Desenvolvimento de Sistemas Embarcados para Aplicações Críticas

Tórgan F. Siqueira[1], **Cristina C. Menegotto**[1], **Taisy S. Weber**[1], **Jõao C. Netto**[1], **Flávio R. Wagner**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

{torgan,ccmenegotto,taisy,netto,flavio}@inf.ufrgs.br

Resumo. *Sistemas embarcados têm sido utilizados para realizar funções de segurança em aplicações críticas, prevenindo e controlando falhas ou erros. Defeitos em equipamentos controlados por tais sistemas ou nos próprios sistemas podem provocar catástrofes. A fim de reduzir riscos, o desenvolvimento dos sistemas deve seguir uma metodologia voltada à segurança. Para isto, concorrem normas, como a IEC 61508, que abrangem a quase totalidade do desenvolvimento. Neste artigo, são apresentados aspectos que devem ser levados em consideração no desenvolvimento de tais sistemas.*

1. Introdução

Sistemas embarcados são dispositivos computacionais de propósito específico, usualmente integrados a um sistema externo que desempenha alguma função específica. Tais dispositivos podem realizar operações simples, que não oferecem maiores riscos ao ambiente e às pessoas, como por exemplo as calculadoras, telefones, etc. Mas eles também podem ser responsáveis por processos industriais, sistemas de suporte à vida, etc, casos em que precisam oferecer um serviço seguro. Segurança, neste contexto, é a tradução do termo *safety*, e refere-se à segurança de funcionamento em situações críticas.

A norma internacional IEC 61508 define um sistema seguro como *"livre de riscos inaceitáveis, envolvendo prejuízos físicos ou danos à saúde de pessoas, resultantes direta ou indiretamente de danos a propriedade ou ao ambiente"* [IEC 61508 Functional Safety Zone 2005]. Os sistemas embarcados responsáveis por funções de segurança contribuem para redução dos riscos em aplicações críticas, prevenindo e controlando falhas ou erros. Uma aplicação crítica é aquela em que os riscos associados aos perigos envolvidos são considerados inaceitáveis e precisam ser tratados por meio de redução de risco [Dunn 2003]. Alguns exemplos de aplicações críticas de sistemas embarcados são: sistemas de proteção e parada de emergência em máquinas e equipamentos, sistemas distribuídos de controle de tráfego, monitoração remota de processos industriais através de rede e sistemas de controle automotivo e de vôo.

Conforme cresce a dependência da sociedade de processos automatizados, defeitos em equipamentos controlados por sistemas embarcados relacionados à segurança ou nos próprios sistemas podem provocar catástrofes. Portanto, é necessário um processo de desenvolvimento adequado para tais sistemas, voltado à segurança, que assegure que os riscos sejam mantidos em níveis aceitáveis mesmo na ocorrência de falhas.

Este artigo visa contribuir com o desenvolvimento de sistemas embarcados para aplicações críticas através do levantamento dos aspectos que devem ser levados em

consideração no desenvolvimento de tais sistemas. Na Seção 2, são apresentadas as principais características de sistemas embarcados críticos quanto à segurança. A Seção 3 aborda a prática tradicional de construção de tais sistemas e suas deficiências. Ela também introduz a norma internacional de segurança IEC 61508. A Seção 4 apresenta os aspectos desejáveis de uma metodologia de desenvolvimento para sistemas críticos, focando principalmente aspectos relacionados a software. Por fim, são apresentadas algumas notas conclusivas.

2. Sistemas Embarcados Críticos

Sistemas embarcados apresentam algumas características distintas de sistemas convencionais. Geralmente são de pequeno porte, operam com baixo consumo de energia, têm complexidade relativamente baixa e são conservadores por projeto. Neste tipo de sistema, prima-se pela robustez dos dispositivos. Técnicas de tolerância a falhas, como redundância, por exemplo, podem ser empregadas para reduzir a ocorrência de defeitos. Entretanto, além do custo econômico adicional, a complexidade agregada acaba por aumentar a probabilidade de falhas, haja vista o maior número de componentes. Por sua vez, o emprego de componentes tradicionais favorece o estabelecimento do modelo de falhas, nem sempre conhecido ou facilmente derivado da especificação [Smith 2001].

Sistemas críticos quanto à missão são caracterizados pela alta dependabilidade. Isto significa que muitos operam em condições extremas, tais como funcionamento ininterrupto, longo tempo de missão, exposição a fatores ambientais rigorosos, etc. Um defeito nestes sistemas pode comprometer a vida e/ou o meio-ambiente, ou provocar prejuízos econômicos enormes. O funcionamento destes sistemas é dito seguro, isto é, uma falha não deve provocar efeitos catastróficos. O sistema, se falhar e não puder se recuperar, deve atingir um estado seguro. Mas definir um estado seguro nem sempre é trivial, como no caso dos controladores de navegação ou de controle de reações químicas, pois o restante do sistema ainda precisa continuar operando.

O projeto de sistemas embarcados de missão crítica exige alto grau de elaboração. Além dos aspectos de engenharia, é preciso considerar a missão crítica que desempenham, o que pode gerar conflitos na especificação (custo, consumo, porte, por exemplo). Incluem-se ainda os fatores operacionais, ou seja, o ambiente de operação e a manutenção. Por vezes, a manutenção destes sistemas é onerosa, como no caso da parada de plantas industriais, processos físico/químicos e a interrupção de serviços essencias. Por outras, a manutenção é inviabilizada pela impossibilidade de acessar fisicamente o dispositivo. Testar por exaustão o dispositivo, antes de empregá-lo, também não é uma opção viável, já que depende da análise do modelo de falhas, por vezes incompleto, ou pode ignorar variáveis e situaçoes previsíveis e imprevisíveis do ambiente de operação.

3. Desenvolvimento de Sistemas Críticos

Usualmente, o desenvolvimento de sistemas críticos é feito *ad-hoc*, empregando disciplinas de projeto da engenharia tradicional. Para atender aos requisitos de segurança, é feita uma análise caso-a-caso do modo de funcionamento e, de acordo com a necessidade, tenta-se obter a máxima cobertura de falhas possível. Os problemas decorrentes desta abordagem são vários, mas a falta de padronização, em particular, compromete dois outros aspectos importantes: a avaliação do correto funcionamento e a verificação formal do projeto.

A avaliação do correto funcionamento envolve estabelecer correspondência entre implementação e especificação, tanto para funcionamento normal quanto para ocorrência de falhas. Aqui surge o primeiro problema: nem sempre é possível cobrir todas as possibilidades de falhas, resultando no teste de apenas um subconjunto. Usualmente, uma bateria de testes é empregada para conduzir esta etapa, refletindo outro fator problemático: a especificação dos testes. Os testes deveriam acompanhar todas as etapas, da especificação à operação do dispositivo em campo, mas, em geral, esta prática não é comum.

A verificação formal é o processo pelo qual se verifica se há correspondência unívoca entre implementação e especificação, através do uso de métodos formais. A unicidade da correspondência garante que a especificação foi implementada na íntegra e que nada além foi implementado. O projeto é avaliado principalmente pela documentação que gerou, da especificação de requisitos aos diagramas esquemáticos. O problema reside no fato de nem sempre a documentação ser corretamente gerada, pois as práticas usuais são omissas ou incompletas em vários aspectos.

Para produtos que devam atender um determinado padrão de qualidade, ou obter uma certificação, deficiências na avaliação ou verificação podem comprometer o projeto. É possível certificar um produto já em produção e operação, mas isto é mais difícil. Por vezes, requer-se modificações em partes dos mesmos, ou, então, a certificação é aplicável somente a alguns subsistemas (dependendo da especificação, isto pode ser suficiente). Exemplos deste procedimento podem ser encontrados em [Exida 2006].

No que diz respeito ao hardware, o desenvolvimento de sistemas embarcados já incorporou parte dos conhecimentos da área de tolerância a falhas. Os circuitos são projetados de modo a tolerarem condições elétricas, térmicas e ambientais adversas, alguns oferecem redundância de componentes e outros contam com proteções extras para atingirem um estado seguro em caso de falha. Ainda, para missões críticas, os próprios circuitos são redundantes.

O desenvolvimento de software para sistemas embarcados relacionados à segurança é um processo bastante delicado e ainda não alcançou o mesmo estágio de consenso de práticas comuns e preceitos gerais que o hardware [Johnson 2003]. Embora muitas empresas adotem bons princípios de engenharia de software, medidas de segurança são geralmente relegadas a segundo plano ou mesmo ignoradas. Seguir bons princípios não é suficiente para garantir segurança e, ao contrário do hardware, não há mecanismos bem definidos para análise da taxa de defeitos de software. Assim, a ênfase está no modo como as atividades de engenharia são realizadas e na inclusão de técnicas e medidas que auxiliam na obtenção de segurança.

Há muitas diferenças entre o processo de desenvolvimento de software em projetos tradicionais e em projetos com características de segurança. Muitas vezes, por exemplo, na escolha de ferramentas de projeto e desenvolvimento, de tradução, de teste e depuração e de gerência de configuração, os projetos tradicionais só consideram aspectos ergonômicos, de facilidade e comodidade de uso. A escolha da linguagem de programação adequada para segurança também é um fator decisivo, atenção que usualmente não é despendida em projetos tradicionais.

Existem normas que tratam do processo de desenvolvimento de sistemas relacionados à segurança, por exemplo, o padrão Std-Mil-882D [Mil-Std-882D 2000] e a

norma IEC [IEC 61508 2000]. A norma internacional IEC 61508, desenvolvida pela Comissão Eletrotécnica Internacional (*International Electrotechnical Commission*), é um padrão para segurança funcional de equipamentos elétricos, eletrônicos e eletrônicos programáveis relacionados à segurança, independente do domínio de aplicação dos mesmos. Ela pode ser aplicada, portanto, ao desenvolvimento de sistemas embarcados críticos.

A IEC 61508 apresenta uma abordagem genérica para todas as fases e atividades do ciclo de vida, incluindo tanto procedimentos técnicos quanto administrativos necessários para atingir a segurança funcional necessária. A norma descreve requisitos específicos para o desenvolvimento dos equipamentos elétricos, eletrônicos e eletrônicos programáveis relacionados à segurança, e também para o desenvolvimento de software para os equipamentos eletrônicos programáveis [Fowler and Bennett 2000]. Ela utiliza o conceito de nível de integridade de segurança (*Safety Integrity Level - SIL*) para especificar o nível desejado de integridade para as funções de segurança a serem implementadas. De acordo com a norma, existem os níveis de integridade de segurança 1, 2, 3 e 4, e as exigências crescem com o aumento do SIL requerido. Para a determinação do SIL, a norma utiliza uma abordagem baseada em riscos.

4. Aspectos de uma Metodologia de Desenvolvimento

O desenvolvimento de sistemas embarcados relacionados à segurança deve seguir uma metodologia que possibilite atingir a redução dos riscos a um nível aceitável. O emprego de normas de segurança, como a IEC 61508, desde o princípio do projeto e ao longo de todas as fases do mesmo, viabiliza a construção desses sistemas.

Segundo o *framework* da norma IEC 61508, o desenvolvimento de um sistema relacionado à segurança deve começar pela especificação dos requisitos globais de segurança. Primeiramente, é preciso adquirir um bom entendimento do equipamento sob controle e seu ambiente (físico, legislativo, etc) e, a partir deste conhecimento, determinar o escopo global do sistema em termos dos limites do equipamento e seu sistema de controle. Segue-se uma análise dos perigos envolvidos e avaliação dos riscos, considerando a freqüência dos eventos perigosos e as conseqüências a eles associadas, com base na qual são especificados os requisitos globais de segurança em termos dos requisitos das funções de segurança e requisitos de integridade de segurança. Após a especificação dos requisitos globais de segurança, os mesmos devem ser alocados aos sistemas relacionados à segurança, e um nível de integridade de segurança deve ser alocado a cada função. Feita essa alocação, pode-se partir para o desenvolvimento dos sistemas e seu software, paralelamente aos planejamentos de operação, manutenção, validação, instalação e delegação. Deve-se, então, realizar a instalação, delegação e validação de segurança dos sistemas. Por fim, o *framework* trata da operação, manutenção, modificação e realimentação, além da retirada dos sistemas relacionados à segurança [Faller 2001].

No que se refere à documentação, o mais importante é o seu conteúdo. Ela deve prover as informações necessárias para que todas as fases dos ciclos de vida possam ser efetivadas, além da gerência, verificação e avaliação da segurança funcional. A documentação deve ser correta, concisa, de fácil entendimento pelos envolvidos no projeto, acessível e manutenível. A sua estrutura pode levar em consideração procedimentos das empresas e as práticas de trabalho de setores de aplicação específicos.

A metodologia de desenvolvimento não pode subestimar os procedimentos admin-

istrativos que garantam a efetiva implementação dos procedimentos técnicos. No gerenciamento de segurança funcional, são especificadas as atividades técnicas e gerenciais a serem realizadas, incluindo as fases dos ciclos de vida a serem aplicadas, assim como as responsabilidades de pessoas, departamentos e organizações responsáveis por elas. A norma apresenta os ciclos de vida a serem usados como base para exigir conformidade com ela, mas outros podem ser usados, desde que todos os objetivos das cláusulas da norma sejam atingidos. Dentre os requisitos para fases e atividades dos ciclos de vida de segurança estão a aplicação de técnicas e medidas para evitar e controlar falhas e defeitos. A combinação apropriada de técnicas e medidas deve ser escolhida de acordo com a aplicação específica, durante o planejamento de segurança, pois não existe algoritmo para a combinação das técnicas e medidas corretas para uma aplicação em particular.

No desenvolvimento relacionado à segurança, é importante que seja estabelecido um sistema de gerência da qualidade de software que contemple o planejamento de segurança funcional e a gerência de configuração de software. "O planejamento de segurança funcional deve definir a estratégia para obtenção, desenvolvimento, integração, verificação, validação e modificação de software de acordo com o rigor do nível de integridade de segurança do sistema relacionado à segurança" [IEC 61508-3]. A gerência de configuração de software deve proporcionar a identificação adequada de componentes de software em diferentes versões e estabelecer procedimentos de controle de modificações.

Há diversas recomendações sobre a escolha de uma linguagem de programação adequada. Recomenda-se o uso de linguagens fortemente tipadas, as quais reduzem a probabilidade de falhas por permitir um alto nível de verificação por parte do compilador [IEC 61508-7]. O uso de um subconjunto de uma linguagem, que exclui construções propensas a erros ou difíceis de analisar, também reduz a probabilidade de introduzir falhas e aumenta a probabilidade de detecção de falhas remanescentes, sendo altamente recomendado [IEC 61508-7].

Alguns aspectos de programação tornam a verificação do software difícil e devem ser evitados. Dentre eles, pode-se enumerar: saltos incondicionais, recursão, manipulação de ponteiros, *heaps* ou qualquer tipo de variáveis e objetos dinâmicos, tratamento de interrupções ao nível de código fonte, declaração ou inicialização implícita de variáveis, múltiplas entradas e saídas de laços, blocos ou subprogramas, entre outras. Algumas técnicas devem ser evitadas, tais como reconfiguração dinâmica e correção de falhas por inteligência artificial.

Escolher ferramentas para o projeto de software relacionado à segurança requer cuidado. Sempre que possível, as ferramentas devem ser certificadas ou ter sido utilizadas com sucesso em projetos anteriores, para garantir um nível de confiança na correção de suas saídas. A certificação de uma ferramenta é geralmente realizada por um órgão independente, contra um determinado critério.

5. Conclusão

Sistemas embarcados, embora tendam à simplicidade, quando realizam funções de segurança, devem obedecer a determinados preceitos, que garantam seu funcionamento seguro. Em sistemas críticos quanto à missão, como suporte à vida, controle de plantas nucleares/industriais, sistemas de navegação, etc, pode-se estabelecer determinados níveis de integridade de segurança, SIL, os quais indicam o grau de dependabilidade apropriado.

Para atender estes requisitos, a prática tradicional de desenvolvimento *ad-hoc* deve ceder lugar a práticas mais restritas, respaldadas por normas consolidadas. Uma destas normas é a IEC 61508, que abrange praticamente todos os aspectos envolvidos no desenvolvimento de sistemas críticos. O limite imposto na criatividade de projeto é compensado pelo estabelecimento de melhores práticas, um processo que evolui positivamente.

O desenvolvimento de software é um ponto-chave e deve abranger aspectos gerenciais, administrativos e técnicos próprios, definindo um ciclo de vida fortemente orientado à segurança. Técnicas de software bem estabelecidas, restritas e com apoio de ferramentas adequadas são fortemente recomendadas. A escolha da linguagem de programação e das ferramentas requer grande cuidado. É recomendado o uso de subconjuntos da linguagem e que certas construções de programação sejam evitadas. Em particular, técnicas inovadoras e de muito alto nível devem ser evitadas, em especial o uso de correção de falhas através de inteligência artificial.

Conclui-se que a solução mais duradoura e eficiente para a disseminação de competência no país nesta área é a adoção de disciplinas curriculares que abordem o assunto de sistemas críticos. Preparar o profissional com melhores práticas, conhecimento das normas e mentalidade adequada a projetos desta natureza, resultará numa cultura empresarial mais receptiva e melhor preparada para atender demandas de tais sistemas.

Referências

Dunn, W. R. (2003). Designing safety-critical computer systems. *IEEE Computer*, 36(11):40 – 46. ISSN 0018-9162.

Exida (2006). Product Report. http://www.exida.com/products2/reports.asp.

Faller, R. (2001). Project experience with IEC 61508 and its consequences. In *SAFECOMP 2001*, volume 2187 of *Lecture Notes in Computer Science*, pages 200 – 214. Springer.

Fowler, D. and Bennett, P. (2000). IEC 61508 - a suitable basis for the certification of safety-critical transport-infrastructure systems?? In *SAFECOMP 2000*, volume 1943 of *Lecture Notes In Computer Science*, pages 250 – 263. Springer.

IEC 61508 (2000). International Electrotechnical Commission IEC 61508, part 1 to 7; Functional Safety of Electrical, Electronic and Programmable Electronic Safety-Related Systems. http://www.iec.ch/functionalsafety.

IEC 61508 Functional Safety Zone (2005). Functional safety and IEC 61508. http://www.iec.ch/zone/fsafety/pdf_safe/hld.pdf.

Johnson, C. (2003). Using IEC 61508 to guide the investigation of computer-related incidents and accidents. In *SAFECOMP 2003*, volume 2788 of *Lecture Notes in Computer Science*, pages 410 – 424. Springer.

Mil-Std-882D (2000). Standard Practice for System Safety, Mil-Std-882D. US Dept. of Defense; http://www.geia.org/sstc/G48/882d.pdf.

Smith, D. J. (2001). *Reliability, Maintainability and Risk*. Elsevier Butterworth-Heinemann, 6^{th} edition.