

## Mô hình hóa và tổng hợp hệ thống nhúng trong môi trường OO. Một nghiên cứu trường hợp cảm biến thông minh

Germán Fabregat & Germán León  
Đại học Jaume I  
Khuôn viên de Riu Sec  
12071 Castellón, TÂY BAN NHA  
fabregat, leon @ inf.uji.es

Bernard Pottier, Olivier Le Berre \* & Loic Lagadec  
Đại học de Bretagne Occidentale  
20 đại lộ Le Gorgeu  
Brest, 29287, PHÁP  
thợ gôm, llagadec@univ-brest.fr

(\*) bây giờ với Đơn vị quản lý mạng / quản lý mạng của Alcatel  
Tuyển de Nozay, 91460 Marcoussis, Pháp  
olivier.le-berre@nmu.alcatel.fr

### trình tượng

*Phát triển ứng dụng hiệu quả cho phần cứng phức tạp hệ thống đang trở thành một thách thức hàng ngày. Giảm thời gian để thị trường và tính đặc thù của phần cứng đang đặt ra vấn đề về việc vào lập trình viên. Trong bối cảnh này, tự động tạo mã dựa trên các mô hình hướng đối tượng của hệ thống mục tiêu tiết lộ như là một giải pháp hiệu quả cả ứng dụng chất lượng cao và để giảm nỗ lực và do đó cắt giảm thời gian phát triển. Chúng tôi trình bày một ví dụ điển hình cho việc này, trong đó hai kiến trúc khác nhau được nhắm mục tiêu bởi một Hệ thống phát triển Smalltalk-80. Chúng tôi trình bày những điều cơ bản của tạo mã được hỗ trợ bởi hệ thống và chính pects được xem xét khi thiết kế một sự phát triển như vậy nền tảng. Chúng tôi mô tả các nền tảng mục tiêu và đưa ra bài kiểm tra của các kết quả thu được.*

### 1. Giới thiệu

Công nghệ tích hợp cung cấp ngày càng nhiều nguồn và chức năng cho các thiết bị mới. Trong số đó, chúng ta có thể đề cập:

- tăng tốc độ và số lượng công,
- thêm tính linh hoạt: công nghệ reconfigurable,

Mặt khác, tích hợp lại phần mềm và hệ thống quirements cũng nhanh chóng tăng lên. Thiết bị đơn giản nhất sẽ cần hỗ trợ cho truyền thông và phân phối trol, phần mềm ứng dụng cấp cao và các công cụ phát triển, trao đổi dữ liệu chuẩn.

Để thêm áp lực cho các nhà thiết kế công nghiệp, thời gian để thị trường thiết kế bao gồm các tính năng mới này đang trở thành ngắn hơn và ngắn hơn

Trong bối cảnh này, có một nhu cầu cấp thiết cho các phương pháp trung cấp: phát triển phần cứng và phần mềm ngắn đường dẫn, thành phần dễ dàng của các mô hình tính toán, liên dữ liệu khả năng hoạt động.

Bài viết này sử dụng bàn làm việc cho cảm biến thông minh tầm nhìn để mô tả một phương pháp cho phép phát triển nhanh chóng- đề cập đến các ứng dụng cho các kiến trúc hệ thống khác nhau. Các toàn bộ bàn làm việc được phát triển bằng một đối tượng duy nhất sự phục hồi Các chức năng của cảm biến được đề cập tại mức tập lệnh. Để nhanh chóng phù hợp với các hệ thống khác nhau, chúng tôi đã sử dụng mô hình hướng đối tượng để tạo ra metacom-phi công bắt đầu từ một mô tả cấp cao của mục tiêu nền tảng để tạo mã có thể biên dịch phù hợp với nó. Environment đã được thử nghiệm trên hai nền tảng khác nhau:

- một hệ thống từ xa nhưng có micropro- riêng của mình giám sát, cảm biến thông minh và hỗ trợ truyền thông,
- một nền tảng phát triển trong phòng thí nghiệm nơi thông minh cảm biến có thể truy cập trên cổng song song (ISA) của PC.

– xử lý dữ liệu thông minh: cảm biến thông minh.

Kết quả là có một nhu cầu mới nổi để xác định và đồng bộ hóa kiến trúc quy mô cho các gia đình của ứng dụng nhúng mới-tions với yêu cầu khắt khe, pha trộn khác nhau mô hình đặt: xử lý song song, thu nhận phẳng, in-cảm biến và thiết bị truyền động, mạng. . .

Ở cấp độ lập trình thấp nhất, cảm biến xuất hiện như một mô hình đối tượng (kiến trúc) hiểu các thông điệp gần với tập lệnh, trong khi mã không cụ thể (mes-sages) được gửi đến bộ xử lý máy chủ. Tại một trung gian cấp độ, chúng tôi xây dựng các đối tượng cấp dữ liệu (dữ liệu) phù hợp cho điều khiển từ xa hoặc gỡ lỗi. Những đối tượng này có thể được sử dụng để

## Trang 2

phát triển và thử nghiệm các chương trình ứng dụng cấp cao. Một objec-Tive là để tránh bị mất ngữ nghĩa do một phần mềm lớp kiến trúc. Một mục tiêu khác là sử dụng quy trình hướng đối tượng ngữ pháp và công nghệ theo chiều dọc, xuống cứng đặc điểm kỹ thuật kho, để có được sự phát triển nhanh chóng và an toàn phương pháp.

Bài viết trình bày kiến trúc mục tiêu và nền tảng. Nó cung cấp một cuộc thảo luận ngắn về nhân vật Smalltalk-80-istics và mối quan hệ của họ với gỗ. Mô hình lập trình viên cho máy ảnh từ xa là một thành phần của phần cứng và các thành phần hệ thống như: một mạch cụ thể, tuần tự chủ đề, hoạt động truyền thông, điều khiển FPGA. Giấy mô tả kiến trúc của trình biên dịch chung và một số điểm kỹ thuật về dịch thuật. Sau đó, chúng tôi trình bày im-mạ trên nền tảng phát triển. Phần cuối cùng của bài báo trình bày các hướng nghiên cứu hiện tại bao gồm tổng hợp logic, mô hình kiến trúc có thể cấu hình lại và hỗ trợ khả năng tương tác. Chúng tôi kết luận với thực tế là-khien và kết quả.

## 2. Cảm biến thông minh MAPP2200

Các mảng ma trận Hình Processor MAPP2200 thông minh cảm biến [1] kết hợp trên một con chip duy nhất một mảng cảm biến 256 x 256 photodiode và một số đơn vị xử lý SIMD để phù hợp với các hệ thống điều khiển dựa trên cảm biến gần mô hình xử lý hình ảnh [2]. Mặc dù nó có thể khử mã và thực hiện các hướng dẫn riêng của nó, nó không bao gồm một trình tự cũng không phải là một đơn vị kiểm soát dòng chương trình, do đó cần hoạt động của logic bên ngoài cho các nhiệm vụ này. Mô hình lập trình được cung cấp bởi các hệ thống sử dụng MAPP luôn bao gồm lỗi thực thi SIMD song song và một điều khiển tuần tự. Mô hình này thường được mở rộng với tiếp tục xử lý tuần tự hoặc song song. Hệ thống dựa trên MAPP tems là một ví dụ hoàn hảo để nhận ra những khó khăn của pro-ngữ pháp cho từng kiến trúc đích khác nhau và để hiển thị làm thế nào một cách tiếp cận từ đặc điểm kỹ thuật cấp cao cộng với thấp mô hình kiến trúc cấp là giải pháp tốt nhất cho chi phí sản xuất hiệu quả mã mục tiêu.

## 3. Smalltalk-80, kịch bản đối tượng và gỗ

½ Ä Ò Ù Ò Ö Ø Ö × Ø ×

Smalltalk-80 đã được thiết kế để giải quyết sự phức tạp-ity của môi trường phát triển cho các máy trạm trong tương lai tại Xe-PARC Nói ngắn gọn, ngôn ngữ có cú pháp đơn giản, không có cấu trúc điều khiển.

Kiểm soát tuần tự hoặc đồng thời được mô tả trong các lớp ai có thể trả lời các tin nhắn cụ thể:

*Boolean* thực hiện thực thi có điều kiện để đáp ứng với *ifTrue*: hoặc *ifFalse*: , *Integer* thực hiện *lầnRepeat*: , *Col-lection* có thể sắp xếp một thực thi được chỉ định bởi *do*: , *thu thập*: ...

Các khối cơ chế đang cơ cấu không bao hàm một thực thi ngay lập tức tại định nghĩa khối . Họ đang đến monly truyền dưới dạng tham số trong tin nhắn có người nhận cide nếu việc thực hiện sẽ diễn ra hay không. Khối là các đối tượng kèm theo một chuỗi các hướng dẫn, cộng với một tham chiếu môi trường lỗi kéo.

Tin nhắn được hiểu bởi các trường hợp từ một gia đình của các lớp học. Họ ngầm ám chỉ một chức năng hoặc một hành động có cùng ngữ nghĩa về gia đình này. Ví dụ, ev-Bộ sưu tập ery biết cách lặp lại các yếu tố trong phản ứng để tin nhắn *làm*: , mọi số đều biết cách tính *bình phương* của nó để đáp lại thông điệp *bình phương* , mọi đối tượng tạo ra một văn bản để đáp ứng với thông báo *printString* . Chuyên nghiệp ngữ pháp biết ngữ nghĩa đằng sau các thông điệp. Họ có thể sử dụng lại tên tin nhắn cho các lớp mới có thể chia sẻ một số hành vi với những người hiện có. Họ cũng có thể lấy lại chức năng sử dụng kiến thức của họ về hệ thống.

Một điểm quan trọng khác là mô đun hạt mịn trong hệ thống. Các lớp học được nhóm trong các thể loại spond để kernels hạt có tổ chức dữ liệu cụ thể và chức năng. Các giao diện tin nhắn cho các hạt nhân này sử dụng lại phần mềm hiện có một cách có hệ thống và an toàn. Như một ví dụ *Bộ sưu tập* tạo thành một hỗ trợ chung đó là tái sử dụng ở mọi nơi, điều đó đã được chứng minh và hiệu quả gorithms, có một liên kết rõ ràng, đầy đủ và trực giao đối mặt. Cấu trúc dữ liệu Smalltalk-80 hoạt động an toàn để tồn tại các lớp ing, và sẽ làm việc cho các lớp học trong tương lai.

¾4İİÖ ×

Smalltalk-80 đã được chọn là môi trường tối ưu-cổ gắng phát triển phương pháp đề xuất cho hệ thống mô hình hóa và lập trình. Nhân vật quan trọng nhất-istic, duy nhất cho hầu hết các ngôn ngữ OO phổ biến (Java, C ++ ...) là không có gõ, dễ dàng cho phép xử lý cao thông số kỹ thuật cấp được phân lập từ các chi tiết mục tiêu cụ thể. Các khía cạnh quan trọng khác là sự bao gồm của C và Smalltalk (hoặc kiểu yacc) trình phân tích cú pháp trong hệ thống cơ bản, giúp ích rất nhiều khả năng mở rộng và chia sẻ bộ chọn tin nhắn. phân tích mã cấp cao và tạo mục tiêu phần mềm.

Smalltalk-80 chỉ dựa vào các lớp để quyết định làm thế nào một message nhân sẽ được thực thi. Trong thời gian chạy, hệ thống tìm kiếm lớp của một người nhận, tìm kiếm trong cây thừa kế cho bộ chọn thông báo, tìm nạp và thực thi một phương thức (mã cho một thông điệp được thực hiện trong một lớp cụ thể). Liên kết động-ing cho phép sử dụng một thuật toán hoặc tổ chức dữ liệu hiện có trên các lớp mới phát triển. Tuy nhiên, nó ngụ ý quá mức đầu vào thời gian chạy để tìm kiếm phương pháp. Đây là chi phí cho

Thay vì sử dụng một ngôn ngữ ép buộc trong direction công với một phương pháp quá mơ hồ, sự lựa chọn là

2

## Trang 3

sử dụng một ngôn ngữ vượt trội trong mô hình hóa và để làm việc trên các ràng buộc bổ sung cho phép tạo mã và sau này, mô tả phần cứng. Xem [3, 4, 5] cho các công việc liên quan.

### 4. Biên dịch thành tập lệnh C và MAPP

Mã Smalltalk hoàn toàn tuân thủ mặc dù một số thông điệp trên các bộ sưu tập có ngữ nghĩa song song. Smalltalk phương thức là chuỗi các tin nhắn được gửi đến người nhận. Các lập trình viên tập trung sự chú ý của mình vào những máy thu này và biết chính xác lớp học hoặc hành vi của họ.

Hệ thống có thể được xem như một tập hợp các đối tượng có par-chức năng ticular: xử lý tuần tự, synization, truyền thông, xử lý cụ thể. . . Khác biệt chính lên men với các tính toán Smalltalk bình thường là addi-các ràng buộc liên kết (gõ) cho phép dịch sang micropro-mã giám sát, gọi hệ điều hành, hướng dẫn, phần cứng bộ điều khiển và nhiều hơn nữa. . .

Tổng hợp hệ thống tách các mô hình với nhau, en-tạo mã abling cho mỗi thành phần. Nó tạo ra mô tả chấm dứt cho dữ liệu trao đổi giữa các thực thể cấp thấp, cũng như các giao diện (mô tả đối tượng và thủ tục trao đổi) giữa cấp thấp và ap-công cụ số nhiều hoặc phát triển.

Cốt lõi của quá trình dịch thuật là Smalltalk sang C lập bản đồ.

°½ Ë Ù Ò Ø Đ Ñ Ó Đ

#### 4.1.1 Ảnh xạ kiểu

Ngôn ngữ C đã được thiết kế để lập trình hệ thống hiệu quả trên các bộ xử lý của thập niên 70, và vẫn như mô hình chuẩn để thực hiện chương trình tuần tự. Chúng tôi-ing C là ngôn ngữ trung gian cung cấp các tiện ích cơ bản cho quản lý tài nguyên bộ nhớ và vi xử lý trong một thiết bị cầm tay

#### 4.1.2 Cú pháp khai báo kiểu

Khi tạo mã C từ Smalltalk, các biến riêng tư xuất hiện trong 3 tình huống: biến thể hiện, thời gian và tham số phương thức. Phiên bản hiện tại của trình biên dịch sử dụng nhận xét lớp và chú thích phương thức chuẩn cho khai báo kiểu. Nhận xét lớp không có cú pháp chính thức. Tuy nhiên, có một sự đồng thuận về việc tuyên bố trước lớp ferred cho mỗi biến thể hiện được khai báo trong lớp.

Đối với mỗi lớp được tham chiếu, người dịch thực hiện một hướng lên truyền tải của cây thừa kế và tích lũy dữ liệu thánh thư. Sau đó, những mô tả này sẽ được viết lại thành C khai báo cấu trúc.

#### 4.1.3 Phân tích mã

Các công cụ bổ sung cho phép cấu trúc lớp de- Smalltalk đoạn mã để tạo thư viện, tệp khai báo C hoặc exe-chương trình dễ cắt. Cốt lõi của các công cụ này là tiêu chuẩn Trình biên dịch Smalltalk được mở rộng để hỗ trợ mã đa mục tiêu thể hệ.

Máy quét tạo ra một cây chương trình bằng cách sử dụng các lớp con của *ProgramNode* cho các phương thức, biến, câu lệnh, as-ký kết. . . Trình biên dịch chuẩn cung cấp hỗ trợ để đăng ký lại xử lý một cách khó hiểu cây này bằng cách phân lớp lớp *ProgramNodeEnumerator*. Một cây mới được xây dựng chứa C các nút tương đương với các nút Smalltalk cho các bài tập, khối, nghĩa đen, tham số, biến, trả về và trình tự của các câu lệnh. Các phương thức được ánh xạ tới hàm C definitions. Tin nhắn được chuyển thành một trung gian Nút *CMessageNode* để xử lý thêm.

Các nút thông báo được dịch sang nhiều hoạt động C khác nhau hành, lặp, kiểm tra, gọi hàm, cũng như cụ thể xử lý mô hình.

Phương thức ví dụ dưới đây xuất phát từ lớp MappExample. Đây là phương pháp cho một vòng lặp bỏ phiếu bị troll bởi tham số thời gian:

**chờ đợi: thời gian**

đường.

Đề tạo mã C ra khỏi các phương thức Smalltalk, đó là cần thiết để cung cấp ánh xạ giữa một số lớp và C các loại đơn giản. Có tồn tại một ánh xạ như vậy trong CORBA IDL đặc điểm kỹ thuật [6], nhưng chúng tôi chủ yếu quan tâm đến việc xây dựng một bắt đầu với tên lớp Các thông số luôn được thông qua trình tạo hệ thống từ trên xuống với sự hỗ trợ phản cứng có thể cơ chế CORBA quá phức tạp.

C loại đơn giản xuất hiện dưới dạng bản dịch của các lớp (hoặc các lớp con) Integer, LargeInteger, Boolean, Float, Double, Chuỗi, Ký tự. . .

Mảng C tương ứng với các lớp có một kiểu duy nhất cho các phần tử (IntegerArray, FloatArray ...). Cấu trúc cor- trả lời tổng hợp các biến thể hiện đọc theo một cây thừa kế.

*tài nguyên: # (# (Số nguyên i thời gian))*  
*time timeRepeat: [0 đến: 7 do: [: i phạm không làm gì cả]]*

Phương thức này được chuyển thành hàm có tên địa chỉ và người nhận được thông qua như là lĩnh vực đặc biệt *tự* :

```
void MappExample_waitLoop (tự, thời gian)
MappExample * tự;
int (* thời gian);
{int i;
  int i3;
  cho (i3 = 1; i3 <= (* thời gian); i3 = i3 + 1)
    {cho (i = 0; i <= 7; i = i + 1) {}}
```

Trang 4

¾ Ế Ồ Ồ Ồ Ồ Ồ Ồ Ồ Ồ

Các biến là các thể hiện của lớp MAPP hoặc phụ các lớp được công nhận bởi trình biên dịch. Khi có tin nhắn được giải quyết cho các biến này, một trình dịch được xây dựng giữ môi trường nút. Có sự chuyển đổi đặc biệt các lớp tor cho mỗi nhóm tập lệnh MAPP. Một chức năng tạo mã được xây dựng trong các lớp này trình biên dịch có thể sử dụng để có được một đại diện C văn bản thích tin nhắn Smalltalk.

Giao diện lập trình MAPP là một tập hợp các thông báo tương ứng với các hướng dẫn cơ bản với các biến thể và pa- lan man. Toàn bộ tập lệnh đã được phát triển trong một thời gian rất ngắn do sự đều đặn tồn tại giữa mô hình phần mềm và bộ giải mã hướng dẫn thực tế.

Tạo mã C là sự lựa chọn cho máy ảnh từ xa. Chương trình sẽ truy cập vào MAPP bằng cách sử dụng một FPGA có thể khóa bus hệ thống để có được đồng bộ hóa với nó liên bang. Các đại diện của opera điều khiển FPGA- Tions được xử lý bằng cách sử dụng con trỏ C tại các địa chỉ không đổi, cộng với chức năng cho phép thiết lập lại hoặc tái cấu hình.

Ví dụ sau đây là một phần của phương thức từ lớp MAPP gửi tin nhắn di chuyển đến cảm biến. Các triệu chứng bols #ad và #r biểu thị kết quả chuyển đổi và công việc- đăng ký tập tin. Cuộc gọi thực hiện 8 lần chuyển liên tiếp được lập chỉ mục bởi biến y :

**sobelTransform**

*tài nguyên: # (# (Số nguyên #y))*  
*0 đến: 7 làm: [: y tự*  
*cửa hàng: #ad số: y trong: #r số: y]. Vv vv.*

Bản dịch C được đưa ra dưới đây. FPGA biểu thị một con trỏ trên một cổng đồ họa ánh xạ cổng lệnh MAPP vào không gian địa chỉ của bộ vi xử lý:

Các công cụ DLLC cho phép sản xuất tự động các giao diện cho các thư viện động ANSI C.

Trình biên dịch sửa đổi phát hiện các lớp cụ thể này và tạo mã C thích hợp để gọi các hàm này.

Điều này được sử dụng để cấu hình lại, truyền thông thư viện, v.v ... Khung nhìn lập trình viên cho các lớp này là một tập hợp các chức năng được xác định tốt cần một sáng tạo tin nhắn, cộng với một bộ tin nhắn để có được dịch vụ. Các văn bản sau đây cho thấy quá trình gắn vào máy ảnh kernel cho dịch vụ chuyển tin nhắn, sau đó kết nối lại với FPGA hoạt động khẩu phần và thiết lập lại:

...

*fpga: = Fpga mới. Lớp điều khiển FPGA*  
*msgPassing: = CommunicationService mới.*  
*hình ảnh: = Đồ họa mới.*  
*msgPassing kinit. Tập tin đính kèm vào kernel*  
*fpga ConfigureFpga: 'someFile.conf'.*  
*fpga ResetFpga. Mùi ... Sản xuất hình ảnh ...*  
*msgPassing sendImage: 256 với: 256 với: bit hình ảnh*

Để dễ dàng vận chuyển từ hệ thống nhúng đến công cụ kiểm soát hoặc phát triển bên ngoài, mỗi lớp có một công cụ riêng Các chức năng tuần tự hóa mô tả đối tượng trên một luồng, hoặc xây dựng lại một đối tượng từ một luồng. Chức năng tương tự- Tionality tồn tại trên nền tảng khách hàng cấp cao.

**5. Biên dịch lại cho máy ảnh MAPP2200 PC**

Tính tổng quát của phương pháp đề xuất đã được chứng minh bằng cách nhắm mục tiêu một nền tảng thứ hai, đơn giản hơn nhiều. Giống nhau Mã Smalltalk cấp cao hiện được biên dịch thành một mã khác lớp ent để tự động sản xuất mã mục tiêu làm việc.

Lớp mới chỉ bao gồm các hướng dẫn được mã hóa gửi đến MAPP bởi máy ảo Smalltalk-80, xác định porta- khả năng ở mức chức năng (bộ lọc Sobel).

```
void MappExample_sobelTrans (tự)
MappExample * tự;
{
int y; /*...*/
cho (y = 0; y <= 7; y = y + 1) {
* FPGA = (0x800 + ((0x0 + y) << 7) + (0x0 + y)) << 10;
} / * v.v. / /
```

$$\frac{1}{2} \frac{1}{2} \ddot{E} \dot{Y} \times \ddot{N} \times \ddot{O} \ddot{O} \ddot{O}$$

Lập trình hệ thống có thể sử dụng phần mềm hiện có hoặc cứng kho tài nguyên được định nghĩa là thư viện C hoặc dữ liệu. Là hệ thống tổng hợp độc quyền từ đặc tả đối tượng, nó là cần thiết để tích hợp các tài nguyên này như là các lớp. Các Môi trường Smalltalk cho phép phân tích các ngôn ngữ bên ngoài để đại diện cho các chương trình bên ngoài như là các ví dụ của tích lũy hạng hạng. Sự hỗ trợ cho C hoàn thiện hơn kể từ khi

$$\frac{1}{2} \frac{1}{2} \ddot{E} \dot{Y} \times \ddot{N} \times \ddot{O} \ddot{O} \ddot{O}$$

Hệ thống thứ hai này là Hệ thống PC MAPP2200 [1], trong- làm mờ camera MAPP giao tiếp với PC bằng cách sử dụng một bảng mạch rất đơn giản. Bảng này cung cấp ba cổng PC để giao tiếp với MAPP: một để gửi hướng dẫn và đọc trạng thái, khác để đọc và ghi dữ liệu vào cổng nối tiếp, và thứ ba cho các nhiệm vụ điều khiển. Hệ thống đã được thử nghiệm trong Windows95, sử dụng Môi trường VisualWorks Smalltalk, từ ObjectShare. Giao tiếp với bảng ISA đã được thực hiện thông qua thư viện TVicHW32, cung cấp giao diện C đến một DLL để truy cập tài nguyên vật lý của PC. Nó là giá trị lưu ý rằng giao diện giữa môi trường Smalltalk và ngôn ngữ C được thiết kế tốt đến mức không có mã C nào có đã được phát triển và các chương trình Smalltalk được truy cập trực tiếp giao diện DLL.

Trang 5

$$\ddot{O} \ddot{O} \ddot{O} \ddot{O} \ddot{O}$$

Mã đã được tạo tự động bằng cách biên dịch Các phương thức Smalltalk trong lớp MappPC. Bây giờ là máy tính cây lation được tạo ra và phân tích, nhưng chỉ có phương pháp lá truy cập phần cứng MAPP (hướng dẫn MAPP) có đã được dịch sang các cuộc gọi DLL để truy cập vào ISA cần thiết cổng. Ví dụ tương tự đến từ sobelTrans- hình thức được trình bày dưới đây. Bây giờ mã Smalltalk vẫn còn đã thay đổi và chỉ lưu trữ tin nhắn : number: in: number: đã được thay đổi cho thông báo writeMappInXD: truy cập DLL và ghi vào cổng đầy đủ. Như mã hóa lệnh phụ thuộc vào các thanh ghi được sử dụng, trong Hệ thống Smalltalk việc dịch được thực hiện một cách nhanh chóng mã tin nhắn:

```
sobelTransform
0 đến: 7 làm: [: y tự
writeMapp Hướng dẫn:
((MappStoreTranslator mới)
src: #ad y; mệnh: #register y) mã]
...
```

6. Tình trạng và quan điểm

Công việc đang thực hiện liên quan đến phương pháp được mô tả bao gồm tổng hợp logic cho các đối tượng, mô hình hóa và các công cụ cho các kiến trúc, phương pháp và công cụ có thể cấu hình lại được xử lý dữ liệu ngày và hơn nữa, dịch chương trình. Những công trình này có thể được xem là nghiên cứu về ứng dụng-

Bộ sưu tập các giá trị đầu vào khối là triệt để sweep để tạo ra một bảng tra cứu vẫn chứa ob- jects. Sau đó, bảng này có thể được gấp lại thành PLA bằng cách sử dụng bi- người dịch nary và gói SIS [7] được sử dụng để sản xuất một bản đồ tương đương được thu nhỏ cho một công nghệ (xem [8] để biết thêm chi tiết). Các yếu tố kết hợp của mạch đến từ các khối không có tác dụng phụ, trong khi các mạch tuần tự được tạo ra bằng cách truy xuất các trạng thái khác nhau của các biến viết bởi các khối . Trong trường hợp của MAPP, tổng hợp logic có thể được gọi từ trình biên dịch để sản xuất một phần cứng tự động. Al- mặc dù vẫn còn nhiều việc phải làm để trưởng thành các công cụ, vòng lặp của sobelTransform đã được chuyển đổi để dàng xếp vào một máy tự động tạo ra các hướng dẫn cho BẢN ĐỒ. Nguyên tắc là phân bổ một biến sẽ giữ lại các giá trị khác nhau trong khoảng 0 ... 7 trong khi MAPP dịch mã được gọi để tạo ra một đầu ra.

```
bước: = 0.
[yy: = bước.
bước: = bước + 1 ÒÒ 8.
((MappStoreTranslator mới)
src: #ad y; mệnh: #register y) mã]
```

Trình tự phức tạp hơn là khả thi bằng cách xây dựng một bộ đếm chương trình ẩn hoặc thành phần bộ điều khiển [9]. Một điểm thú vị là việc phân bổ lại hiệu quả của FPGA nguồn trong một chương trình đầy đủ, và nghiên cứu về hỗ trợ hệ thống để hoán đổi hoặc thành phần của bộ điều khiển.

$$\frac{3}{4} \times \ddot{O} \ddot{O} \ddot{O} \ddot{U} \ddot{U} \ddot{O} \ddot{O} \ddot{O} \ddot{O} \ddot{O} \ddot{O} \ddot{N}$$

mục tiêu chia sẻ đường dẫn hệ thống như: linh hoạt, tốc độ để phát triển, khả năng tương tác dựa trên đối tượng Technol- yêu tính Ý tưởng chung đằng sau là tránh giao diện mù trong các công cụ phát triển và để xây dựng các mô hình cung cấp con- trol trên khắp con đường.

$$\frac{1}{2} \frac{1}{2} \text{ } \ddot{A} \text{ } \ddot{O} \times \text{ } \ddot{Y} \text{ } \ddot{O} \text{ } \times \text{ } \times$$

Tổng hợp logic đã được chứng minh là làm việc hiệu quả cho Khối Smalltalk-80 có số lượng đầu vào / đầu ra nhỏ tín hiệu. Ý tưởng là sử dụng các khối làm exe cấp thấp hơn hạt ngũ cốc như là một thay thế cho các thư viện công nghệ. Điều này ý tưởng giả định sự tồn tại của các công cụ khác cho kiến trúc sư- tổng hợp Ture và lên kế hoạch sản.

Đối với tổng hợp C, một đặc tả kiểu cho khối các biến và tham số là bắt buộc. Các loại cho logic tổng hợp là tập hợp các giá trị ( Khoảng thời gian Smalltalk-80 , Đặt . . . bao gồm các đối tượng). Mỗi đối tượng xuất hiện như một giá trị phải có khả năng thể hiện chính nó như là một chuỗi bit. Vì thế, không cần hạn chế biểu diễn nhị phân hoặc sử dụng fixe bộ các loại dữ liệu đơn giản.

Để đối phó với sự tổng hợp cho các GPU, một cấu hình trừu tượng- nền tảng urable (ARP) đang được thiết kế. Mục đích của máy này hình thức là để bao gồm một phần quan trọng của các chức năng và công cụ của các mạch có thể cấu hình lại, để cho phép tính di động của chương trình hành vi. ARP cho phép xác định cụ thể Kiến trúc có thể cấu hình lại (CRA) từ các mẫu của các ô, cơ sở định tuyến, mạch cụ thể. Nó cũng cho phép sản xuất bộ định tuyến, trình soạn thảo và các hoạt động điều khiển khác nhau cho CRA. ARP là một mô hình đối tượng phân cấp nhúng các thành phần với các giao diện phổ biến. Tập hợp các thành phần được mở rộng- chảy máu. Cấu trúc liên kết mô hình và các thành phần có thể là char- acterized bằng cách sử dụng một bộ công cụ đầy đủ.

Trong giai đoạn này của công việc, có một CRA được thực hiện cho kiến trúc Xc6200. Các chức năng có sẵn là chỉnh sửa, thao tác hình học (gương, cắt, dán, sao chép- tions), định tuyến điểm tới điểm, cấu hình một phần và đọc trở lại. Định tuyến toàn cầu vẫn đang được điều tra với một cơ bản thuật toán tìm đường được thực hiện trong ARP.

Ý tưởng cơ bản đằng sau thiết kế ARP là cung cấp một hoàn thành khung để tổng hợp và hiệu suất hệ thống

## Trang 6

đánh giá cho các gia đình của các ứng dụng và gia đình của ar- kiến trúc. ARP có lập CRA từ cấp cao công cụ tổng hợp, nhưng do môi trường đối tượng, nó rất dễ dàng để có được sự tương tác giữa các mô tả cấp cao và phần cứng. Hình 1 cho thấy khung nhìn của trình soạn thảo với một bộ điều khiển nhỏ cho MAPP. Bộ điều khiển này có thể làm việc trên bảng PCI Hot2 với điều khiển đầu vào / đầu ra từ Môi trường Smalltalk-80 (VisualWorks trên Linux).

một camera thông minh từ xa tương tác với mạng, hav- ing một vi điều khiển thực thi phần mềm ứng dụng. Bàn làm việc tổng hợp và tương đương với commer- hỗ trợ phần mềm cial cho MAPP đã được phát triển trong 4 tháng nam. Điều này bao gồm tạo mã, hình ảnh cơ bản lọc, số học song song, vận chuyển đến và đi từ máy quay thời đại, mức độ ứng dụng tương tác với máy ảnh. Hải cảng- ing đến nền tảng thứ hai nơi thông dịch viên vẫn ở phí kiểm soát đã đạt được trong vòng chưa đầy ba tuần. Các Trình tạo mã MAPP đã được thực hiện trong ba ngày.

Một lập trình viên phụ trách phát triển cơ bản có một mô hình rõ ràng của cảm biến có thể sử dụng trực tiếp trong một nguồn điện môi trường phần mềm đối tượng. Cả hai nền tảng được trình bày đã được thử nghiệm rộng rãi bởi chương trình chưa có kinh nghiệm- mers, đã có thể phát triển các ứng dụng hoàn chỉnh.

Để hoàn thành môi trường phát triển, hai chính mục tiêu vẫn còn. Đầu tiên, lập kế hoạch hướng dẫn hiệu quả để ex- chông chéo nhiệm vụ cấp thấp có thể được thực hiện bằng cách sử dụng cây chương trình C / MAPP. Điều này rất quan trọng cho hiệu suất kể từ khi tính toán và I / O có thể diễn ra cùng một lúc trong khi photodiode đang được tích điện.

Thứ hai, thể hệ phần cứng bổ sung cũng rất quan trọng kể từ khi bộ xử lý điều khiển dành thời gian vô ích để lập chờ một số điều kiện trong cảm biến. Sản xuất kiểm soát như vậy bên trong một đồ họa cũng rất hấp dẫn để tăng khả năng pháp sư. Các phương tiện lập trình tương tự có thể là ex- ploited để thực hiện các bộ lọc hoặc biến đổi thông thường khác tại đầu ra đăng ký thay đổi.

**Hình 1. Trình soạn thảo Xc6200 với Mapp con-xe đây**

°; İÖ Ò × Đ Ø ÓÒ ÓÓĐ ×

Công việc này dựa trên công nghệ BUỐC trích dẫn. BẬC THANG khuyến nghị sử dụng ngôn ngữ Express cho dữ liệu và mô hình ràng buộc. Một phương thức và một môi trường có đã được phát triển để cho phép mô tả chính thức về transla-sản xuất và bán tự động các công cụ liên quan (xem [10]). Mặc dù phương pháp này chỉ được áp dụng cho vấn đề thực tế trong công nghệ phần mềm, nó được mong đợi rằng nó sẽ hỗ trợ sản xuất các giao diện dữ liệu khác nhau cho khả năng tương tác và hơn nữa, chuyển đổi chương trình cho các mục tiêu khác nhau. Phiên bản mới của môi trường này là được thiết kế trong Smalltalk-80 và sẽ dễ dàng trộn với các hệ thống-công cụ tổng hợp tem.

° ÓÒ ĐỪ × ÓÒ

Hai nền tảng cảm biến thông minh đã được giải quyết từ các biến thể nhỏ của cùng một môi trường. Một nền tảng là

Phát triển hơn nữa cũng nên tập trung vào mối quan hệ-vận chuyển giữa các thành phần hệ thống: trans- sync fers, giao tiếp đệm, giao tiếp không đồng bộ-Tions và các chi tiết phần cứng khác nên được quản lý bởi các trình biên dịch. Bối cảnh chung là automa hiệu quả-Tion tổng hợp hệ thống, sử dụng một máy chủ có thể cấu hình lại được hình thức với các mô hình và công cụ tương thích. Những nỗ lực đang diễn ra đang giải quyết tổng hợp phần cứng từ các hành vi của Smalltalk-thông số kỹ thuật ioral.

Sức mạnh khác nhau của các loại được yêu cầu làm đầu vào cho loại trình biên dịch này: yêu cầu tổng hợp logic hiệu quả nhiều thông tin về đầu vào đơn vị chức năng hơn mã hóa cho vi xử lý. Áp dụng các ràng buộc kiểu cho Smalltalk các phương thức có phần giống như một đánh giá một phần trong đó có thể đơn giản hóa biểu thức chương trình trong Rela-tion để quy tắc tổng hợp cấp thấp. Tiến bộ về những điểm này ngụ ý các cơ chế mới để quản lý các loại ngôn ngữ cấp độ.

**Tài liệu tham khảo**

[1] *Hướng dẫn sử dụng hệ thống PC IVP, MAPP2200* . Liên kết, Thụy Điển: IVP, Sản phẩm Tầm nhìn Tích hợp AB, 1994.

**Hình 2. Hệ thống MAPP từ xa hiển thị một hình ảnh được chụp**

[2] J.-E. Eklund, C. Svensson và A. Astrom, VLSI im-mạ của bộ xử lý hình ảnh mặt phẳng tiêu cự - một thực tế

ization của khái niệm xử lý hình ảnh cảm biến gần  
*Giáo dịch của IEEE trên Hệ thống VLSI*, tập. 4, trang 322  
 335, tháng 9 năm 1996.

- [3] MB Ballard và A. Wirfs-Brock, Nhanh QuickTalk: a  
 Phương ngữ Smalltalk-80 để xác định các phương thức nguyên thủy,  
*OOPSLA'86 Proc.* , tập 21 tháng 11 năm 1986.
- [4] TD, ứng dụng Ubiquitous: hệ thống nhúng  
 đến máy tính lớn, *Truyền thông của ACM* , tập. 38,  
 Tháng 10 năm 1995.
- [5] ID, KT, MJ, WS và KA, đã trở lại  
 Tương lai. Câu chuyện về SQUEAK, một Smalltalk thực tế  
 được viết bằng chính nó, Tiếng Nhật trong *OOPSLA'97, ACM.* , 1997. (xem  
 cũng <http://squeak.cs.uiuc.edu/>).
- [6] J. Siegel, *CORBA, Nguyên tắc cơ bản và lập trình* .  
 Wiley và con trai, 1996.
- [7] E. Sentovich và cộng sự, SIS: một hệ thống cho tuần tự  
 Mạch tổng hợp, công nghệ cao. Dân biểu UCB / ERL M92 / 41, U.  
 C., Berkeley, tháng 5 năm 1992.
- [8] J.-L. Llopis và B. Pottier, Hồi xem lại Smalltalk-80  
 các khối, một trình tạo logic cho các *GPU* , trong *FCCM'96* ,  
 IEEE, 1996.
- [9] D. Micheli, *Tổng hợp cấp cao của các mạch kỹ thuật số* .  
 Tháng 11 năm 1992.