

E4

Java 言語による Internet Lidar System (ILIS) の Graphical User Interface (GUI) 化の検討

Internet Lidar System (ILIS) improved by using Java language for Graphical User Interface

相原英典¹、浅井和弘¹、青木哲郎²、水谷耕平²、板部敏和²

Hidenori Aihara¹, Kazuhiro Asai¹, Tetsuo Aoki², Kouhei Mizutani², Toshikazu Itabe²

¹東北工業大学、²独立行政法人通信総合研究所

¹Tohoku Institute of Technology, ²Communications Research Laboratory

Abstract: For achieving an unattended lidar system, we have already proposed the new generation of lidar, which was named the Internet Lidar System (ILIS), by combining IT technologies with lidar technologies. The first ILIS was successfully realized by using a commercial product's software for the C language, which was supported by Windows as Operating System (OS). This system has a shortcoming, i.e. clients have to use same software operating on same OS. Java language has many advantages, which are an independent software against any OS and very useful in the Internet and LAN.

In this paper, we propose a new generation of ILIS improved by using the Java language for accomplishing more graphical user interface.

1. はじめに

ライダーは、大気科学の研究にとって欠かすことの出来ない大事なリモートセンシング技術である。我々は、数年前に IT 技術とライダー技術を融合させ、“インターネットを介してライダー装置を遠隔制御が可能な Internet Lidar System(ILIS)”を提案し、その実現に向けて研究を行って来ている。¹⁻³⁾

初期の ILIS 用プログラミングは、Borland C++ Builder 3 により構築されたため、スタンドアロン・ソフトウェアとして動作した。そのため ILIS の実行には、専用ソフトウェアのインストールが必要であり、さらに、プログラミング時に使用した Operating System(以下、OS と略す)に依存するので、当然の事ながら特定の OS が要求された。

一方、インターネットおよび WWW ブラウザは現在も急速な発展を遂げており、これに併せて Java はインターネット上またはネットワーク上でのアプリケーションを開発する事が出来る言語として注目を浴びている。さらに、Java プログラムは OS のプラットホームに依存せず、且つブラウザ上で実行可能である。ILIS の究極的な目標は、遠隔地にあるライダー装置を“いつでも”・“どこからでも”・“だれでも”動作させられる環境作りにある。

そこで、我々はネットワーク対応言語である Java によるプログラミングにより、ブラウザ上で実行可能な ILIS の構築に関する研究に着手したので進捗状況を踏まえ報告する。

2. Java による ILIS(Internet Lidar System)の構築

これまでの ILIS は、Borland C++ Builder により作成された専用ソフトウェアでのみ遠隔制御されていた。クライアント(東北工大 or C.R.L)は専用ソフトウェアにより HTTP プロトコルを介しライダーサーバ(北海道、陸別)へコマンドを送る。ライダーサーバは送られたコマンドに対して GPIB、リレーボードなどの制御を C 言語によって行う。しかし、C 言語によるプログラミングは、OS のプラットホームに依存したコードにコンパイルされるため、実行処理速度は高速であるがクライアントは特定の OS のプラットホームに制限されていた。

一方、Java によるプログラミングの特長は、コンパイル時の OS のプラットホームに依存し

ない中間コードのバイト形式のアプレットコードとして表すことができる。すなわち、プログラム開発者はクライアントのあらゆる OS に対する依存性を考慮することなく、プログラムを構築することができる。また、アプレットコードは Java インタプリタにより実行され、Java Virtual Machine(以下、Java VM と略す)をソフトウェア的に実現することができ、Java VM は WWW ブラウザによって具現化することができる。その結果、クライアントが WWW ブラウザを用いてインターネットに接続できるならば、クライアントは OS の依存性にとらわれることなく、“いつでも”・“どこからでも”・“どんな OS からでも”ライダーシステムを遠隔制御することが可能なバーチャル・ラボラトリーを実現することができる。その一方で、Java は WWW ブラウザ上で実行するために任意のプロセッサが理解できるコードへ変換するため、特定のプロセッサ向けに作られたプログラムよりも低速(C 言語に比べ約 10~20 倍)である。また C 言語では可能であった GPIB およびリレーボードの制御は Java 言語では実現できない。しかし、GPIB、リレーボードのコントロールは Java Native Interface(以下、JNI)によって制御することができる。Table.1 は C 言語と Java 言語の特徴を比較したものである。我々はネットワーク対応言語である Java に着目し、次世代 ILIS の研究を行った。

Table.1 Comparison of C programming with Java programming

	OS への依存性	ハードウェア制御	実行処理速度
C 言語	<ul style="list-style-type: none"> OS プラットホームへの依存性強い 実行画面はすべて等しい 	<ul style="list-style-type: none"> GPIB、リレーボード等の制御可能 	<ul style="list-style-type: none"> 特定のプロセッサ用にコンパイルされているため高速
Java 言語	<ul style="list-style-type: none"> OS プラットホームへの依存性なし 実行画面の違い(ボタンのどのコンポーネントの配置) 	<ul style="list-style-type: none"> GPIB、リレーボード等の制御不可能 JNI の使用により制御可能 	<ul style="list-style-type: none"> 不特定のプロセッサに対してコンパイルされているため低速 ブラウザの発展によりプロセッサの区別ができ高速化

ILIS は、Fig.1 に示されるようにクライアント、サーバ(ライダーサーバ、カメラサーバ、ウェザーサーバ)、インターネットで構成されている。Fig2 はライダーシステムのブロックダイアグラムを表している。

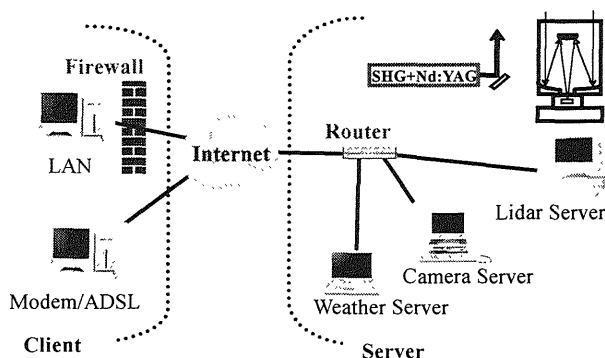


Fig1. Concept of ILIS

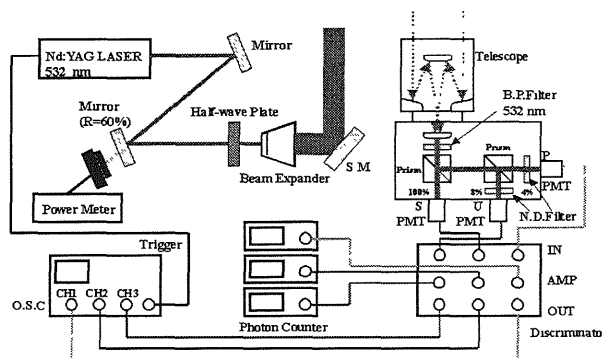


Fig2. Lidar Server

また、各デバイスへの電源供給および制御は GPIB、リレーボードにより制御されており、これらの構成は Fig.3 に各デバイスとともに示されている。ライダー・サーバ・ソフトウェアおよびクライアント・ソフトウェアは Sun Microsystems Java2 JDK1.3(OS:Windows98)によって構築された。

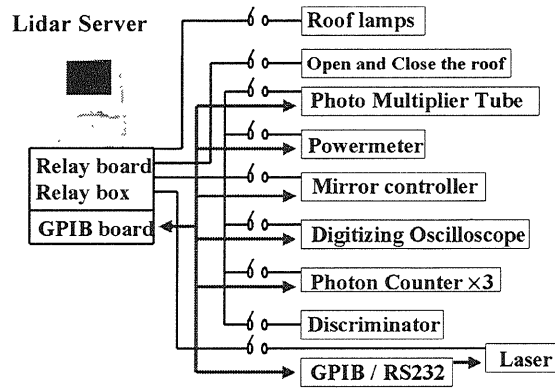


Fig.3 Lidar Server

ライダー・サーバ・ソフトウェアは、クライアントがインターネットへの接続法に関係なく WWW ブラウザによって任意のファイルを要求、取得可能とするために、WWW サーバとして稼働している。また GPIB、リレーボードの制御は、Java のみでは実現することができないため JNI を使用した。JNI は、GPIB ならびにリレーボードの制御を C 言語によってプログラミングされた Dynamic Link Library(以下、DLL)ファイルをリンクしこれらを制御できる。DLL ファイルは Microsoft Visual C++6.0 によって作成された。また、マルチスレッドを用いているため、デバイスの制御とデータファイルの保存など、複数の処理は同時に実行することができる。これは、ネットワーク間の処理時間を最小限に抑え、ILIS をよりリアルタイムに実行するための手法である。Fig.4 はサーバ・ソフトウェアを示している。

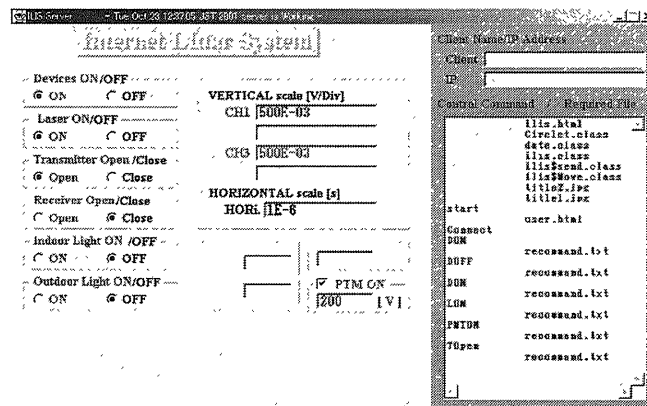


Fig.4 Software of Lidar Server

クライアント・ソフトウェアは WWW ブラウザによってアプレットコードをダウンロードし、Java VM を具現化するアプレットとして構築されている。アプレットコードは WWW ブラウザを用いて HTTP プロトコルを介しているため、インターネットへの接続形態を問わずダウンロードし、実行することができる。このアプレットもまたサーバ同様、複数の処理を同時に行うためマルチスレッドを用いている。クライアントはサーバに対して、制御コマンドを送り、処理が終了したことを示すステータスレポートコマンドを取得する。これと同時にスレッド処理により、オシロスコープの波形データ等を取得して表示する。Fig.5、Fig.6 はアプレット実行画面を表しており、Fig.7 は LAN 内部からのクライアントとサーバ間のコマンド、データの送受信および制御に関するフローチャートを示している。

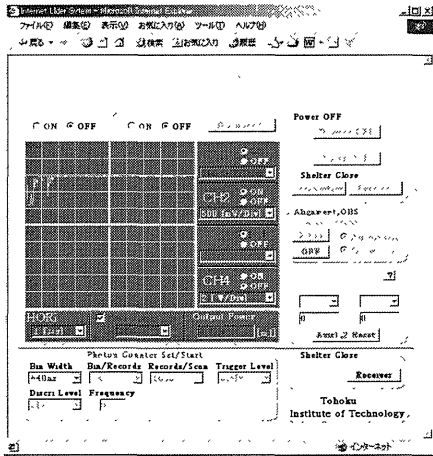


Fig.5 Applet using Internet Explorer 5.5 and OS: Windows98

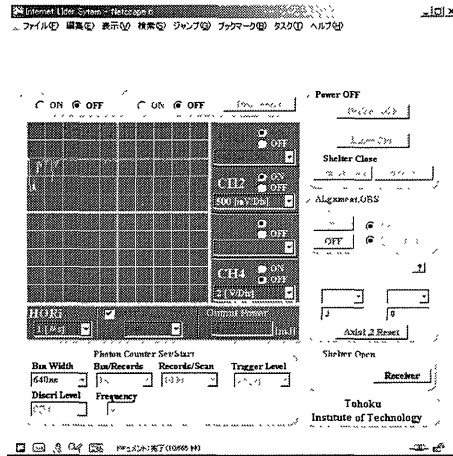


Fig.6 Applet using Netscape 6 and OS: Windows2000

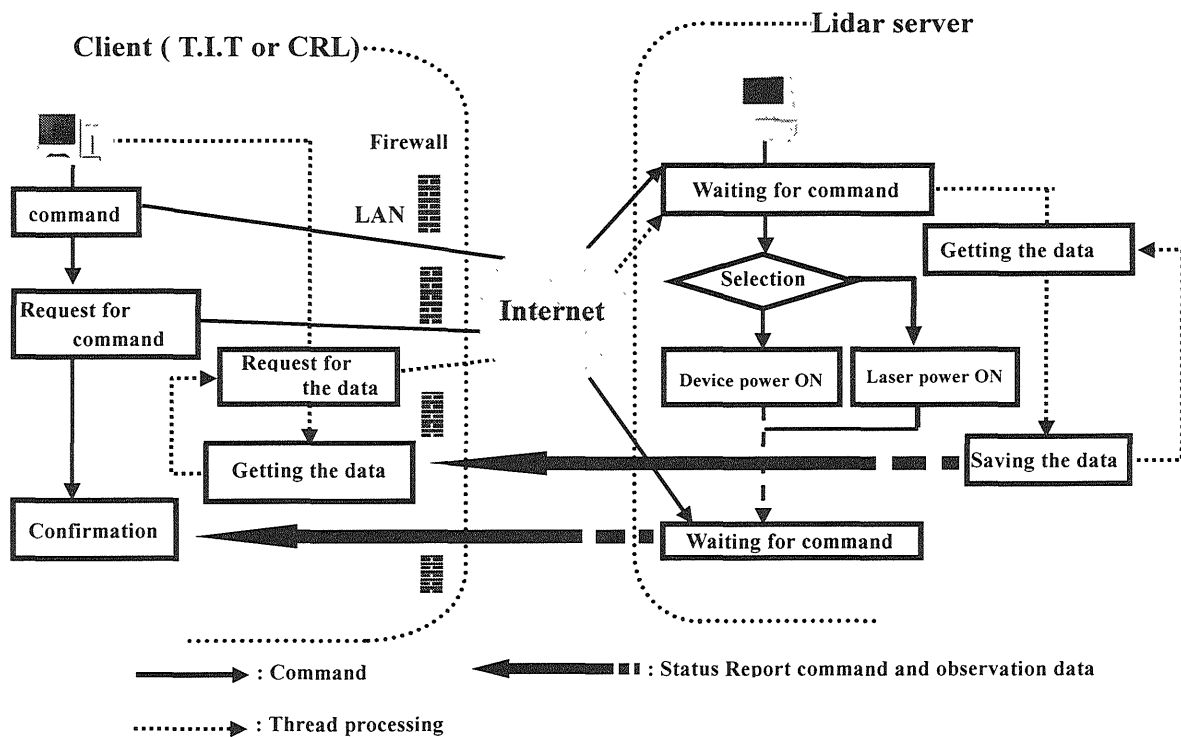


Fig.7 Flow chart using the Tread technology

3. おわりに

現在、JNI プログラミングによるライダーサーバと各種 GPIB ボード間とのコミュニケーションが完全には確立されておらず、目下バグ取りを含め改良中である。詳細は学会にて報告します。

参考文献

- 1)菅田,十束,浅井,水谷,板部,“第 19 回レーザセンシングシンポジウム予稿集” P.49-50 (1998)
- 2)T.Sugata, K.Asai, T.Aoki, K.Mizutani, T.Itabe, “ILSS’99 20th Japanese Laser Sensing Symposium Abstracts of Papers” P.135-136 (1999)
- 3) H.Aihara, K.Asai, T.Aoki, K.Mizutani, and T.Itabe, “Internet Lidar System (ILIS): New generation of lidar system using Internet technology for realizing unattended lidar”, Proceeding of SPIE, Lidar Remote Sensing for Industry and Environment Monitoring, Vol.4153, P.50-55, Oct., 2000