

Classification of communication and cooperation mechanisms for logical and symbolic computation systems

an article by : Jacques Calmet & Karsten Homann

presented by : Sidi Ould Biha

2 mars 2007

Logical & Symbolic Computation Systems

Logical CS

- Formalize & proof problem in some logical framework
- Use AI methods
- Theorem proof systems : Coq, Hol ...

Logical & Symbolic Computation Systems

Logical CS

- Formalize & proof problem in some logical framework
- Use AI methods
- Theorem proof systems : Coq, Hol ...

Symbolic CS

- Library for mathematical computation
- Use efficient algorithm
- Computer Algebra systems : GAP, Maple ...



Combining Logical & Symbolic CS

Logical & Symbolic CS are complementary



Combining Logical & Symbolic CS

Logical & Symbolic CS are complementary

Advantages

- Improve power
- Improve soundness

Resolve problem of higher complexity



Combining Logical & Symbolic CS

Logical & Symbolic CS are complementary

Advantages

- Improve power
- Improve soundness

Resolve problem of higher complexity

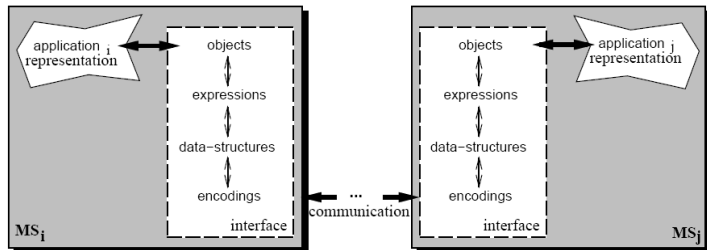
Problems

- Systems implemented as stand-alone
- Systems (large majority) do not provide interfacing



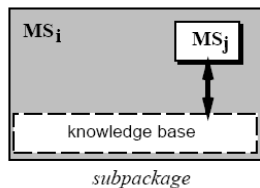
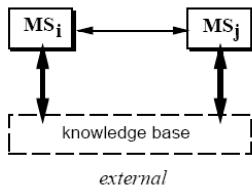
Communication mechanisms : Language & encoding

- Define how math info can be exchanged
- Open Math



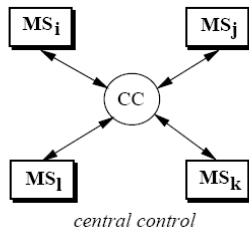
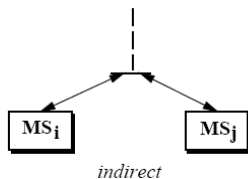
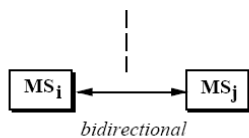
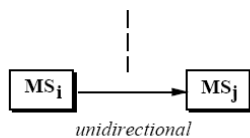
Communication mechanisms : Knowledge representation

- Communicating can be neither efficient nor practical : information may not be explicitly known
- Sharing the mathematical object between MS



Communication mechanisms : Flow direction

- 4 communications architectures



Cooperation mechanisms : Master/Slave

Mathematical problems typically solved by divide and conquer strategie



Cooperation mechanisms : Master/Slave

Mathematical problems typically solved by divide and conquer strategie

- Solving sub problems by suitable system

Cooperation mechanisms : Master/Slave

Mathematical problems typically solved by divide and conquer strategie

- **Solving sub problems by suitable system**
- **It is often sufficient to solve the problem in one MS with aid of other MS**

Cooperation mechanisms : Master/Slave

Mathematical problems typically solved by divide and conquer strategie

- **Solving sub problems by suitable system**
- **It is often sufficient to solve the problem in one MS with aid of other MS**
- **Implementing it is easy : large majority of interface are Master/Slave cooperation**

Cooperation mechanisms : Master/Slave

Mathematical problems typically solved by divide and conquer strategie

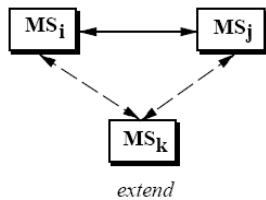
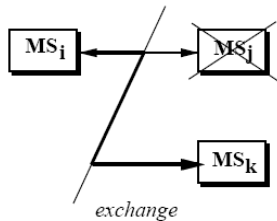
- **Solving sub problems by suitable system**
- **It is often sufficient to solve the problem in one MS with aid of other MS**
- **Implementing it is easy : large majority of interface are Master/Slave cooperation**

Oracles

- Black boxes (yes/no/result)
- Systems as tools in master/slave cooperation
- Not adequate in general : dont provide internal mathematical information

Cooperation mechanisms : Extensibility & Interchangeability

General interfaces must be system-independent



Cooperation mechanisms : Others

Consistency & Closure

- Consistency : guarantee that a MS **understand** transmitted terms
- Closure : ensure that the MS result does not contain **unknown symbols**



Cooperation mechanisms : Others

Consistency & Closure

- Consistency : guarantee that a MS **understand** transmitted terms
- Closure : ensure that the MS result does not contain **unknown symbols**

Trust

3 levels of trust between CAS and TPS

- No trust at all : use the result as an aid
- Partial trust : accept result temporary
- Complete trust : accept result as truth

An example : Coq & GAP

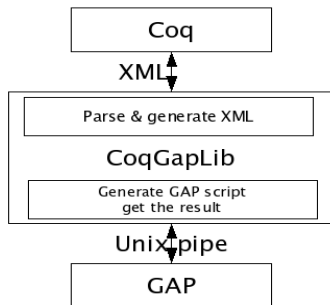
- **GAP** : CAS specialized in Group computation theory
 - ▶ Developed by international cooperation : Germany, US, UK
 - ...
 - ▶ Open Source and free
 - ▶ Kernel write in C, powerful script language
 - ▶ Implement wide variety of algorithms for Group Algebra(Matrix, Sylow subgroup ...)

An example : Coq & GAP

- **GAP** : CAS specialized in Group computation theory
 - ▶ Developed by international cooperation : Germany, US, UK
 - ...
 - ▶ Open Source and free
 - ▶ Kernel write in C, powerful script language
 - ▶ Implement wide variety of algorithms for Group Algebra(Matrix, Sylow subgroup ...)
- **Coq** : theorem provers based on CIC
 - ▶ Developed by Logical team (INRIA)
 - ▶ Open Source and free
 - ▶ ssreflect : Coq extension developed by G. Gonthier to proof the 4 color theorem (with B. Werner)

CoqGAPLib

- Interface between Coq & GAP, write in C
- Communication with XML (Coq) & Unix pipe (GAP)
- Use a new Coq tactic external to call external program



CoqGAPLib features

- Importation of finite group from GAP to Coq
- Representation of a group as a matrix (multiplication matrix)
- Last result : verification of a group of size 300
 - ▶ XML file size : 76 Mo
 - ▶ Time : 9 hours 45 minutes

THANKS :-)

