

STARS 上での ACOP Beans の利用

○小菅隆^{A)}、Philip Duval^{B)}

^{A)} 高エネルギー加速器研究機構 物質構造科学研究所

^{B)} DESY (Deutsches Elektronen-Synchrotron)

概要

STARS(Simple Transmission and Retrieval System)^{[1][2]}はシンプルで汎用性の高いソフトウェアであり、高エネルギー加速器研究機構 Photon Factory では、ビームライン制御用に広く利用されている。一方、ACOP(Advanced Component Oriented Programming) Beans^[3]は、DESY (Deutsches Elektronen-Synchrotron : ドイツ) により開発された Java をベースとして開発された強力な GUI 開発のためのツールである。今回、STARS の Java 用インターフェースの開発が行われた事で、ACOP Beans の利用が可能となった。ここでは STARS 上での ACOP Beans の利用について述べる。

1 STARS

STARS は、比較的小規模な制御システム向けの非常にシンプルなソフトウェアで、Photon Factory ではビームライン制御等に広く利用^{[4][5]}されている。2012 年 1 月現在、30 を超えるビームラインに導入されており、それぞれ安定に動作している。また、ビームライン制御システムだけではなく、ビームラインインターロック集中管理システムや Photon Factory 実験ホール入退室管理システムのソフトウェアとしても利用されている。なお、STARS に関するソフトウェア群は STARS の Web サイト (<http://stars.kek.jp/>) より無料でダウンロード、利用が可能¹⁾である。

STARS はサーバクライアント型のソフトウェアで、1つの STARS Server に複数の STARS Client が TCP/IP Socket を利用して接続する (図 1 参照)。各 STARS Client が宛先情報を持ったメッセージを STARS Server に送信すると、STARS Server は宛先として指定された STARS Client にそのメッセージを配信する。なお、1つのサーバに接続された各 STARS Client はユニークな名前 (Node Name) を有する事になっており、これらの Node Name がメッセージの宛先として利用される。STARS では Node Name に階層化構造を利用する事が可能で、区切り文字としてはピリオド「.」が使用される。(例:「Br2.Dev2」)

STARS が扱うメッセージ

STARS において送受されるメッセージはすべてテキストのメッセージであり、メッセージのターミネータとしては「LF」(0Ah) が使用される。STARS のメッセージは「送信元」+「>」+「宛先」+「スペース」+「コマンド or リプライ or イベント」で構成され、例としては次のように構成される。

Term1>Dev1 GetValue

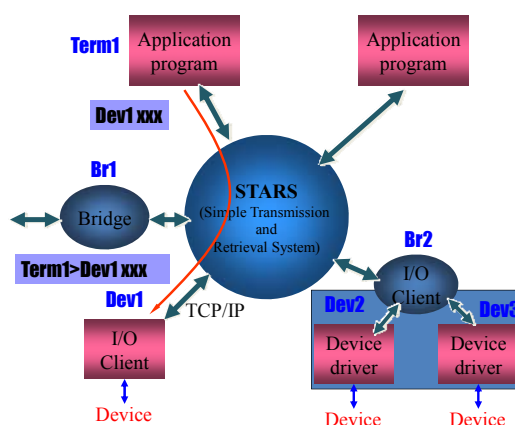


図 1. STARS の構成イメージ

¹⁾ 無保証、自己責任においての利用。なお、可能な範囲で、問い合わせ及び相談に対応。

ここで、STARS Client からメッセージを送信する際、メッセージを送信する STARS Client は、「送信元」及び「>」（例では「Term1>」）の部分を省略する事が出来る。この時、STARS Server は宛先の STARS Client に送信するメッセージに「送信元」及び「>」を自動的に付加する。

STARS では以上のようなメッセージの送受を行うが、更に「コマンド or リプライ or イベント」の部分の開始文字に以下のような規定を設けている。

- 「@」（アットマーク）から始まるもの => コマンドに対するリプライ
- 「_」（アンダースコア）から始まるもの => イベント情報
- それ以外の文字から始まるもの => コマンド

以上のような規定を設ける事で、STARS では「コマンド及びリプライによる動作」、「イベントドリブン型の動作」の両方とも扱えるようにしている。

STARS Client 接続時のセキュリティーチェック

STARS では、STARS Client が STARS Server に接続する際、簡単なセキュリティーチェックを行う事で、関係の無い者が勝手に STARS Server に接続する事や、誤って別のサーバに接続してしまう事が無いようにしている。なお、STARS における接続時のセキュリティーチェックは以下の 3 段階に分かれている。

1. 予め登録された PC からの接続であるかのチェック：ホスト名あるいは IP アドレスをチェックし、接続を要求してきた PC が予めリストにあるものかどうかを確認、リストになければ接続要求を却下する。

2. キーワードによるチェック（図 2 参照）：

各々の STARS Client は独自のキーワードリストを持つ事になっており、接続しようとする STARS Client については、そのキーワードリストのコピーを予め STARS Server に登録しておく。STARS Client が STARS サーバに接続してくると、STARS Server はランダムな番号を送ってくる。この番号はキーワードリスト中の「何番目のキーワードを使用するか」を示しているのので、STARS Client は自分の Node Name と指定された番号に合致するキーワードを付けて STARS Server に送信する。STARS サーバはこの Node Name とキーワードをチェックし、正しければ接続を許可する。

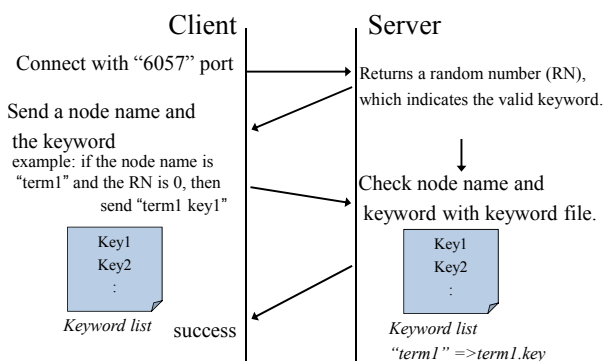


図 2. キーワードによるチェック

3. Node Name +ホスト名 (IP アドレス) のチェック (選択可) : STARS には、ある Node Name を持つ STARS Client は必ず決められた PC 上で動作している場合のみ接続を許可するよう、設定を行う事が可能である。その場合、STARS サーバは、上記 1 及び 2 のチェックを行った後、再度 Node Name とホスト名あるいは IP アドレスの確認を行う。

STARS インターフェースライブラリ

STARS Client の開発は、TCP/IP Socket 及びテキストを扱うスキルがあれば、誰でも容易に行う事が出来る。またその際、利用する開発言語及びオペレーティングシステムは任意である。STARS では更に STARS Client

開発の際に有用な「インターフェースライブラリ」を各言語用に用意する事で、開発者の利便性を高めている。これら「インターフェースライブラリ」を使用すると、TCP/IP Socketを意識することなく STARS Clientの開発を進める事が可能である。また、比較的煩雑な「接続時のセキュリティチェック」のプロセスも自動で行われるので、更に STARS Client の開発が容易となる。現在、利用可能なインターフェースライブラリは以下の通りである。

- Perl : Perl モジュール、様々な OS で利用可
- C : 様々な OS で利用可、メッセージの長さに制限あり、Windows では Call back の利用不可
- ActiveX Control : VB6 等で利用
- .NET : C#, Visual Basic .NET 等で利用可、mono を利用すれば Linux 上でも動作
- Java : Windows、Linux、Android etc.

2 STARS Java 用インターフェースライブラリ

Java 用のライブラリは最近新たに開発された。若干の仕様変更の可能性もまだあり、現在はソースコードでの供給となる。使用方法は他のライブラリとほぼ同様であり、STARS サーバとの接続のプロセス等は自動で行う事が可能である。図 3 は STARS Java 用インターフェースライブラリを利用した際に利用可能なメソッド等である。図 3 中、使用準備の項目は本ライブラリを使用するための宣言が記されている。また、「connect」は STARS Server への接続を行う為に使用する。この時、「connect」により、接続のためのキーワードチェックなどのプロセスは自動的に実行される。「send」及び「receive」は STARS Server とのメッセージの送受の為に、StarsMessageHandler 関連は、非同期受信によりイベントドリブンのプログラミングを行う為にそれぞれ用意されている。

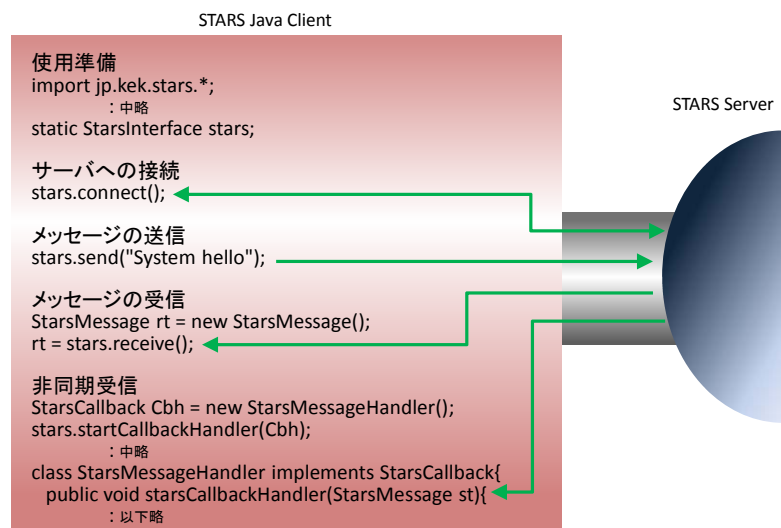


図 3. STARS Java 用インターフェースライブラリの利用

「connect」は STARS Server への接続を行う為に使用する。この時、「connect」により、接続のためのキーワードチェックなどのプロセスは自動的に実行される。「send」及び「receive」は STARS Server とのメッセージの送受の為に、StarsMessageHandler 関連は、非同期受信によりイベントドリブンのプログラミングを行う為にそれぞれ用意されている。

3 ACOP Beans

ACOP Beans はドイツ DESY により開発 (DESY が出資、Cosylab がコーディングを行った) された ACOP のライブラリで、迅速なプログラム開発を行うための、基本的な ACOP の GUI 用 Java beans を有している。ACOP はネットワークを利用したプロセス間通信を行う機能も有しており、DESY で開発された TINE (Three-fold Integrated Networking Environment) プロトコル^[6]等を利用する事が可能である。しかし、今回の STARS での ACOP の利用に関しては単に GUI 作成のためのライブラリとして使用している。なお、主な ACOP Beans には、Button、CheckBox、Dial Knob、Gauger、Icon、Label、Number Label、Slider、Table、Video、Wheelswitch などがある。

なお、図 6 は Debugger として STARS サーバに接続し、本テスト用 GUI の実行中に実際に送受されるメッセージをモニタした結果である。ACOP Slider を動かした場合や、「testdev」に GetValue コマンドを送信した際に、正しいデータの送受が行われていることが観測できている。

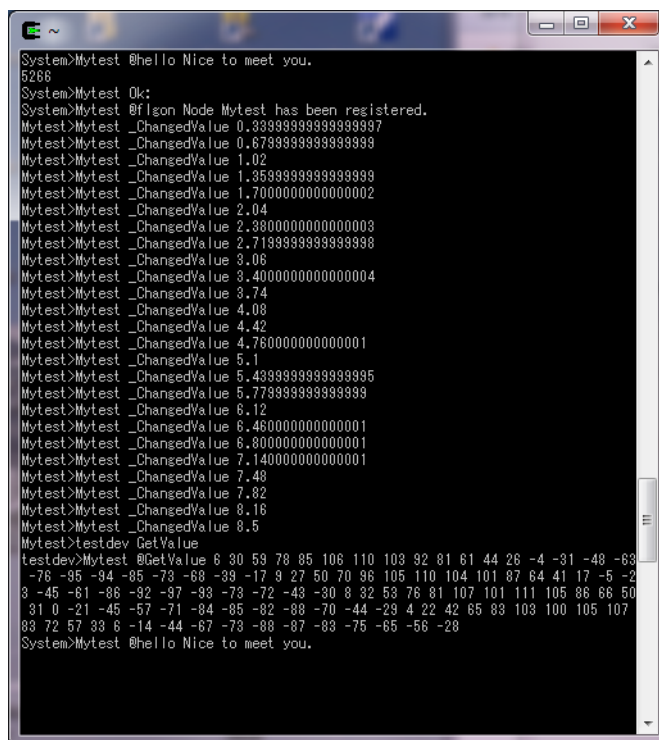
6 まとめと今後の予定

STARS Java 用インターフェースライブラリが開発されたことで、今回 ACOP Beans の利用が可能になった。また、STARS GUI Client を作成する際に ACOP Beans は非常に有用であり、ACOP Beans を利用すると効率の良い STARS Java GUI Client の開発が可能であることが分かった。

ACOP Beans は直接 TINE プロトコル等の通信プロトコルをサポートしているが、今回は ACOP Beans を単なる GUI ツールとして使用した。今後は、直接 STARS のプロトコルを ACOP Beans に組み込むなどの可能性も模索してゆく予定である。また、DESY においては .NET フレームワーク上で動作する ACOP に関しても有志により開発が進もうとしており、こちらに関しても検討及び開発の協力を進めてゆく予定である。更に STARS Java 用インターフェース及び ACOP Beans の Android 端末上での利用に関しても調査を進めてゆく。

参考文献等

- [1] T. Kosuge, Y. Saito, "RECENT PROGRESS OF STARS", Proceedings of PCaPAC2005, Hayama, Japan, 2005.
- [2] <http://stars.kek.jp/>
- [3] <http://cosylib.cosylab.com/pub/acop/site/>
- [4] 小菅隆、濁川和幸、永谷康子、斉藤裕樹、"STARS の PF ビームラインへの広がり開発状況"、平成 21 年度技術研究会（2010 高エネルギー加速器研究機構）
- [5] 濁川和幸、小菅隆、斉藤裕樹、永谷康子、渡邊一樹、"KEK-PF での STARS の現状"、平成 22 年度技術研究会（2011 熊本大学）
- [6] <http://tine.desy.de/>



```
System>Mytest @hello Nice to meet you.
5268
System>Mytest Ok;
System>Mytest @f1son Node Mytest has been registered.
Mytest>Mytest _ChangedValue 0.33999999999999997
Mytest>Mytest _ChangedValue 0.67999999999999994
Mytest>Mytest _ChangedValue 1.02
Mytest>Mytest _ChangedValue 1.3599999999999999
Mytest>Mytest _ChangedValue 1.7000000000000002
Mytest>Mytest _ChangedValue 2.04
Mytest>Mytest _ChangedValue 2.3800000000000003
Mytest>Mytest _ChangedValue 2.7199999999999998
Mytest>Mytest _ChangedValue 3.06
Mytest>Mytest _ChangedValue 3.4000000000000004
Mytest>Mytest _ChangedValue 3.74
Mytest>Mytest _ChangedValue 4.08
Mytest>Mytest _ChangedValue 4.42
Mytest>Mytest _ChangedValue 4.7600000000000001
Mytest>Mytest _ChangedValue 5.1
Mytest>Mytest _ChangedValue 5.4399999999999995
Mytest>Mytest _ChangedValue 5.7799999999999999
Mytest>Mytest _ChangedValue 6.12
Mytest>Mytest _ChangedValue 6.4600000000000001
Mytest>Mytest _ChangedValue 6.8000000000000001
Mytest>Mytest _ChangedValue 7.1400000000000001
Mytest>Mytest _ChangedValue 7.48
Mytest>Mytest _ChangedValue 7.82
Mytest>Mytest _ChangedValue 8.16
Mytest>Mytest _ChangedValue 8.5
Mytest>testdev GetValue
testdev>Mytest @GetValue 6 30 59 78 85 106 110 103 92 81 81 44 26 -4 -31 -49 -83
-76 -95 -94 -85 -73 -68 -39 -17 9 27 50 70 96 105 110 104 101 87 64 41 17 -5 -2
3 -45 -61 -86 -92 -97 -93 -73 -72 -43 -30 6 32 53 76 81 107 101 111 105 86 66 50
31 0 -21 -45 -57 -71 -84 -85 -82 -88 -70 -44 -23 4 22 42 65 83 103 100 105 107
83 72 57 33 6 -14 -44 -67 -73 -88 -87 -83 -75 -65 -56 -28
System>Mytest @hello Nice to meet you.
```

図 6. TEST 用 GUI のメッセージ送受の様子