この研究集会の目的

自作他作をとわずソフトウエアの, みせっこをする.

OpenXM の仕様その他について議論する.

おもしろそうなソフトを一緒に実装開始する.

おもしろそうなソフトをインストールしてみる.

私個人の目的: 級数型. Simplify. OX-RFC-104 (OoHG). 院生. 次期 knoppix/Math. Contributed software ないかな. その他.

Q. OpenXM の目標は?

- 数学での並列計算,
- 数学ソフトウエアの統合化 または Conglomerate 化 (A.Solomon)
- 数学的知識のマネージメント (Mathematical Knowledge Management)
- 実際に数学の研究や数学の応用に使えるパッケージの開発

Q. どうやってためす?

`http://www.openxm.org` よりソースを download.
asir-contrib

`knoppix/math`

Q. 開発の歴史は?

- 歴史の始まり: 1997 年頃.
- 1.1.1 (January 24, 2000): 最初の実験版. XMLがだんだん盛んに
- 1.1.2 (March 20, 2000): とりあえず使える版.
- 1.1.3 (September 26, 2000): 1.1 系の最終版. OpenXM RFC 100 形式のプロセス木. 1077 個の数学関数を提供. 提供しているサーバは `ox_asir`, `ox_sm1`, `ox_phc`, `ox_gnuplot`, `ox_m2`, `ox_tigers`, `ox_math`(ematica), `OMproxy`. WSDL の定義(2001)
- 1.2.1 (March 2, 2002): Cygwin (Windows) への対応開始. マニュアル自動生成(gentexi)など. RFC101
- 1.2.2 (May 13, 2003): RFC102, RFC103 (部分的), autoconf (•) OpenXM/fb, Digital formula book, OpenMath, tfb, CD (hypergeo*, weylalgebra*, intpath*), 数値計算.
- 1.2.3 (February 10, 2005): RFC 104 (部分的web対応): polymake 等, asir-contrib の充実, knoppix への対応が軌道に乗る.

Q. RPC って何?

Remote Procedure Call. 他のプロセスとか他の機械のプロセスをサブルーチンみたいに呼ぶこと.

`man rpc, callrpc`

```
[1247] P=ox_launch(0,"ox_asir");
[1248] ox_cmo_rpc(P,"fctr",x^3-1);
0
[1249] ox_pop_cmo(P);
[[1,1],[x-1,1],[x^2+x+1,1]]
```

- Corba, Interface Definition Language (IDL) は言語に依存しない関数のプロトタイプ宣言みたいなもの. cf. w3m (MS, IBMの) の WSDL.

- Java, RMI (Remote method invocative).

- OpenXM, RFC100の CMO.

Q. Web 技術にのせた RPC の仕組みはありますか?

SOAP (Simple Object Access Protocol, w3c と MS), XML RPC, IAMC (数学用: Internet Accessible Mathematical Computation, P.Wang ら, 米国/中国, 1990 年代後半より), Web Mathematica (数学用, 2000 年代始め),

```
telnet www.math.kobe-u.ac.jp 80
GET /index.html  HTTP/0.9      newline を二回.

HTTP/1.1 200 OK
Date: Sat, 12 Feb 2005 04:13:15 GMT
Server: Apache/1.3.27 (Unix) PHP/4.3.3RC1
途中略.
Content-Type: text/html

<HTML>
<HEAD>
以下略.
```

SOAP のメッセージ例. http://www.atmarkit.co.jp/fxml/tanpatsu/02soap/soap03

```
POST /StockQuote HTTP/1.1
```

```
略
SOAPAction: "Some-URI"

<SOAP-ENV:Envelope  封筒
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>     中身
 <m:GetLastTradePrice xmlns:m="Some-URI"> 終り値はいくらですか?
  <symbol>DIS</symbol>
 </m:GetLastTradePrice>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

返事
HTTP/1.1 200 OK
Content-Type: text/xml;
略
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
<SOAP-ENV:Body>
<m:GetLastTradePriceResponse xmlns:m="Some-URI">
 <Price>34.5</Price>              34.5 円だよ.
</m:GetLastTradePriceResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

HTTP には session の概念はない．一回きり．

- `Set-Cookie:`
- Java Servlet
- php，持続的データベース接続． pfsockopen，ingress_pconnect cf. `php.gr.jp` 親と子供がいる．fork して接続を保つ．
- XML RPC どこに定義?

[RFC1945] Hypertext Transfer Protocol – HTTP/1.0. T. Berners-Lee, R. Fielding, H. Frystyk. May 1996. (Format: TXT=137582 bytes) (Status: INFORMATIONAL)

[RFC2660] The Secure HyperText Transfer Protocol. E. Rescorla, A. Schiffman. August 1999. (Format: TXT=95645 bytes) (Status: EXPERIMENTAL)

Q. 言語に依存しない関数宣言 (関数仕様記述), データ型/オブジェクトの記述言語はないか?

Corba IDL,
WSDL (2001): Web Service Description Language.

`http://www.microsoft.com/japan/developer/workshop/xml/general/wsdl.asp`
に定義の訳がある. (2001)

`http://www.atmarkit.co.jp/fxml/tanpatsu/21websvc/websvc04.html`

- プログラムの一部の自動生成を目指す.
- wsdl:type, データ型の定義に相当.
- wsdl:message, 関数引数および戻値の宣言に相当.
- wsdl:portType, wsdl:operation, 関数名に相当.
- port や method (抽象的) を 具体的なものに bind する.

OpenMath/monet project.

MSDL, Mathematical service description language. “MSDL より WSDL へ” というアプローチ.

monet broker.

Q. 数学に特化した WSDL 相当のものはありますか?

Monet/OpenMath project.

`http://www.orcca.on.ca/MONET/`

XML, WSDL, UDDI(どこに何があるかを記述) を基礎として数学に特化した3つの技術を加えている.

OpenMath と MSDL (Mathematical Service Description Language) と Monet broker

(tan(x) の) 不定積分の例題を解くサイト. MSDL の記述.

```
<definitions targetNamespace="http://www.orcca.on.ca/MONET/samples/msdl"
             xmlns="http://monet.nag.co.uk/monet/ns">

 <service name ="IndefIntService">

  <classification>
    <taxonomy taxonomy="http://gams.nist.gov" code="Gams0"/>
    <problem href="http://monet.nag.co.uk/problems/indefinite_integration"/>
    <format>http://www.openmath.org</format>
    <directive-type href="http://monet.nag.co.uk/owl#evaluate"/>
  </classification>
```

```
 <implementation>
   <software href="http://monet.nag.co.uk/owl#Maple9"/>
   <hardware href="http://monet.nag.co.uk/owl#PentiumSystem"/>
 </implementation>
```
以下略

これからOpenMath encoding を用いた WSDL を自動生成しているようにみえるが本当か...

OpenXM RFC 104. OoHG (OpenXM RFC 100 over HTTP Get)

- OpenXM-RFC-100 通信を http にのせる．高速分散計算のための通信から http get による WAN 通信まで Scalability (伸縮自在性) のあるプログラムが書ける．OpenXM RFC 100 は HTTP GET と整合性が高い．理由は "サーバは控え目である" という設計方針だから．

- インタフェースの抽象化, 仕様記述 について．ここまで厳密にやらなくても, OpenXM-RFC-100/CMO と自然言語による仕様記述と WSDL/UDDI の考えの一部取り込みで十分じゃないの? と思うが, それをやってみせなきゃ...

Q. OpenXM RFC 100 プロトコルの概要は?

`http://www.math.kobe-u.ac.jp/OpenXM/Current/OpenXM-RFC.html` に定義あり.

Data:

TAG	BODY

二つの層: OX message 層. コマンド, CMO, その他の形式によるデータの層.

例 1:

```
P = ox_launch(0,"ox_sm1");
ox_push_cmo(P,1);
ox_push_cmo(P,1);
ox_execute_string(P,"add");
ox_pop_cmo(P);
```

CMO は OpenMath とおなじように XML を用いた数学データのエンコーディング法.


OX_DATA
CMO_ZZ
1
OX_DATA
CMO_ZZ
1
OX_DATA
CMO_STRING
add
OX_COMMAND
SM_executeString
OX_COMMAND
SM_popCMO

Asir-contrib の上で cmo を XML 形式でみるには?

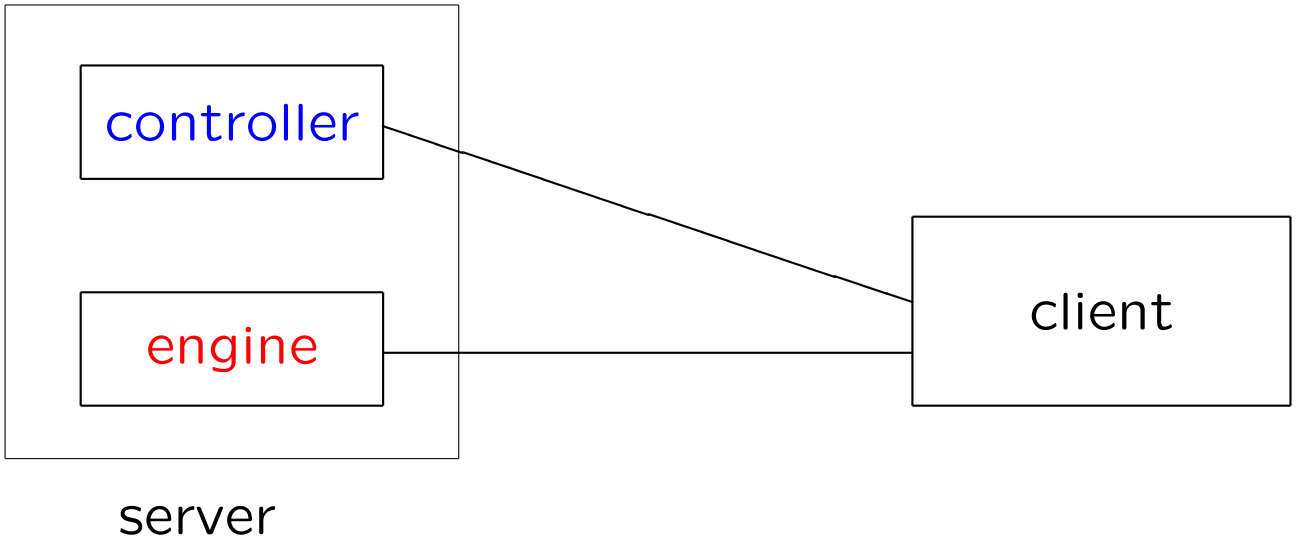
例: `taka_cmo100_xml_form(x+1)`

```
<cmo_recursive_polynomial>
   <cmo_list><cmo_int32 for="length">1</cmo_int32>    <cmo_string>"x"</cmo_string>
</cmo_list>
<cmo_polynomial_in_one_variable><cmo_int32 for="number of terms">2</cmo_int32> <cmo_int3
   <cmo_int32 for="exponent">1</cmo_int32>     <cmo_zz>1</cmo_zz>


   <cmo_int32 for="exponent">0</cmo_int32>     <cmo_zz>1</cmo_zz>

</cmo_polynomial_in_one_variable>
</cmo_recursive_polynomial>
```

```
P=ox_launch(0,"ox_asir");
ox_rpc(P,"fctr",x^1000-y^1000);
ox_reset(P);
```



OoHG (OX RFC 104) の概要

OpenXM 100 形式の通信を HTTP GET (または POST) で wrap する方法を定義する. この wrapping により, 次のことを実現することを目標とする.

1. OpenXM 100 対応のサーバを容易に CGI サービス化する.
2. OpenXM のサービスを http を基本にグリッド化する.

OpenXM 100 ではサーバを立ち上げるには "login" することが一般に必要である. OpenXM-100 over HTTP GET (以下 OoHG と略記) は次のように login不要, login必要の二つの動作モードをもつ.

1. 不特定人が一度のみの計算利用をするような場合に login 不要で動作する.
2. OpenXM 100 と同様に login して session を維持することも可能である.

OoHG での通信は送信，受信で非対称である．GET を利用するためデータの形式は RFC 2396 で指定されているいわゆる URL エンコーディング方法を用いる．GET の利用の代わりに POST method を利用してもよい．送信データはキーワード ＝ 値　を & で区切ったものである．キーワードは英数字と _ をもちいるものとする．キーワードと値の組は入れ子構造となってもよい．

例1. `fctr(x^2-1);` を fctr.cgi へ送信する.

```
http://fctr.openxm.org/fctr.cgi?oxMessageBody=fctr(x%5E2-1)%3B
```

oxMessageBody 以外の全てのキーワードが省略されているので, anonymous, 1 回のみの実行, executeString & popString, 返答は text 形式となり `[[1,1],[x-1],[x+2]]` が戻る.

例2. 同じ問題を session を確立してから実行する. URL encoding 部分は " ... " で記述. RSA 暗号化されてる部分は ' ... 'で記述.

```
  --->
    http://fctr.openxm.org/fctr.cgi?loginRSA="takayama@hoge.org"
  <---
    OX100-OVER-HTTP-GET-controlChannelId: 4010
    OX100-OVER-HTTP-GET-dataChannelId: 4011
    OX100-OVER-HTTP-GET-challenge: '............'
  --->
    http://fctr.openxm.org/fctr.cgi?chanllenge-response='.........'
  <---
    OX100-OVER-HTTP-GET-nextChannelKey: 143245
    OX100-OVER-HTTP-GET-nextControlChannelKey: 534256
  --->
    http://fctr.openxm.org/fctr.cgi?dataChannelId=4011&
                                    channelKey=143245&
                                    oxMessageEncoding="lisplike_ox100"&
                                    responseEncoding="text"
          oxMessageBody="(OX_DATA,(CMO_STRING,\"fctr(x^2-1);\"))
                         (OX_COMMAND,(SM_executeString))
                         (OX_COMMAND,(SM_popString))"
<---
   Content-Type: text/plain
   OX100-OVER-HTTP-GET-nextChannelKey: 345137

   [[1,1],[x-1],[x+2]]
```

```
--->
   http://fctr.openxm.org/fctr.cgi?dataChannelId=4011&
                                  channelKey=345137&
                                  logout&
```

Web サービスの仕様記述 (自然言語およびキーワード)

まだあんまり考えてない.

URL=http://fctr.openxm.org は server=ox_asir, mathcap=... を用いて因数分解のサービスを行う. operator=fctr, argc=1.
利用可能なデータ形式は message_format=ox100lisplike および message_format="asir のユーザ言語" である.
利用には認証 (user=limited) が必要.

複雑な仕様は淘汰される. 例: SGML から XML へ

試験実装はありますか?

OpenXM/src/kan96xx/Doc/gfan.sm1, OpenXM/src/kan96xx/Doc/cgi.sm1
(グレブナ扇の計算)

polytope の計算 (dim, facet) に polymake を呼ぶ.

http://polymake.math.kobe-u.ac.jp/cgi-bin/cgi-polymake.sh

```
sm1>(gfan.sm1) run ;
Polymake is not installed in this system.
Using doPolymake.OoHG  ( OX-RFC-100 か OX-RFC-104 かは自動選択)
Using polymake.start.OoHG

cone.sample   と入力.
 略
Reduced GB is obtained:
 略
Calling polymake FACETS.
Trying web service.
Done.

Calling polymake DIM.
```

```
Trying web service.
ちなみに ctrl-C を押すと,
CTrace: extension<-loop<-readHTTP0<-cgiQueryPolymake
 <-ifelse<-doPolymake.OoHG<-doPolymake<-getConeInfo
 <-ifelse<-getStartingCone<-getGrobnerFan<-;

10.1.100.41 - - [12/Feb/2005:14:45:43 +0900] "POST http://polymake.math.kobe-u.ac.jp/cgi
```

まだやってないこと.

- Directory service (Brokerに相当) の実装. 設計のつめ.
- Web service の Security のガイドライン (自然言語で).
- WSDL 相当のことを自然言語と機械可読形式(キーワード)でやる方法のつめ.