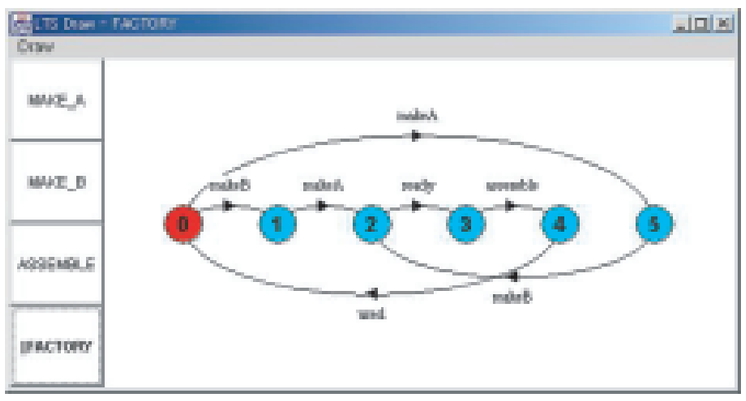
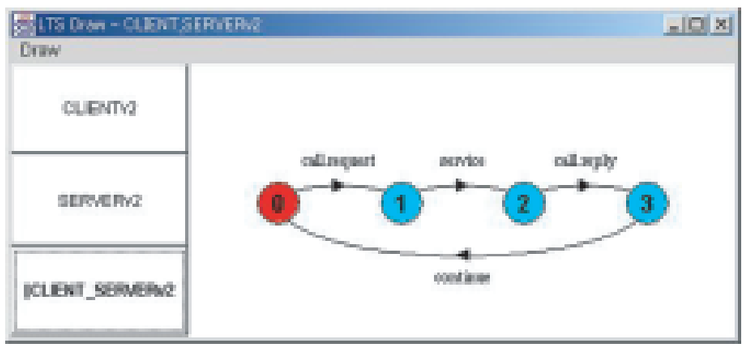
モジュールベース開発向けプロセス代数の提案

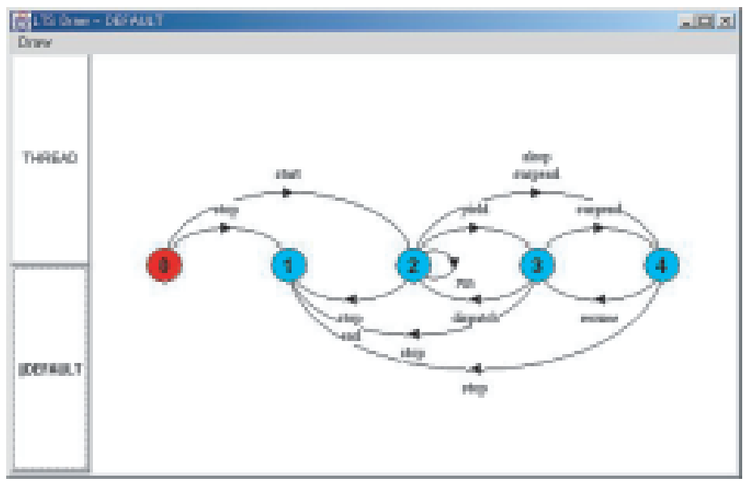
NE13-0203G 三船　真哲

研究の背景と目的

IC タグ、ネットワーク家電など、ユビキタス社会の到来に伴い、各機能モジュールを個々に設計し、組み合わせてシステム全体の設計を行う必要性が高まっている。しかし、組み合わせたシステムの仕様を検証する有効な手法は、十分に研究されているとは言えない。本研究では、プロセス代数の一種である FSP（finite state processes）を拡張し、モジュールベース開発に最適な記述技法と、該当システムの動作検証を行う方法を提案する。FSP は、プロセス代数に分類されるプロセス記述用言語である。また FSP で記述したプロセスは、LTSA というツールを用いて、並行動作を含めたプロセスの動作検証を行うことができる。







概要

以下は簡略化したナビとエンジンの組み合わせモデルを FSP で記述した例である。

```
ENGINE = (e_on -> e_off -> ENGINE).
TV = (pwr_on -> TV_READY).
TVREADY=(pwr_off -> TV|| t_on -> t_off -> TVREADY)
||ENGINE_TV = (ENGINE || TV) / {on / e_on , off / e_off , on / pwr_on , off / pwr_off}.
```

上記等式は、左辺がプロセス名、右辺がそのプロセスの動作を表している。ENGINEは、エンジン始動(e_on)、エンジン停止(e_off)という動作を実行した後、ENGINEを再帰的に呼び出す（つまりe_off→e_on…と続く。TVも同様）。この2つのサブシステムを組み合わせる場合、動作を同期させる必要がある（エンジンの始動以前にテレビが点くことは現実的に有り得ない）。この場合共有動作を設けるのはテレビのON/OFFとエンジンのON/OFFの部分である。FSPでは、2つの動作を同期させるために、動作の名前を同じものにする。 ||ENGINE_TVでは、ENGINEとTVの各動作ラベルをonとoffに書き換えている。動作のラベルを書き換えた上で、2つのサブシステムを統合すると||ENGINE_TVの式のようになる。

本研究では新たに「トリガ」という文法をFSPに加えて定義した。トリガは、プロセスに共有動作を定義するのではなく、動作間の順序関係を明示的に示すものである。前述のモデルでは||ENGINE_TVで2つの動作を同期させていたが、トリガを用いることによって共有動作を記述することなくプロセスを統合することができる。

```
ENGINE = (e_on -> e_off -> ENGINE).
TV = (pwr_on -> TV_READY).
TVREADY=(pwr_off -> TV|| t_on -> t_off -> TVREADY).
||ENGINE_TV = (ENGINE || TV).
ENGINE.e_on ! TV.pwr_on.
TV.pwr_off ! ENGINE.e_off.
```

上記等式はトリガを定義して記述した、ナビとエンジンの組み合わせモデルである。トリガは、xがプロセスPの動作で、プロセスQの動作qのトリガとするとき、P.x!Q.qと表現する。このように記述することによって、共有動作で記述するのに比べ、統合する際の制約条件を容易に把握することができる。