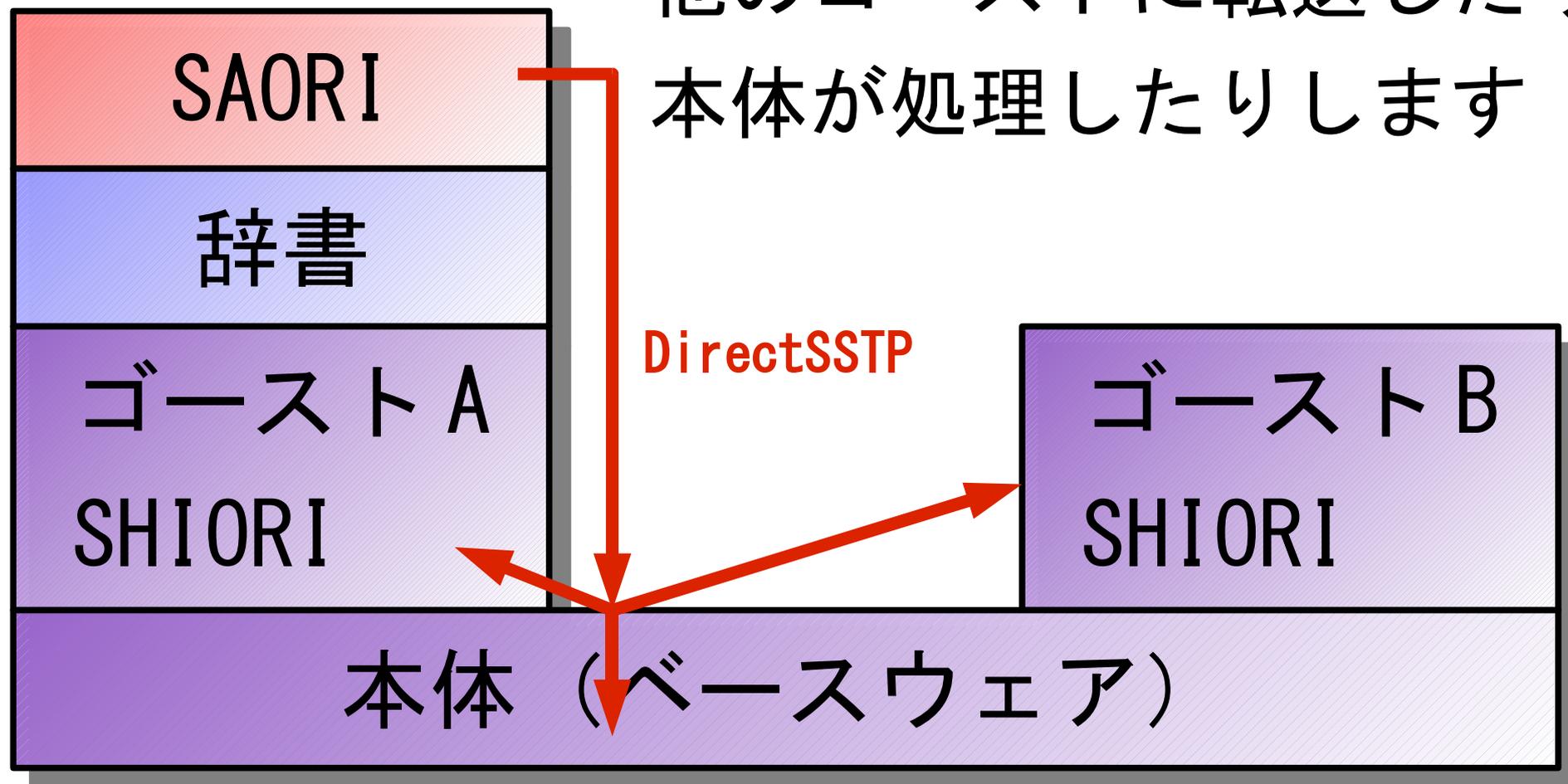
<http://home.384.jp/evidence/update/ghostbook/readme.txt>

■DirectSSTP (1)

- 本体に向けて他のプログラムから情報を送るしくみです (Windows のみ)
- コミュだけでなく色々な用途に使えます
- コミュに使った場合、RaiseOther と違うのは
- 使うのに SAORI が必要です
- SSP 以外でもうごきます
- さくらすクリプトではなく、辞書の中で直接 SAORI を使って動かすので、好きなタイミングで動作させることができます

DirectSSTP (2)

- 送られた DirectSSTP は本体が解釈して、他のゴーストに転送したり本体が処理したりします



■DirectSSTP (3)

- 詳しく説明してるとキリがないので省略！
- 応用システム
SSTP Bottle、SSTP Viewer、その他色々
- The Handを作ったときは、RaiseOtherが無かった
たので、DirectSSTPを使って、他のゴーストに
「触られた」と錯覚させています。

■ 切り替え反応（１）

- ゴーストの起動／終了に関連した一群のイベントを切り替え反応と呼ぶことが多い
- SSP の場合、同時複数起動できるので、呼び出し～呼び出され関連のイベントもある
- 相手ゴーストの名前や、相手が切り替えの際に喋った台詞が通知されるので、それらに応じて反応を変えるのが一般的
- うまく使うとコミュの幅が広がる
- 例： ROAMING * K I D ～二匹の子やぎ
～（やぎむつき氏） 170パターン以上

■ 切り替え反応 (2)

イベント	いつ通知	誰に通知
OnGhost Changing	他のゴーストへ切り替えしようとしたとき	切り替わり元ゴースト
OnGhost Changed	他のゴーストから切り替わったとき、OnBootの代わりに	切り替わり先ゴースト
OnGhost Calling *	他のゴーストを呼び出そうとしたとき	呼び出したゴースト
OnGhost Called *	他のゴーストから呼び出されたとき、OnBootの代わりに	呼び出されたゴースト
OnGhostCall Complete *	他のゴーストを呼び出して、呼び出されたゴーストの起動が完了したとき、OnOtherGhost Bootedの代わりに	呼び出したゴースト
OnOtherGhost Booted *	他のゴーストが呼び出されて、呼び出されたゴーストの起動が完了したとき	呼び出したゴースト以外全部
OnOtherGhost Closed *	他のゴーストが終了したとき	起動中のゴースト全部
OnOtherGhost Changed *	他のゴーストが他のゴーストに切り替わって、切り替わり先のゴーストの起動が完了したとき	起動中のゴースト全部

■ ゴーストとユーザのコミュニケーション

- ユーザコミュのしくみ
- ユーザコミュ考察

■ ユーザコミュのしくみ

- ¥! [open, communicatebox] タグで、コミュニケーション入力ボックスを開きます
- ユーザが入力したら、OnCommunicate で内容が通知されます
- その際の「ゴースト名」 (Ref.0) は「user」になります
- ゴースト名が「user」になる以外は、ゴースト間コミュと一緒にです

■ ユーザコミュ考察

- 人間が入力するので、ゴースト間と違い、何が入力されるか予測がつきません
- つまり、自然言語処理が必要です！
- 入力からキーワードを拾って反応するのが一番一般的なやりかたです
- 言葉のゆらぎの吸収がポイントです
- 実装例はたくさんあります。ユーザコミュに特化した例：CMN、ゆうとぴあ 等

■ その他、これからのコミュ

- ゴースト作者とユーザのコミュ
- ユーザ間コミュ

■ ゴースト作者とユーザのコミュ

- ゴーストから Web 拍手
(隙間の世界 / リーライナ : おぞん氏)
- 質問を貰って全力更新!
(猫と魔法使い : あきつ氏)
- ゴーストにチャットクライアント搭載、
チャットで作者がゴーストのトークを発言
→ 起動中のゴーストがリアルタイムで喋
る、とか (IRC、Lingr、Skype 等)

■ ユーザ間コミュ

- Web サイトで意見交換（うわひょ、ゴースト掲示板、MarbleNote、とか）
- ゴーストでチャットー SSTPBottle が既に実装、応用として IRC、Lingr 等
- ゴーストに何かを入力すると、同じゴーストを起動している人に届くとか
- アバターと違いトーク（ゴーストの人格、キャラクター）の影響

■ おわり

- 駆け足の説明で失礼しました
- 誤り・疑問等あればご指摘ください
- C. Ponapalt 氏、さと一氏にご支援頂きました。ありがとうございました
- 御清聴ありがとうございました