Programa Doutoral MAPi
Proposta de Tema para Doutoramento

Jorge Sousa Pinto e Simão Melo de Sousa
{jsp@di.uminho.pt, desousa@di.ubi.pt}

Outubro de 2009

1 Tema

Verificação de programas concorrente de tempo real em ADA/Ravenscar

2 Unidade Proponente

Centro de Ciências e Tecnologias de Computação, Universidade do Minho

3 Enquadramento

Este trabalho é proposto no âmbito das actividades dos projectos de investigação financiados pela FCT *RESCUE, Reliable and Safe Code Execution for Embedded systems* e *FAVAS, A Formal Verification Platform for Realtime Systems*, com arranque respectivamente em Janeiro de 2008 e Janeiro de 2010, ambos envolvendo equipas do CCTC(UM) e do LIACC(UP) lideradas pelos dois proponentes deste tema. O projectos têm ainda como participantes a Universidade da Beira Interior, e no caso do segundo projecto a Universidade da Madeira a empresa Critical Software.

Ambos os projectos visam o desenvolvimento de técnicas de verificação para os sistemas críticos embebidos, e no segundo caso em particular para os sistemas de tempo real. É inegável a importância crescente do desenvolvimento de técnicas de verificação de correcção e segurança (*safety*) para os sistemas críticos desta classe, e estes projectos juntam pela primeira vez em Portugal uma equipa com múltiplas valências, quer na programação de sistemas embebidos, quer em verificação formal.

Os projectos proporcionarão, além do apoio das respectivas equipas multidisciplinares de investigadores, também apoio financeiro para deslocações, bem como para a contratação de recursos humanos (bolseiros que colaborarão de forma próxima neste trabalho de doutoramento).

O tema de tese de doutoramento aqui proposto centra-se na verificação formal de sistemas concorrentes de tempo real desenvolvidos na linguagem Ada. Reconhecidamente, o problema da verificação de aplicações concorrentes é em geral de resolução extremamente difícil, se não impossível. No contexto desta linguagem muito utilizada no contexto dos sistemas críticos, foi proposto um perfil restrito para o desenvolvimento de aplicações concorrentes, designado por *Ravenscar*. Apesar de um dos grandes objectivos do perfil ser precisamente possiblitar a verificação formal das aplicações, os trabalhos desenvolvidos até agora ficam claramente aquém das expectativas criadas na comunidades Ada.

Acrescente-se ainda que no caso do código Ada *sequencial*, popularizou-se a utilização de uma sub-linguagem do Ada, designada por SPARK. Mais do que um simples sub-conjunto sequencial do Ada, o SPARK é uma linguagem concebida com base nos princípios do *design-by-contract*, suportando verificação estática (funcional e de segurança /ausência de erros de execução) com base num VCGen e ferramenta de prova dedicadas. O *toolset* SPARK inclui também verificação estática de fluxo de dados e de informação.

4 Objectivos

Os trabalhos publicados em torno deste tema têm seguido uma de duas abordagens à verificação de código Ravenscar:

- A utilização de um *model checker* baseado em autómatos temporizados para a verificação de propriedades de aplicações. Esta abordagem obriga à construção de modelos abstractos do software, e nomeadamente de um "escalonador Ravenscar" de tarefas.
- A extensão da linguagem SPARK (designada RavenSPARK) e das ferramentas associadas, ou outras exteriores como demonstradores de teoremas, por forma a permitir a verificação de algumas propriedades de aplicações concorrentes.

O grande objectivo desta tese é, depois de efectuado um levantamento exaustivo de todo o trabalho existente e publicado sobre verificação de programas ADA / Ravenscar, e também das abordagens mais recentes às verificação (baseada em modelos) de programas concorrentes noutras linguagens de programação, propor uma abordagem integrada à verificação deste tipo de aplicações; implementar integralmente todos os componentes de software necessários para a sua utilização; e finalmente avaliar e validar a abordagem com recurso a casos de estudo de dimensão média/grande.

Apesar de esta proposta deixar em aberto a opção por uma das grandes tendências na verificação de ADA / Ravenscar (e em particular se a abrangência se limitará ou não a código RavenSPARK), é de prever que a abordagem resultante deste trabalho seja heterogénea e recorra a diferentes técnicas de verificação. A abordagem poderá resultar na proposta de uma metodologia de certificação de aplicações Ravenscar.