

# CoLA: A Coordination Language for Massive Parallelism\*

Béat Hirsbrunner

Marc Aguilar

Oliver Krone

*Institut d'Informatique, Université de Fribourg, Chemin du Musée 3, CH-1700 Fribourg, Suisse*  
Tél. +41/37/826550, FAX +41/37/826551, e-mail: {hirsb, aguilar, krone}@cfruni52.bitnet

## Introduction

The emerging generation of supercomputers will be suitable for massive parallel applications. To orchestrate the communication needed to coordinate concurrent tasks, we are currently developing a new coordination language [1] called CoLA. Specifically CoLA is aimed to provide a high programming abstraction level for massively parallel computing, especially for applications in the field of distributed artificial intelligence. CoLA is embedded in classical programming languages and combines both declarative as well as imperative semantic properties in its definition.

CoLA incorporates a high level identification concept, defining communication objects as correspondents [2], a postal mail delivery model for efficient and transparent message passing [3][4] and new communication and synchronization mechanisms.

## Programming Model

CoLA is based on the concept of correspondents, a high level identification abstraction for purpose of communication of processes. Correspondents are defined logically; sets of correspondents can be computed during runtime of an application using PROLOG-like declarative rules. The expressiveness of the rules can be utilized to define abstract communication topologies like rings, trees or hyper-cubes. CoLA offers several predefined communication packages which serve as a base for the construction of new user defined communication structures. The mechanism of defining communication structures using the concept of correspondents facilitates the implementation of complex information channels by completely decoupling the logical communication structure from the physical structure. The introduced abstraction has several advantages: e.g., the user does not have to care about migrating processes; the actual physical site of a process remains transparent and the resulting algorithms are highly portable and scalable since they do not depend on certain physical communication structures of the target machine.

In order to support dynamic migration and load balancing strategies, CoLA uses a postal mail delivery

model (called the *Post*) providing transparent and efficient process identification, mail management and routing facilities. The model uses a fully distributed address-book and a finite state machine to accomplish transparent communication. If a process sends a message to another process, it uses the appropriate correspondent; the Post will then resolve the logical address into physical site information and transmit the message. For efficient message passing we have developed and tested several communication protocols which can be used by the Post to optimize message paths according to various system parameters, such as the average migration frequency of processes and the average message size.

The communication and synchronization facilities are based on extended blocking and non-blocking send/receive primitives offering a programmer a flexible and application oriented programming model. Control operations such as filter and select operators which work on sets of correspondents allow for easy determination of communication partners. Interrupt driven communication primitives, functions to create and start processes and barrier synchronization facilities based on sets of correspondents, are all included.

## Results and Future Work

We have developed a specification of the syntax and semantics of CoLA and we are currently underway to implement a first prototype on top of ISIS [5] for a cluster of workstations. Future work will include an investigation of the practicability of this new programming approach for the implementation of classical algorithms in the field of distributed artificial intelligence.

## References

- [1] N. CARRIERO, D. GELERTER. *How to Write Parallel Programs, A First Course*, MIT Press, Cambridge 1992.
- [2] M. AGUILAR. "Communication et migration dynamique dans Pact, un environnement pour la programmation heuristique parallèle", Ph.D. Thesis, University of Fribourg, 1993.
- [3] M. AGUILAR, B. HIRSBRUNNER. "Post: A new postal delivery model for parallel heuristic programming", submitted for publication, *Working Conference on Programming Environments for Massively Parallel Distributed Systems, IFIP WG 10.3*, Ascona, Switzerland, April 25-30, 1994.
- [4] M. AGUILAR, B. HIRSBRUNNER. "Pact: An Environment for Parallel Heuristic Programming", *International Conference Massively Parallel Processing Application and Development*, Delft, The Netherlands, June 21-23, 1994.
- [5] K. BIRMAN. "The Process Group Approach to Reliable Distributed Computing", *Communications of the ACM*, Vol. 36, no. 12, pp. 37-53, 1993.

\*Grant No: 5003-034409 of the Swiss National Science Foundation 1993-1995