Combining and Distributing Hierarchical Systems

Chris George\textsuperscript{1} and Đỗ Tiến Dũng\textsuperscript{2}

\textsuperscript{1} United Nations University International Institute for Software Technology, Macau
\textsuperscript{2} Ministry of Finance, Hanoi, Vietnam

\textbf{Abstract.} It is possible with RAISE to specify and do most refinement in an applicative framework, and then transform the concrete applicative specification into an imperative sequential or concurrent one. This is a change from a style more appropriate to proof of refinement to a style more appropriate to implementation.

The resulting imperative specification is typically hierarchical, with upper levels calling the functions of lower ones. This paper presents a further stage of development in which the hierarchical structure is transformed into a distributed one, and components communicate asynchronously. This also allows “horizontal” communication between components of previously separate hierarchies.

A major design aim is to reuse the hierarchical specification, as far as possible extending the existing modules by standard, generic components. The method should achieve correctness by construction, and be amenable to quality control; it is an example of an engineering approach using standard components and standard assembly techniques.

The method is illustrated by collaborative work done between UNU/IIST and the Vietnamese Ministry of Finance in developing a specification of a national financial information system.

\textbf{Keywords:} Formal specification, development, refinement, reuse, restructuring, distributed systems, coordination, software engineering.

1 Introduction

We take it that engineering, as opposed to science, creates artifacts as far as possible through combining existing components. Speed and reliability are maximised and cost minimised through having to invent from scratch as little as possible. In this process engineers exploit the known properties of the components, and the known laws of the combining activity, which allows them to compute the properties of the combinations.

In this paper we describe the development of a distributed system, a financial information system, by developing first an applicative (or functional) specification and then transforming this, first into an imperative concurrent but still hierarchical system, and then into a distributed system. The first of these transformations follows the existing ideas of the RAISE method \cite{1}. The second transformation is new and is based on a small number of standard components. Hence it exemplifies an essentially engineering approach.
In section 2 we describe the problem we tackled, the development of a specification of a national financial information system for Vietnam. In section 3 we show how the RAISE method (outlined in appendix A) was applied to the problem to produce a hierarchical, synchronous system. In section 4 we describe how the transformation to a distributed, asynchronous system was achieved. Section 5 is a concluding discussion.

2 A Financial Information System

During 1996–7 United Nations University International Institute for Software Technology (UNU/IIST) in Macau and the Vietnam Ministry of Finance (MoF) undertook a joint project called MoFIT (Ministry of Finance Information Technology) aimed at doing the domain analysis and specification for a national financial information system for Vietnam. The aim was to specify the major components of such a system and the main activities and information flows, and also to train software engineers from Vietnam in the relevant techniques. As well as the first author from UNU/IIST, the project involved six mainly young software engineers from Vietnam: four from the MoF, one from the Institute of Information Technology in Hanoi and one from Hanoi University. During the 16 months of the project these people each spent between 6 and 12 months working at UNU/IIST. As well as the main work described here, studies were also made of other aspects like system security and the possible effects of changes in taxation policy. The results are described in two UNU/IIST technical reports [2,3] which in turn reference a number of more detailed project reports.

Vietnam is divided into 61 provinces, provinces are divided into districts, and districts into communes. The major government ministries reflect this structure, with offices at the national, province, district and in some cases commune levels. So, much of the collection or dissemination of information follows this hierarchical structure. In collecting information about taxes, for example, districts will supply information to their provincial offices, which will merge and perhaps summarise it and send it to the national office for the final merge into national information. Changes in taxation policy, or requests for information, flow down the hierarchy in the obvious manner.

The main component concerned with generating revenue is the taxation system, which is part of the MoF. In the first phase of the project all the engineers had experience of developing software for this system, mainly packages for particular tasks for province and district taxation offices. So in this phase we concentrated on analysing and specifying the taxation system.

In the second phase we considered other components. The treasury system is concerned with the actual collection and disbursement of money, with offices at national, provincial and district levels. The budget system is concerned with collecting budgetary estimates at the commune, district, province and national levels and, after government decision on the final figures, distributing actual annual budgets at the various levels and then monitoring these budgets. We also