CALIF

Computer Algebra Information Interchange Format

Andreas Strotmann

2nd OpenMath Workshop

Oxford, 18-19 July 1994

CALIF

Aims

Support distributed-memory message-passing parallel processing among cooperating processes in a heterogeneous Computer Algebra environment

- heterogeneous workstation hardware and OS
- heterogeneous CA systems
- heterogeneous software systems (CA, visualization, numerics)

Subgoals

- Transport Layer: communications protocol
- Data representation

Setting

Parallel processing: „normal“ and...

- co-operative parallel processing
  use algorithms for separate sub-tasks implemented in different systems [PoSSo]
- competitive and speculative parallel processing
  use several algorithms or parameters for same task, take first result [PAClib]

Distributed computing

- evaluation/simplification/computation services
- directory services
  - translation service (e.g. CALIF <-> system internal)
  - CA knowledge base service (names, rules, definitions...)
  - resource location service [cf. ToolTalk, Central Control]
- dynamic host & process control (e.g. PVM)

CALIF - current state

- data format in concepts stage; implementation to be done
- proof-of-concept implementation of parallel REDUCE/PVM available as testbed (master and evaluation server)
- OEM (i.e., C-callable) version of Maple available; testbed implementation planned (evaluation server only)
- ‘demo’-system configuration planned:

Transport Layer

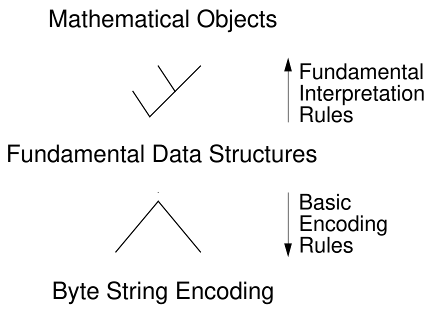
Base protocol for parallel processing is message passing:

- well understood
- PD software abounds (PVM etc.)
- standardization process underway (MPI released 31-03-94)
- ==> take what’s there

Advanced parallel processing paradigms to be built on top of message passing module:

- farms of evaluation servers
- blackboard shared memory w/ pattern matching (~Linda)
- need all protocol layers available for potentially complex communication patterns in parallel CA algorithms
- CA info servers for efficient math knowledge retrieval

CA Data Representation Layers



OpenMath Proposals I

- separate standardization documents for (at least) these:
  OpenMath ‚Basic Encoding Rules‘ (BER)
  OpenMath ‚Fundamental Data Structures and Interpretation Rules‘ (FDS)
  OpenMath ‚Mathematical Objects‘ (MO)
- OM FDS fixed for all time
- OM MO steadily growing (extensions)
- few OM Encoding Rules may be added, e.g. for compatibility
  BER always used unless different ER negotiated successfully
  BER mail-safe-ASCII based for human legibility

Precedent:

- ISO/CCITT 8824 (ASN.1) and 8825 (Basic Encoding Rules)
- MIME: ‚contents-type‘, ‚contents-encoding‘, ‚character-set‘

CALIF Byte String Encoding

- String, not stream [message passing model]
- ASN.1 BER based encoding (binary, tagged byte string)
  - integer, float, character (includes bytes), boolean
  - string of characters
  - sequence

  \+ small set of performance critical extensions expressed in ASN.1
  - label
  - pair (attribute/value)
  - several kinds of symbols (registered, user, anonymous)
- compromise between standards compatibility and compactness

CALIF Fundamental Data Type

An “expression” is an

- atomic expression, i.e.,
  - a base data object (small integer, IEEE float, boolean, bit/byte/ character string), or
  - an XDR-encoded vector of machine integers or floats, or
  - one of several kinds of symbols, or
  - a reference to a shared sub-structure;

or it is a

- compound expression, i.e.,
  - a sequence of expressions, or
  - a labelled subexpression, or
  - for efficiency, an attribute/value pair of expressions

Fundamental Interpretation

- atomic expressions represent „themselves“
- a sequence of expressions is interpreted as
  - either an operator followed by its operands
  - or a type specifier followed by its instance components
- an operator or type specifier (i.e. the first element of an expression sequence) is either
  - a registered or user symbol
  - a potentially labeled expression sequence representing a compound data type specifier (a symbol followed by a set of attribute/value pairs, each attribute being a name and each value, an expression)
  - a reference to either kind of shared sub-structure

Compound Data Type Specifiers

- Name of abstract mathematical domain (registered), e.g. `Polynomial`
- (optional) comments
- (optional) mathematical properties of domain
  - component domains (“signature”), e.g. „over the Integers“
  - description of domain (e.g., algorithm to convert to general expression `(+ (* (^ x 2) 3) 4))`
- (optional) technical properties of particular representation
  - name or description of representation used (may be implicit), e.g. „sparse distributed“, „dense recursive“
  - characteristics of representation used, e.g., „canonical“, „normal“, „sorted“ (with ordering criteria, e.g. variable ordering) so certain transformations may become unnecessary

CALIF Symbols

denote

- mathematical functions, operators, constants, types
- data types, representations
- programming constructs
- typographical entities (greek symbols...)
- other concepts

require

- standardization & reference to existing standards (ASCII, ISO8859-*, ISO10646 (Unicode), SGML) and systems (Axiom, Maple, REDUCE...)
- compact encoding if frequently used
- user extensibility

OpenMath Proposals II

Name spaces

- mathematical context (group theory, high energy physics)
- extensions
- reference to external standards (Unicode, SGML,...)
- reference to systems (Axiom, Maple, REDUCE; OpenMath; EuroMath; UNIX, Khoros, WWW, ...)
- user name space(s), anonymous symbols
- name space of name spaces

Tag selected name spaces efficiently in Basic Encoding (e.g. ‚&' for SGML) (optional)

Precedent: LISP dialects and probably others

OpenMath Proposals III

OpenMath symbol's registration info to consist of:

- associated name space id (e.g. `calculus`)
- symbol's reference id within name space (e.g., `20`)
- descriptive ASCII string, including symbol's category (function symbol, type specifier symbol, ...), e.g. `Euler constant`
- abbreviation for use in Basic Encoding (optional), e.g. `e`
- reference to related entities of external standards, e.g. SGML's code for *e* , and to those of existing CA systems, e.g. REDUCE's `e` and Maple's `E`
- mathematical definition of entity, e.g. $e = \sum_{i=0}^{\infty} \frac{1}{i!}$

Precedent: ISO 10646 / Unicode

Representing Mathematical Objects

- Multiple representations of „equal" mathematical objects, e.g.
  - polynomial as a general expression involving +, *, ^
  - polynomial in sparse distributed representation with specified monomial ordering scheme (e.g. when parallelizing Groebner)
  - polynomial in sparse recursive representation with specified variable ordering (e.g. for efficient data exchange among REDUCE systems)
  - typing information optional if deducible
- Sender chooses standard representation closest to its internal form for minimal overhead in homogeneous environment
- Recipient potentially has to do transformations
  - can call on translation server if unable to do so itself

Related Standards: Message Passing Interface

MPI (Message Passing Interface): interesting aspects

- What is standardized?
- What is not standardized?
- Rationales
- Hints for implementors, programmers, and users
- Fast standardization process (2 years)
    - subgroups reporting to plenary session
    - publicly available minutes and e-mail discussion archives
- Data type „handle“

see http://www.netlib.org/mpi/index.html or e-mail „send index for mpi“ to netlib@ornl.gov

Related Standards: Unicode / ISO10646

- Conformance rules
- Internationalization
- Conversion tables for existing related standards and systems
- Details:
    - standardizes code point & descriptive name, the latter including the coded entity‘s category (letter, syllable,...)
    - uses „combining“ characters to reduce code space
    - expanding

Model for OpenMath symbol registry?

Further Related Work

Distributed Memory Parallel CA

||Maple||, LindaMaple, ParSAC, DSC, RR,...
shared memory: PAClib, MuPAD, and many more

Distributed CA

CAS/pi, PoSSo (PoSSo/XDR, ASAP, Central Control), MP, ACELA
Interfaces CA<->other systems: many! (TeX, plotting, visualization, numerics...)

CA systems

REDUCE, Maple, Axiom, Macsyma, Mathematica, and all the others

Others

Euromath, SGML Math extensions (TEI, HTML+,...); MIME
ASN.1, XDR,...; FITS, netCDF, HDF...; CGM, PHIGS,...
SMTP, RPC, DTM, HTTP, NNTP,...; MPI, PVM, P4,...; X11,...
ToolTalk, CORBA, DME,...; Kerberos, PGP,...; ...