

Program Integrated Information (PII) 翻訳のための PII 概念構築

加藤 直孝¹ 有澤 誠²

(株)日本アイ・ビー・エム ナショナル・ランゲージ・サポート¹

慶應義塾大学 環境情報学部大学院 政策・メディア研究科²

katosan@jp.ibm.com¹

概要: 本稿はソフトウェアのローカライゼーションにおける文字列の翻訳に関するものである。現在の多くのアプリケーション・システムはユーザーとのインタラクションを前提としており、ユーザーとの意思疎通のためにテキストを用いる。このテキスト情報のうちプログラムに統合した情報を Program Integrated Information (PII) と呼ぶ。従来の PII に関する文献は PII のプログラミングの側面のみを議論していた。そこで前稿[1]は PII の自然言語の側面に焦点をあて PII の概念構築を行った。本稿は[1]で構築した概念のより詳細な部分を説明する。同時に、詳細な部分を明らかにしたことによって見えてくる PII 翻訳をサポートする新しい手法を示す。

キーワード: 翻訳、PII、Program Integrated Information、文字の外部化、地域化、国際化、モデル、GUI、言語記述、概念

Elaboration of Program Integrated Information Concept

Naotaka Kato¹ Makoto Arisawa²

IBM Japan, Ltd. Translation Service Center.¹

Keio University, Faculty of Environmental Information and Graduate School of Media and Governance²

katosan@jp.ibm.com¹

Abstract: Most of the application systems require interaction with users. Such applications use text strings to communicate with users. Those strings are integrated in software and are called “Program Integrated Information (PII).” This paper focuses translation of those text strings. Most of the articles that explain PII focus only the programming aspect of PII. Our previous paper would be the first article that built the concept of PII from the natural language point of view. This paper extends and elaborates the concept of PII, particularly from the point of Japanese “So.” The elaboration indicates that there exists new translation approaches for PII translation. Two approaches are briefly presented in this paper.

Keywords: translation, PII, Program Integrated Information, string externalization, localization, internationalization, model, resource file, architecture, MRI, Machine Readable Information, character coding, locale, initialization file, GUI, language

1. プログラムの国際化と PII の翻訳の問題

国際化を必要とする多くのアプリケーションはユーザーとの意思疎通のためにテキストを用いる。このテキスト情報のうちプログラムに統合した情報を Program Integrated Information (PII) と呼ぶ。プログラムを国際化するには、この PII を何カ国語かに翻訳する必要がある。プログラムの開発組織では翻訳する可能性のある文字列は別のテキストリソースファイルに分離する。このテキストリソースファイルにはキーおよび分離したストリングを置く。そして、プログラム中にキーを埋め込み、キーの参照により適切な文字列を得る。IBM では、このテキストリソースファイルの翻訳作業はプログラムの開発組織では行わず、翻訳の業務に特化した組織が行っている。加藤はこの PII の翻訳業務に従事しており、PII の翻訳および翻訳検証、翻訳の修正等の

メンテナンスにともなうさまざまな問題に直面している。前稿[1]では、その問題を解くための基礎となる PII の概念構築を行った。[1]では色々な点に焦点をあてながら、PII の世界を巨視的に捉えている。簡単に要約すると次のような点を述べている。

- (1) PII には多くの側面があり、扱う立場により色々な「相」が存在すること。
- (2) コンピュータ科学やエンジニアリングが、PII のプログラミングの側面にのみ注意を払い[4][5]、PII の自然言語の側面に光を当ててこなかったこと。
- (3) PII のモデル構築: PII の世界を記述している Feature にはどのようなものがあるかを明確にし、PII の巨視的なモデルを作成していること。

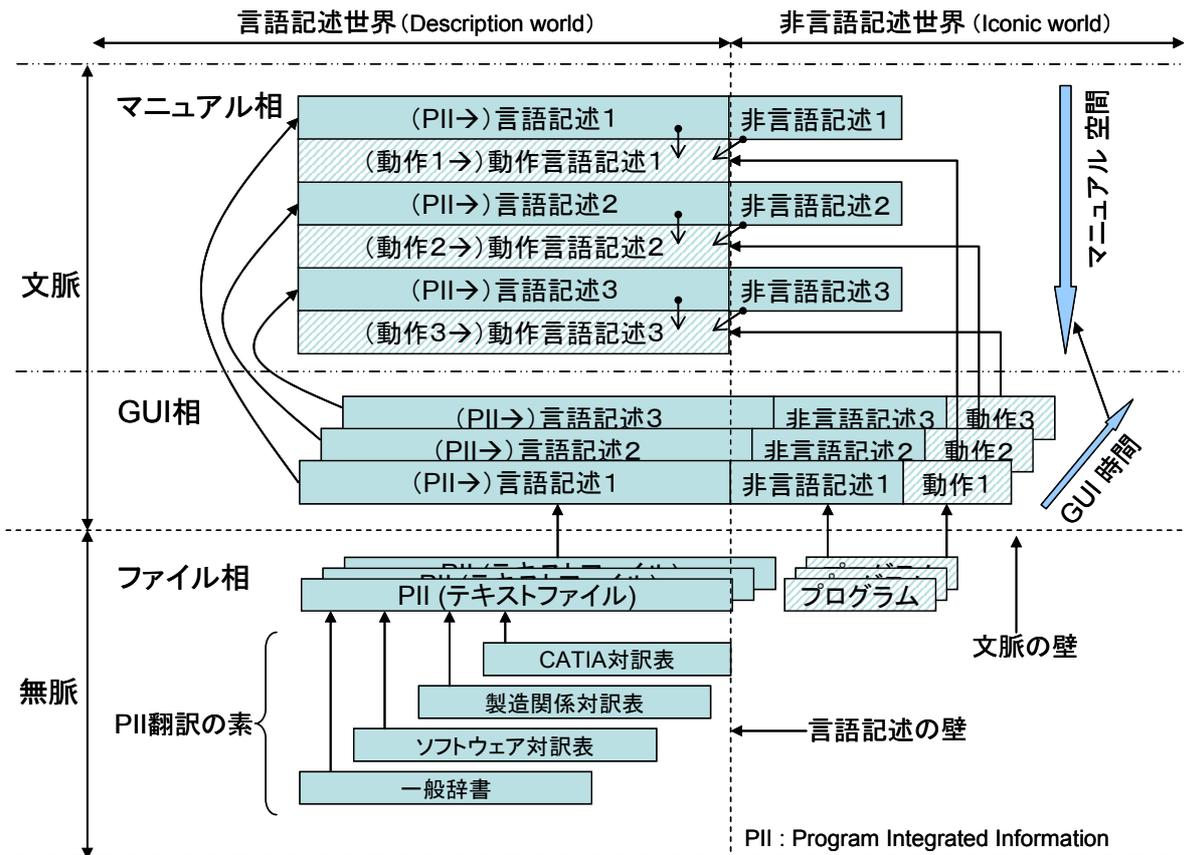


図 1 PII の概念 — 言語記述と非言語記述

- (4) PII の概念構築: PII の世界で、文脈、相、Feature の関係を明確にすることで、PII の概念構築を行っていること。
- (5) 「着ぐるみ」の比喩を用いることによる PII の Identity Feature の説明と PII と対訳表(辞書)の関係の説明。
- (6) Identity Feature は PII の連鎖を司る PII の鍵となる Feature であること。

本稿では[1]から発展し、次の 2 点をより詳細に述べる。

- (1) ファイル相の PII 中の語句がどのようにそれぞれの対訳表の語句と結びついているか。
- (2) ファイル相から GUI 相、そしてマニュアル相にどのように相転移が起こるか。

以下第 2 章では上記 2 点に関して、図1を用いて PII の概念のより詳細な説明を行う。第 3 章では、対訳表と PII の関係を発展すれば、新しい翻訳手法がありうること、第 4 章では、詳細な概念から新たなマニュアル作成サポートツールがありうることを示す。

2. PII の概念(相の詳細)

2.1 PII と対訳表

IBM では Translation Manager(TM) という翻訳支援ツールを使用して翻訳を行う。このツールは文章をセグメントに分割し、分割したセグメントが既翻訳の翻訳対の原語と同じあるいは似ているときに、その翻訳対を翻訳者に提示するツールである。このツールは既翻訳とは別にカテゴリー別に分けた翻訳対(IBM ではこれを辞書と呼んでいる)をあらかじめ準備しておく、原語に対して対応する語句があれば翻訳対のカテゴリー名と同時に表示する。翻訳者は対訳表の上位にあるものを優先して訳語を選んでいく。PII は文脈を見ずに翻訳するので、このような手段をとることが多い。

2.2 PII の相転移 (縦軸方向)

図1は、PII を縦軸方向で文脈と無脈(文脈のない世界)に分けた。[1]では無脈の世界の手前には対訳表が存在することを指摘した。本稿では[1]の内容を発展し、対訳表および一般の辞書(英和辞書等)を「PII 翻訳の素」と命名し PII 翻訳の素となる要素とした。TM による翻訳で、同じ語句に対し複数のカテゴリーに翻訳対がある場合、翻訳

者は「PII 翻訳の素」の上位の対訳表から訳語を選ぶ。なお、図1で「CATIA 対訳表」とあるのは PII 翻訳対象アプリケーションの例で、フランス Dassault 社の CAD システムアプリケーションシステムである。また、本稿は横軸方向の存在を図1に取り入れた。言語記述世界(Description world)と非言語記述世界(Iconic world)による分類である。言語記述世界とは言葉による記述の世界であり、Statement world と言い換えることもできる。非言語記述世界は言葉を使わない記述の世界であり、Simulation world と言ってもよい。それらの境界を「言語記述の壁」と呼んでいる。

テキストファイルとしての PII は無脈の世界に属する。GUI 上の PII とマニュアル上の PII は文脈の世界に属している。図1では、PII のそれぞれの相を上から、マニュアル相、GUI 相、ファイル相と分けて示している。同じ記述が違って見えるのは PII の相が変わったからである[1]。

[1]で述べた縦軸方向の説明を要約すると次のようになる。ファイル相で PII をいくら眺めても、PII の正確な意味を知ることはできない。しかし、全く同じ PII を GUI 相やマニュアル相で読むと意味が理解できる。ファイル相の PII と GUI 相の PII の間には本質的な違いがある。その違いを生んでいるのは、プログラムとユーザーとのインターアクションである。文脈と無脈の間には壁が存在し、このインターアクションのエネルギーにより PII は壁を越える。この壁を[1]では「文脈の壁」と呼んでいる。

2. 3 PII の非言語記述に対する関係

2. 3. 1 相変化に直交する軸(横軸)の追加

本稿では、PII をより深く理解するために、相転移を縦軸方向の分類とし、それに対して横軸方向の分類を追加した。

プログラムと人間が意志疎通するときには、大きく分けると 2 つの形態で行っている。言語記述(Description)による方法と非言語記述(Iconic representation)による方法である。PII は本来言語による記述である。従って PII は言語記述世界に属する。非言語記述はアイコンやウィンドウといったグラフィックスによるものや音や音声などを指す。図1ではこの言語記述と非言語記述を横軸方向に示した。

PII の厳格な定義[3]を適用すると、PII は非言語記述も含む。しかし、本稿ではテキストファイル上に存在する PII に限定して PII と言うことにする。

2. 3. 2 非言語記述(Iconic world)

ファイル相の言語記述の部分が PII ファイルであるならば、ファイル相の非言語記述は PII 以外のプログラムである。ただしここでのプログラムは PII 以外のリソースファイルも含む。図1で、そのプログラムを PII (テキストファイル)の右側に示した。

このプログラムはファイル相から GUI 相へ転移する。プ

ログラムを実行し、ディスプレイ上にグラフィックスとして表示すると、当然見え方は変わる。このときファイル相にあるプログラムは GUI 相に転移し「非言語記述」と「動作の流れ」に分かれる。図1の GUI 相の言語記述部分の右側(非言語記述側)にそれらを示す。ここで非言語記述と述べているのはウィンドウ中の PII 部分以外といってよい。動作とは、マウスボタンによる操作やキータイピングや音声入力等とそれに対するプログラムの反応を指す。動作1、動作2、動作3が「動作の流れ」に相当する。

ソフトウェアのマニュアルを作成するときには、実際にはプログラムが出力するウィンドウを gif ファイルといったイメージでキャプチャーしマニュアル中に取り込む。そして、それらのウィンドウの内容やそのウィンドウに対する操作を説明することで、プログラムの使い方を説明する。このマニュアル作成は PII にとっては GUI 相からマニュアル相への相転移となる。図1の非言語記述世界では、GUI 相にある「非言語記述」は、基本的にはそのままマニュアル相に転移する。一方、「動作」はマニュアル相では非言語記述(Iconic representation)ではなく言語記述(Description)によってそれを示す。図1の GUI 相では非言語記述世界に属していた「動作」が、マニュアル相では言語記述世界に属する。図1の「動作記述1」等がそれに相当する。GUI 上でアイコンやボタン等の動作として表示していたものは、マニュアル上では言語記述として表示する。

この GUI 相からマニュアル相への転移は時間軸上での GUI の流れ(文脈)が空間軸上での流れ(文脈)に相転移している。図1では太い矢印でその流れを示した。

2. 3. 3 PII と動作言語記述

マニュアル相の動作言語記述は言語記述を参照する。図1の黒丸で始まる矢印は、矢印側が黒丸側を参照することを意味する。たとえばある GUI 上の入力域に何らかの値を入力するときや、あるラジオボタンを選択するときには、どの入力域かどのボタンかを示すために、入力域の近傍にある言語記述を参照することが多い。またアイコンそのものがボタンであるときには、言語記述中にアイコンを挿入したりする。図1のマニュアル相の「言語記述」から「動作言語記述」に引いた矢印や、「非言語記述」から「動作記述」に引いた矢印がそれらの参照を示す。従って、ファイル相に始まる PII の連鎖は動作言語記述にまで及ぶ。

2. 4 マニュアルにおける PII の連鎖

原典のファイル相の PII を変更すると、単に PII のファイル上の翻訳を変更するだけでなく、マニュアルに関しても PII の関連箇所を変更する必要がある。マニュアル作成では、PII を変更した場合はキャプチャーの取り直しを意味する。変更した PII を参照する動作記述も変更する必要がある。翻訳には関係ないけれども、原典のアイコンを変更す

れば翻訳したマニュアルのアイコンも変更せねばならない。

マニュアルの修正は HTML ファイルであれば、部分的なファイルの変更で済む。しかし、PDF や紙のマニュアルでは全体の再作成を必要とする。製品につける紙のマニュアル等では、再作成の時間がない場合は訂正の用紙を追加することになる。

原典の PII の変更によせよ翻訳の訳語の修正によせよ、問題となるのは PII の変更がマニュアル上のどこに及ぶのかを容易に見えてくれないことである。何千ページにもおよぶマニュアルでは、何箇所どこに影響するかを把握することは容易ではない。なお、ファイル相の PII と GUI 相の PII の連鎖を容易に見つける手法は[2]で紹介した。PII の全文字列にコンパクトな ID を振る手法である。

3. 対訳表と新しい翻訳手法

本章では対訳表を「PII 翻訳の素」と概念化することで、新たな発想が生まれる例を示す。

PII には Identity Feature が存在する[1]。Identity Feature の応用として、PII 文字列にコンパクトな ID を振ると翻訳検証作業の効率が飛躍的に改善する[2]。飛躍的に改善する理由は、コンパクトな ID により PII のファイル相と GUI 相の連鎖が見えるようになるからである。図1の概念からこの考え方を「PII 翻訳の素」にも広げることができる。

PII が GUI 上での構成要素となっているのと同様に対訳表中のそれぞれの語句はファイル上の PII の構成要素の一部になっている。対訳対全てに適切な規則で ID を振り、PII 中の語句には対訳表中の ID をもたせる。この ID は言葉の意味に対する ID に発展しうる。IBM のマニュアルは通常9カ国語に翻訳する。翻訳対上に現れる原語の語句に ID をふり、翻訳対の語句の意味を記述できれば、もはや翻訳者が訳語を選択しなくとも、原典作者に ID を選んでもらえれば自動的に翻訳対は決まるので、9カ国の翻訳者が個別に訳語選択に関して悩む必要はなくなる。PII の新たな翻訳手法である。

4. マニュアル作成支援ツール

前章の言語記述と非言語記述の分類から、マニュアルでは「動作」を「非言語記述世界」から「言語記述世界」に変換していることがわかった。Windows のような OS では、OS がアプリケーションに送るメッセージをトラップすることによってどのような動作がアプリケーションに対してなされるかを知ることができる。トラップしたメッセージを言語に変換するテーブルをもてば、動作は言語記述に変換できる。たとえば、メッセージをトラップするツールがドキュメントファイルを生成するようにしておけば、マニュアル作成支援ツールになる。

そのツールはドキュメントファイル上に「GUI のキャプチャーイメージ」と「動作言語記述(動作を言語記述に変換したテキスト)」を順に記録する。するとそのツールが作成したドキュメントがマニュアルのドラフトとなる。具体的には、目的のアプリケーションのウィンドウフレームが生成した後(MS Windows ならば WM_CREATE メッセージを受けた後)にツールはアクティブなウィンドウのイメージをキャプチャーする。そして、そのイメージをツールのドキュメントに貼り付ける。次にツールは今開いたアクティブなウィンドウフレームに渡るメッセージをトラップして動作言語記述に変換する。そして、貼り付けたイメージの続きに言語記述を追加する。たとえば、WM_LBUTTONDOWN というメッセージをトラップしたなら「マウスの左ボタンを押します」に変換し、ドキュメントに追加する。ツールがドキュメントに追加する動作言語記述(たとえば、「マウスの左ボタンを押します」)を翻訳すれば、多言語のマニュアルのドラフトが生成できる。

5. むすび

本稿は、PII 翻訳の概念構築をより詳細な部分に発展した。[1]の PII の概念および本稿の詳細な概念を基に PII を理解すれば、コンピュータには人間による PII 翻訳やマニュアル翻訳を色々な角度からサポートできることがわかってくる。第3章、第4章はその具体的な例である。いずれの例もさらに研究をする必要がある。

謝辞

[2]の研究会や[6]で京都大学の佐藤理史先生はじめ色々な方から示唆をいただいた。ここに謹んで感謝の意を表す。

参考文献

- [1]加藤直孝, 有澤誠, “Conceptualization of Program Integrated Information”, 情報処理学会研究報告, 2004-NL-166 (2005)
- [2]加藤直孝, 有澤誠, “翻訳検証テストのためのストリングソース ID”, 情報処理学会研究報告, 2004-NL-162, pp. 95-102 (2004)
- [3]IBM, Designing Internationalized Products, National Language Design Guide Volume1, Fifth Edition, February 2004
- [4]Andrew Deitsch and David Czarnecki, Java Internationalization, O'REILLY, 2001
- [5]Nadine Kano, Developing International Software for Windows95 and Windows NT, Microsoft Press, 1995
- [6]佐藤理史, “「言い換え」特集号編集後記”, 自然言語処理, Volume11, Number4, pp. 209-210, 2004/10
- [7]佐藤理史, “境界認定の提案: (1) コンセプトと実現法”, 情報処理学会研究報告, 2004-NL-164, pp.25-32 (2004)
- [8]佐藤理史, “境界認定の提案: (2) 背景と思想”, 情報処理学会研究報告, 2004-NL-164, pp.33-40 (2004)