
Concurrent processes and their description

並行プロセスの挙動とその記述

慶應義塾大学名誉教授
徳田英幸

ちょっと復習。。。。

基本的な概念：自律分散協調

- 自律性
 - 個の確立
 - 主体的行動
- 分散性
 - 多数の個
 - 空間的・ネット的に分散
- 協調性
 - 個と個の協調プロトコル
 - 協調により全体の機能を維持・形成する
 - 構成論的手法 vs. 自己組織論的手法
- システムとしての評価
 - 評価の軸
 - 良いシステム vs. 悪いシステム

What is an Autonomous Distributed Cooperative System (ADC System) ?

- ❑ No supervisor which can control/manage the entire system
システム内にシステム全体を制御／統治するスーパーバイザは存在しない。
- ❑ The system consists of autonomous, distributed and cooperative subsystems
各サブシステムは、自律、分散した構成要素からなる。
- ❑ The system functions are realized by cooperations among subsystems
全体のシステムの機能は、サブシステム間の協調作業によって遂行される。

プロセス間通信: Direct IPC

- The communication is done between two processes without using any communication entity.

Bsend(process_id, message, size)
Nsend(process_id, message, size)

Brec(process_id, buffer, size)
Nrec(process_id, buffer, size)

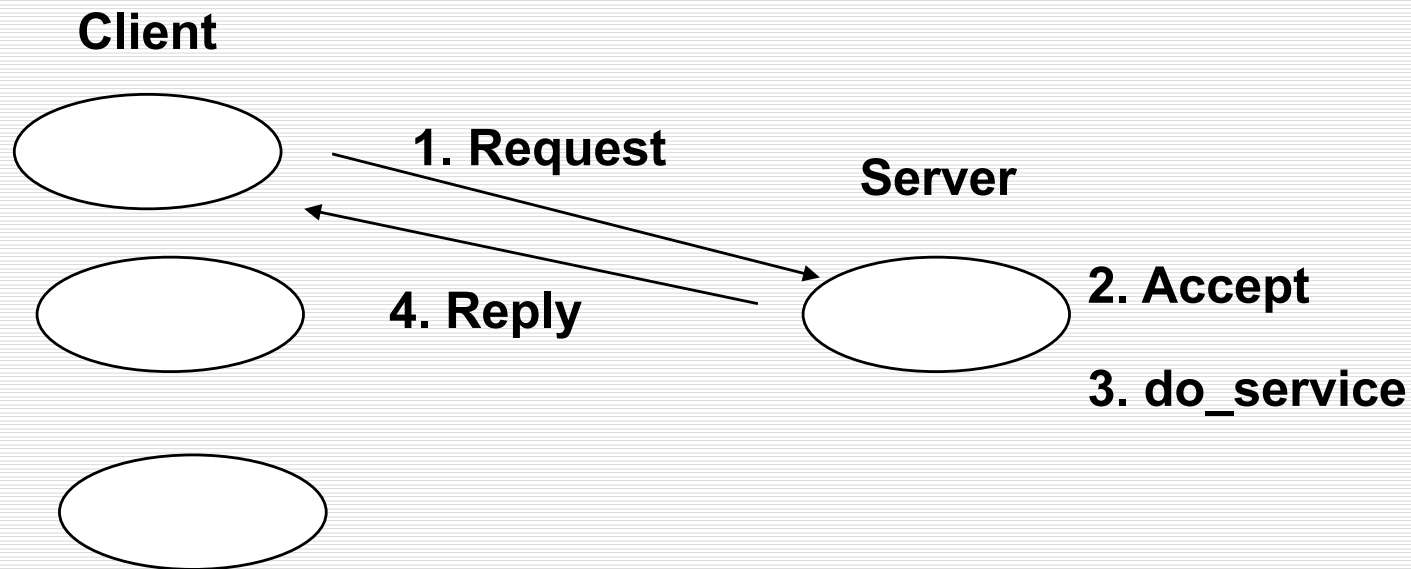
process_id = Brecany(buffer, size)
process_id = Nrecany(buffer, size)

**Request(process_id, message, sizeof(message),
buffer, sizeof(buf))**

Reply(process_id, message, size)

Client/Server Model

- Client/Server Model is a well-known model.
- A client (i.e., a user process) sends the request message to a server process, which then does the work and sends back the results.



プロトコルの記述

～N-way Protocols～

プロトコルとは？

～協調動作の記述～

プロトコルとは？

- 外交上の儀礼
- 通信の送信側と受信側で取り決めた約束事
- 通信手段

pro·to·col

1 U(外交上の)儀礼, 典礼.

2 C 条約原案; 議定書, プロトコル.

3 C (国家間の)協定.→#

4 C 《米》(実験・治療の)実施要綱[計画].

5 C 【電算】プロトコル《データ通信の手順》.

[株式会社研究社 新英和・和英中辞典]

プロトコルの記述

- プロトコルの記述は、通信上の約束事すべてを定義する。
- 通信メッセージのフォーマット(書式)の詳細 (format) -> (syntax)
- メッセージを交換する際の手順 (procedure) -> (grammar)
- 正しいメッセージが表わしている意味、語彙 (correctness) -> (semantics)
- Completeな記述 vs. Incompleteな記述

何のためにプロトコルが必要か？

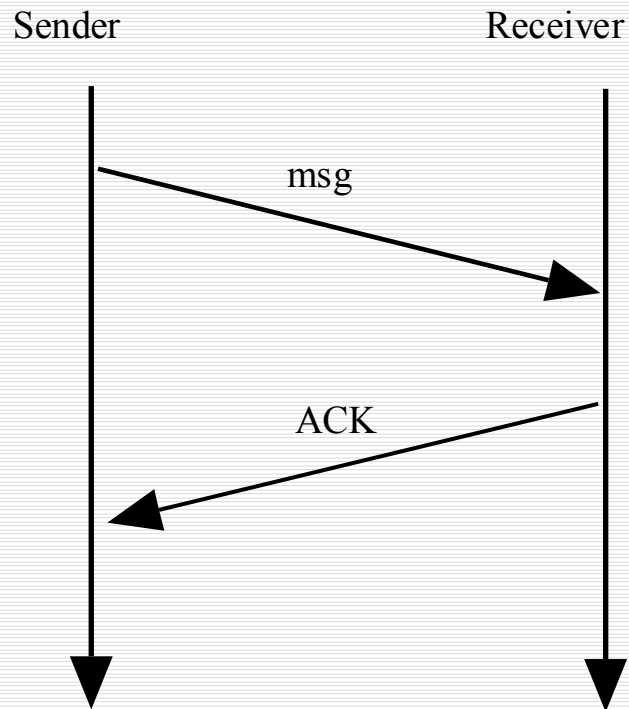
- データ転送に必要な初期化や終了
- 送信者と受信者との同期(条件)
- 転送エラーの検出や訂正
- データの書式や符合化を行う

Protocolの記述

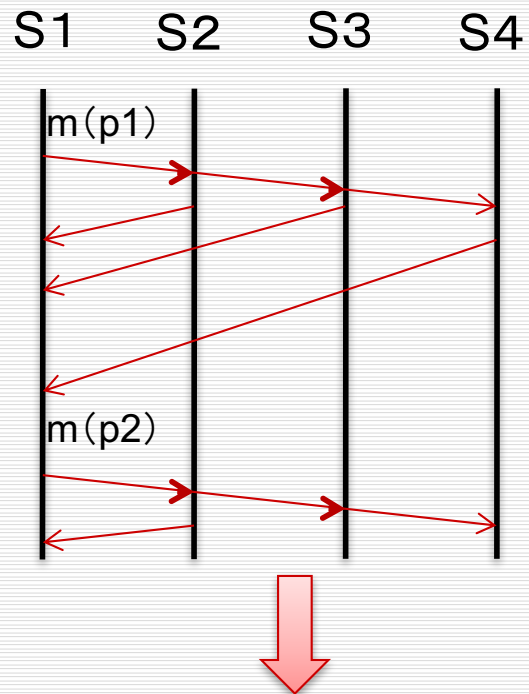
- ☐ 自然言語
- ☐ Time-space chart
- ☐ 状態遷移図
- ☐ 擬似Prog. Languages

Time-Space Chart

- ・プロトコルのデザインに適しているが、messageのフォーマットやセマンティクスは別に記述する必要がある



入札プロトコル(1)



Q: 終了条件は？
winnerへの通知は？

Q: タイムアウトは？
全員がいつもタイムリーに返信？

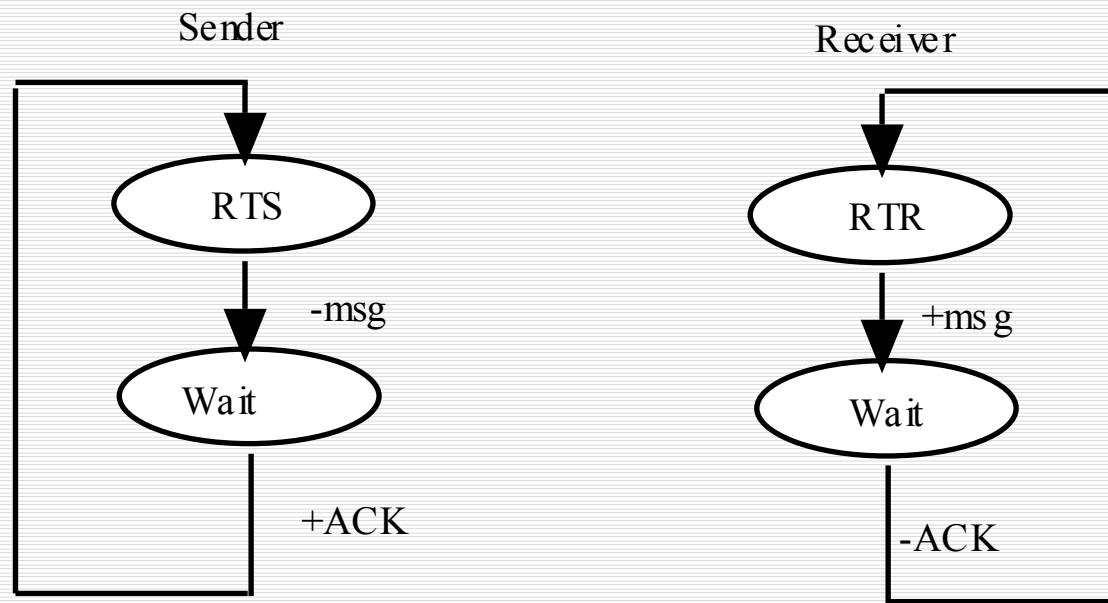
Note: S1からのグループキャスト
は、ユニキャストと区別するために
各受信者のところで矢印マークを入
れる



State Transition Diagram 状態遷移図

ノード: プロセスの状態

エッジ: エッジには、-msg (msgを送信)、+msg(msg)といったようにメッセージ名だけでなく、送受信の区別を<+><->で明記



Concurrent Programming Language

```
Process Sender ( )
{
    while(TRUE) {
        prepare_message(buffer);
        frame.data = buffer;
        send_to_network(frame);
    }
```

```
Process Receiver ( )
{
    while(TRUE) {
        wait_for_frame(buffer);
        receive_from_network( frame);
        buffer = frame.data;
    }
```


Stop-and-Wait Protocol

Process Sender ()

```
{  
    while(TRUE) {  
        prepare_message(buffer);  
        frame.data = buffer;  
        send_to_network(frame);  
        wait_for_frame(ack);  
        . . .  
    }  
}
```

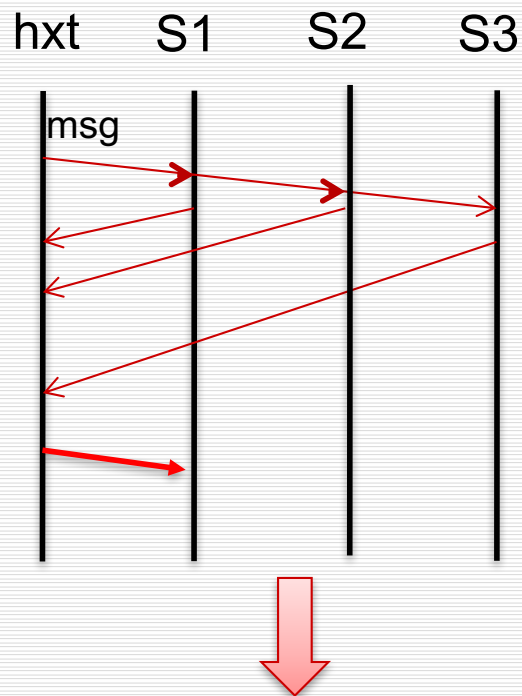
Process Receiver ()

```
{  
    while(TRUE) {  
        wait_for_frame(buffer);  
        receive_from_network( frame);  
        buffer = frame.data;  
        prepare_message(ack);  
        frame.data = ack;  
        send_to_network(frame);  
    }  
}
```

いくつかの例題

- ☐ 回覧板プロトコル
- ☐ 入札プロトコル
- ☐ 仮想リングプロトコル
- ☐ 逆オークションプロトコル

回覧板プロトコル



条件：送信者から全ての受信者にメッセージが高々1回伝わる。確認のマークを追記。最後の受信者は、送信者にメッセージを返却。

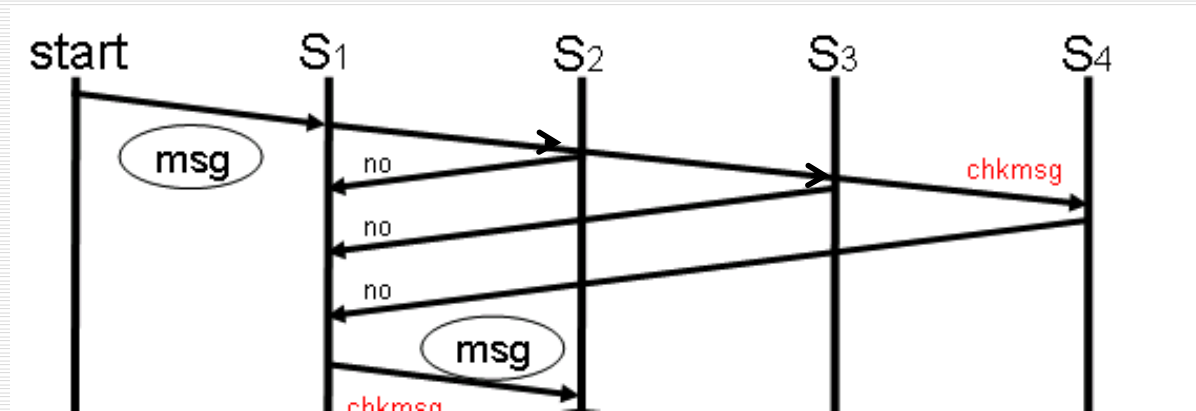
Q: メッセージのフォーマットは？
何が書いてあるべきか？

Q: タイムアウトは？
全員がいつもタイムリーに返信？

Note: hxtからのグループキャストは、ユニキャストと区別するために各受信者のところで矢印マークを入れる

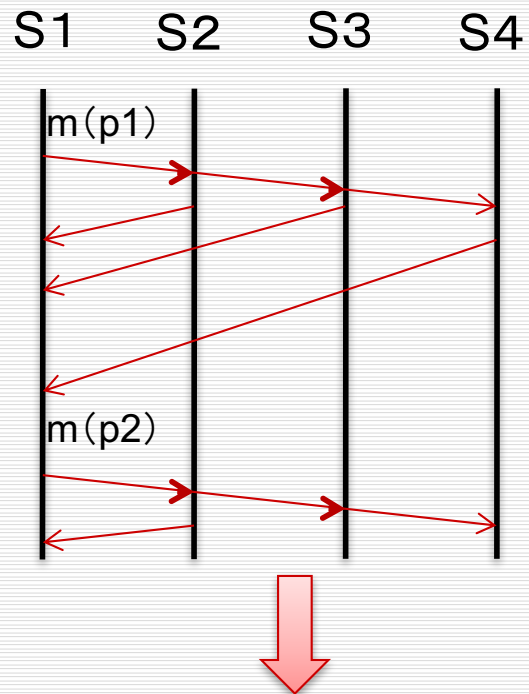


回覧板プロトコル(1)



[illegible]

入札プロトコル(1)



Q: 終了条件は？
winnerへの通知は？

Q: タイムアウトは？
全員がいつもタイムリーに返信？

Note: S1からのグループキャスト
は、ユニキャストと区別するために
各受信者のところで矢印マークを入
れる

