

Um Estudo das Propostas de Modelos de Componentes para Sistemas de Tempo Real

Cássia Yuri Tatibana , Rômulo Silva de Oliveira , Carlos Montez

¹DAS - Departamento de Automação e Sistemas UFSC - Universidade Federal de Santa Catarina
Campus Universitário, Caixa Postal 476 - CEP 88040-900 - Trindade - Florianópolis - SC

{cytatiba,romulo,montez}@das.ufsc.br

Resumo. *Sistemas de tempo real são cada vez mais complexos e presentes em diferentes áreas de atuação. O desenvolvimento desses sistemas usando abordagens baseadas em componentes tem se mostrado uma alternativa promissora. Este artigo objetiva uma comparação qualitativa das principais propostas encontradas na literatura.*

1. Introdução

O desenvolvimento de software baseado em componentes permite o reúso de serviços que podem ser compostos, configurados e instalados para construir aplicações de maneira rápida e robusta. Essa tecnologia é empregada na redução do tempo e esforço requeridos no desenvolvimento de aplicações complexas em muitos domínios, inclusive tempo real. Os sistemas de tempo real crescem, cada vez mais, em variedade e complexidade e o desenvolvimento de software baseado em componentes está ganhando importância entre os trabalhos dessa área.

A diversidade de interpretações do conceito de componente e de objetivos de pesquisa no contexto de tempo real estimula a concepção de uma grande quantidade de trabalhos envolvendo ambos os temas [16, 7, 5, 11, 14, 15, 10, 6, 13]. Embora estejam dentro de um mesmo tema, devido à abrangência deste, os trabalhos abordam diferentes aspectos de componentes, enfatizando desde a definição de modelos e *frameworks* para componentes até o estudo de ferramentas necessárias para o desenvolvimento, análise e implantação de um determinado modelo de componentes. Os diferentes enfoques de cada trabalho dificultam a classificação e comparação entre eles, contudo, enriquecem a aplicabilidade e o conhecimento sobre o tema.

Este artigo objetiva apresentar alguns dos principais trabalhos na área e oferecer uma comparação qualitativa dos mesmos. A seleção dos mesmos considerou o estado de amadurecimento das propostas e a consistência das abordagens quanto a fundamentação teórica e aplicabilidade. Não é de conhecimento dos autores a existência de outros estudos comparativos na mesma linha de pesquisa apresentada aqui.

Neste texto, após a apresentação de cada um dos trabalhos selecionados (seção 2) - RCCF[14, 15], ACCORD[10], Cadena[6] e CIAO[13] - procurou-se estabelecer uma forma de comparação entre eles respeitando os objetivos de cada um (seção 3). No final, é apresentada uma discussão salientando as principais propriedades de cada trabalho e comparando a maneira com que estas mesmas propriedades são tratadas por outras abordagens.

2. Propostas de Modelos para Componentes de Tempo Real

Cada trabalho apresenta uma proposta diferente e uma abordagem ou uma combinação de mecanismos que contribui para o desenvolvimento de sistemas baseados em componentes com restrições temporais.

2.1. RCCF

O *framework* de configuração (*customization*) de componentes de tempo real é apresentado em [14, 15]. O RCCF (*Real-time Component Customization Framework*) propõe um modelo de componentes que define desde a arquitetura do componente até as ferramentas a serem utilizadas na implantação dos sistemas baseados nestes componentes. Essa proposta objetiva amenizar problemas encontrados durante o processo de integração destes sistemas. A solução apontada envolve a utilização de componentes que possam ser configurados de acordo com as necessidades do sistema final. Nessa abordagem, componentes que não oferecem a capacidade de configuração desejada, podem ser estendidos.

A configuração e a extensão do componente são realizadas através de interfaces específicas, meta-descritores e ferramentas. A configuração é limitada pelas propriedades configuráveis, e a extensão do *framework* de componente é feita pela inserção de pontos de configuração adicionados através de ferramentas apropriadas. A abordagem oferece flexibilidade de configuração em tempo de projeto para cada tipo de aplicação. Entretanto, a arquitetura do componente no modelo e os mecanismos usados para sua configuração e extensão tornam a abordagem bastante restrita, pouco portátil. Os autores declaram que a abordagem é independente de qualquer tecnologia de *middleware*, foi testada através do DCOM [1].

2.2. ACCORD

O trabalho apresentado em [10] introduz o conceito de desenvolvimento de sistemas de tempo real baseado em componentes orientado a aspectos. A proposta apresentada, ACCORD - *Aspectual Component Based Real-time System Development*, também é completa, no sentido de definir o desenvolvimento de um sistema de tempo real baseado em componentes desde a arquitetura do componente até a fase de integração e implantação da aplicação final. O foco desse trabalho é a utilização do desenvolvimento baseado em aspectos como meio de inserção de propriedades temporais em componentes. A abordagem é apontada como a melhor solução para o problema de inserir aspectos de tempo real, que entrecortam um sistema baseado em componentes. O trabalho envolve a definição de uma arquitetura de componente com aspectos (RTCOM) e um conceito de desenvolvimento (ACCORD) fundamentado na decomposição em componentes e aspectos.

O processo de desenvolvimento de sistemas a partir desta proposta envolve a definição de aspectos de aplicação, execução e composição que podem ser armazenados em uma biblioteca e utilizados conforme o mesmo princípio da utilização de componentes. As funcionalidades de componentes RTCOM e os aspectos são implementadas a partir de um conjunto pré-estabelecido de primitivas cujo valor de WCET é conhecido. Desta forma, estender a funcionalidade do componente implica na adição de nova combinação destas primitivas (aspectos) nos pontos de inserção e adição de valores de WCET, contribuindo com a previsibilidade temporal do sistema.

Como pontos de inserção são operações, os aspectos são inseridos sob a forma de *advices*: antes, depois, ou no lugar do mesmo. Graças ao uso de primitivas como blocos de construção da funcionalidade do componente e de aspectos, a análise temporal é possível em aplicações baseadas em componentes RTCOM. A proposta, apesar de bastante detalhada quanto à fundamentação teórica, até o momento, não é destinada a ambientes distribuídos.

2.3. CIAO/QuO

No trabalho [13] é apresentada uma proposta de desenvolvimento de sistemas distribuídos embutidos de tempo real baseado em componente com requisitos rígidos de qualidade de serviço (QoS - *Quality of Service*) através de tecnologias de *middleware* de componentes existentes. A proposta apresenta a combinação do CIAO - *Component Integrated ACE ORB* [12] e QuO [17] para provisão de requisitos de QoS estático e dinâmico. Esse trabalho parte da combinação de tecnologias e ferramentas existentes que resultam em uma abordagem de desenvolvimento de sistemas baseados em componentes com propriedades de QoS definidas tanto em tempo de projeto (CIAO) como em tempo de execução (QuO). A capacidade dessa abordagem em

oferecer QoS para componentes reside na utilização do TAO [8] como ORB de tempo real e na extensão do CCM [3].

O processo de desenvolvimento de componentes CIAO/QuO procura ao mesmo tempo preservar e estender o modelo CCM. A proposta do CIAO é separar as propriedades de aplicações e transformá-las em unidades que podem ser instaladas e configuradas no modelo de componentes já existente (CCM). O uso destas unidades permite ao desenvolvedor selecionar políticas de tempo real e aspectos de QoS que se aplicam tanto ao servidor quanto aos clientes. O CIAO tem capacidade para configuração de políticas de CPU, de comunicação e de *middleware* distribuído.

O QuO é usado neste trabalho para prover aos sistemas QoS dinâmico e adaptabilidade. Permite a especificação de requisitos QoS, monitoração e adaptação do sistema às mudanças dos requisitos em tempo de execução. Todas as funcionalidades do QuO são encapsuladas em uma unidade de reuso de comportamento, Qosket. O CIAO é capaz de implantar o Qosket do QuO e fazer com que esta unidade de comportamento seja parte de um componente CIAO.

O resultado da integração do CIAO e o QuO é um sistema que em tempo de execução trabalha com dois fluxos de dados. O caminho funcional, destinado ao fluxo de dados específicos da aplicação e o caminho de sistema QoS, que determina o quão bem o sistema funciona tendo como base os requisitos QoS especificados pelo desenvolvedor.

2.4. Cadena

O Cadena [6], ambiente de desenvolvimento de arquitetura de componente para sistemas aviônicos, objetiva a construção e modelagem de sistemas CCM [3], empregado em aplicações da Boeing. Este ambiente é formado por um conjunto de ferramentas para análise de comportamento de sistemas CCM. O processo de desenvolvimento da aplicação é realizado através do OpenCCM [9]. O Cadena funciona como uma camada sobre o OpenCCM adicionando formulários às IDLs da aplicação em desenvolvimento que permitem a especificação de comportamento temporal e dependências inter e intra-componentes. Através dos formulários, o Cadena permite a especificação de restrições temporais e distribuição. O Cadena traduz o componente projetado, os formulários inseridos e referências de correção em um modelo de entrada para o analisador DSpin [4].

Componentes construídos com o Cadena são componentes OpenCCM que reagem a eventos. Toda a comunicação nesses sistemas é realizada através de eventos cuja taxa de envio é associada a portas pelo Cadena de acordo com a especificação descrita no formulário. O analisador verifica a viabilidade do sistema resultante e através de iterações com o desenvolvedor, permite o ajuste de requisitos temporais e da alocação de componentes do sistema final.

3. Comparação entre as Abordagens

Esta seção estabelece uma comparação direta entre os trabalhos com o objetivo de facilitar a análise. Cada característica ou propriedade considerada relevante é abordada a seguir. Os critérios de comparação são utilizados devido a influência de seu papel no tratamento de aspectos temporais durante o desenvolvimento dos sistemas.

3.1. Diversidade de Objetivos

O objetivo de cada trabalho difere a partir da própria motivação dos mesmos. O RCCF visa facilitar o processo de integração de componentes de tempo real a partir de um modelo de componentes com capacidade de ser configuração e estendido. O objetivo do ACCORD é introduzir requisitos temporais através de aspectos no desenvolvimento de componentes de tempo real. O CIAO/QuO compõe um ambiente que visa o desenvolvimento de sistemas baseados em componentes com requisitos de QoS estático e dinâmico. E

finalmente o Cadena, tem como objetivo a modelagem e análise de modelos de sistemas CCM para software aviônico.

3.2. Modelos de Componente

Em decorrência da diversidade de objetivos, as propostas geraram seus próprios métodos ou mecanismos de desenvolvimento de sistemas baseados em componentes. O modelo de componente determina a arquitetura do componente usado e/ou definido pela abordagem. O RCCF desenvolve um modelo de componentes próprio. A arquitetura do componente RCCF não se baseia em qualquer modelo de componentes existentes (CCM [3], EJB [2] ou DCOM [1]). Um componente RCCF, em tempo de execução, não conta com o suporte de um *container*. Todas as ações referentes ao componente, sejam elas funcionais ou não, são implementadas no próprio componente. A organização interna desta arquitetura em espaços de solução e de meta-serviço delimita a parte funcional específica de cada aplicação e a parte de serviços comuns a todo componente RCCF.

Os componentes do ACCORD, (RTCOM) também não têm base em modelos estabelecidos no mercado. Internamente, componentes RTCOM apresentam uma primeira divisão entre parte funcional, parte de tempo de execução e parte de composição. A parte funcional deste componente se divide em operações e mecanismos. Os mecanismos são usados como blocos de construção de operações e aspectos. Por isso, este modelo de componentes oferece informações sobre WCET mesmo com relação a componentes estendidos. A extensão é feita através de recombinação dos mecanismos já existentes. A parte de tempo de execução se divide em restrições temporais, demanda de recursos e portabilidade. A parte de composição do componente RTCOM se refere à verificação de compatibilidade em relação a funcionalidades, requisitos temporais, disponibilidade de recursos e plataforma do componente.

Os componentes Cadena e CIAO/QuO são componentes baseados no CCM. A arquitetura ou o modelo de componentes não é o foco de nenhum desses dois trabalhos. O esforço é centrado no componente em tempo de execução e nos mecanismos de comunicação e provisão de requisitos temporais ou de QoS. Como no CCM, os componentes CIAO/QuO desfrutam do suporte de um *container*, responsável por toda a parte não funcional do componente. O componente CIAO/QuO implementa somente a lógica da aplicação. As propriedades referentes ao comportamento temporal deste componente são decorrentes de configuração de mecanismos aplicados ao *container*.

3.3. Mecanismos de Configuração de Componentes

A configuração de componentes é a definição de valores de propriedades que determinam o comportamento deste em tempo de execução. Este processo implica na adequação de um componente que é candidato a fazer parte da aplicação em desenvolvimento. O quanto a configuração influencia no comportamento da aplicação final depende tanto da maneira como a configuração é realizada quanto das propriedades que podem ser configuradas.

A configuração de um componentes RCCF ocorre durante o processo de integração. A partir dos requisitos do componente, deve ser feito o mapeamento das tarefas que implementam a funcionalidade do componente e documentado em formato de meta-descrição.

A configuração de um componente RTCOM (ACCORD) é feita através da inserção de aspectos durante a integração de componentes. De certa forma, pode se dizer que os aspectos configuram os componentes RTCOM (tanto a parte funcional e não funcional). A inserção de aspectos pode ser aplicada a cada componente ou ao sistema resultante da integração dos mesmos, o que implica em dois momentos possíveis de configuração. Esta forma de configuração de componentes objetiva encapsular em aspectos, propriedades que entrecortam o sistema.

O CIAO/QuO estende os descritores CCM para configurar componentes. A configuração é aplicada ao *container*. Os descritores determinam o uso dos mecanismos, políticas e módulos de QoS disponibili-

zados no *container* que passa a estar preparado para utilização destes elementos, inseridos também no lado cliente da aplicação.

O Cadena também configura componentes através de descritores. Os formulários Cadena equivalem aos descritores usados pelo CIAO/QuO. A configuração do Cadena também é aplicada ao *container* e está centrada na comunicação de componentes, uma vez que estes são componentes disparados por eventos.

3.4. Propriedades Configuráveis

As propriedades configuráveis de cada abordagem diferem de acordo com a capacidade de configuração suportada pelo componente. Na maioria dos trabalhos, somente a parte não funcional da aplicação pode ser configurada, o que inclui os requisitos temporais das mesmas.

As propriedades configuráveis de componentes RCCF pertencem ao espaço de meta-serviço, são propriedades não funcionais, dentre elas, propriedades de natureza temporal. Como o *framework* tem capacidade de lidar somente com aplicações que usam algoritmos de escalonamento de prioridade fixa, a meta-descrição é estruturada para definir a prioridade, *deadline*, período e política de sincronização de *threads*. A capacidade de extensão de *framework* deste trabalho permite que ele seja estendido para satisfazer aplicações que exigem propriedades adicionais ou que não possam ser implementadas com um algoritmo de escalonamento de prioridade fixa. Além de propriedades de tempo real, também são definidos na configuração propriedades de compatibilidade do componente com a aplicação final.

Na abordagem do ACCORD a configuração é feita através de aspectos que podem alterar também a funcionalidade do componente. Desde que os aspectos sejam construídos a partir dos mecanismos do componente, a inserção dos mesmos pode ser efetuada. No ACCORD, podem ser configurados aspectos de otimização de memória, sincronização, segurança, propriedades e políticas de tempo real, compatibilidade, portabilidade, versão de componente e flexibilidade (capacidade de agregar novos aspectos).

O CIAO/QuO permite a configuração do ORB do lado cliente, da política de QoS, da interação com mecanismos de garantia de QoS presentes no *container*, da política de escalonamento, dos recursos de comunicação, do *middleware* distribuído e dos mecanismos que monitoram o nível de QoS sendo oferecido em tempo de execução.

A configuração realizada pelo Cadena permite a definição de taxas de envio de mensagens associadas às portas de eventos de cada componente. As dependências entre componentes também são especificadas na configuração para que sejam consideradas durante a implantação dos componentes. Em tempo de implantação, uma ferramenta é responsável pela alocação de componentes em hospedeiros, ela também permite a especificação de hospedeiros específicos.

3.5. Formas de Implementar a Configuração

No RCCF a configuração especificada pelo integrador na meta-descrição é traduzida em meta-código. O meta-código, por sua vez, é aplicado via processador de meta-código ao componente. A configuração é aplicada ao componente, que possui interfaces para este fim. A ferramenta automatiza a configuração, mas a capacidade de modificar propriedades reside no próprio componente.

No ACCORD, aspectos são construídos em conjunto com componentes. Os componentes implementam a funcionalidade e os aspectos implementam propriedades que afetam o desempenho e semântica do sistema. A configuração é aplicada diretamente no componente RTCOM que usa interfaces de composição para integrar aspectos.

O CIAO estende descritores de componentes e *assembly* do CCM para compor os mecanismos e implementações de provisão de QoS dentro de *containers*. A configuração exige o TAO como ORB subjacente, a configuração da aplicação pressupõe módulos de *middleware* presentes neste ORB. No lado cliente, o ORB é configurado para que possa oferecer suporte às diversas políticas de QoS promovendo

desta forma, a provisão de QoS fim a fim. Toda interface requerida para lidar com a provisão de QoS é implementada pelo *container* CIAO, portanto, a configuração é aplicada ao *container* do componente. Através do QuO, essa abordagem permite especificar outros requisitos de QoS, e elementos do sistemas a serem monitorados e controlados a fim de tornar possível a adaptação do sistema às variações nos recursos em tempo de execução.

Através de ferramentas do Cadena as especificações do formulário são agregadas à IDL que define os componentes. A configuração de taxas de envio de mensagens é aplicada ao canal de eventos do CORBA de tempo real utilizado, fundamental para a proposta do Cadena.

Diferente da abordagens RCCF e ACCORD, o CIAO/QuO e Cadena não exigem o código fonte dos componentes em tempo de configuração. Por outro lado, a configuração é bastante restrita quando comparadas à aplicada aos componentes no RCCF e ACCORD.

3.6. Extensões de Componente

A extensão de componentes faz parte da configuração nas abordagens RCCF e ACCORD, que são abordagens centradas na arquitetura do componente. No RCCF esta etapa pode ser aplicada quando a configuração de componentes via alteração de parâmetros não é suficiente para adequar o componente à aplicação final. No ACCORD é parte essencial da configuração.

A extensão do *framework* de componentes no RCCF objetiva permitir que o desenvolvedor determine quais serão as propriedades configuráveis de seus componentes. A ferramenta deve interpretar a especificação de pontos de inserção e gerar o meta-código correspondente a partir da especificação e do código fonte do componente. O código resultante é compilado em um componente binário e armazenado. O princípio usado pela abordagem é o mesmo da abordagem de desenvolvimento baseado em aspectos. Quando o componente não apresenta propriedades configuráveis suficientes para a adequação do componente aos requisitos de projeto, é estendido.

Os componentes RTCOM são estendidos através da inserção de aspectos em pontos de inserção. No caso do RTCOM, aspectos são construídos a partir de primitivas, por isso a extensão do componente é limitada pelas primitivas pré-estabelecidos do componente. Entretanto, esta limitação permite que o WCET de componentes RCCF, mesmo quando estendidos, seja calculado.

O CIAO/QuO e Cadena não permitem a extensão de seus componentes. A configuração de componentes é o único meio de modificação dos mesmos, e em geral, é feita somente pela definição de propriedades que determinam como ou se o componente usará os mecanismos disponibilizados para ele pelo *container*.

3.7. Requisitos Temporais Suportados

Devido à estrutura de cada modelo, o tratamento de requisitos temporais difere em cada abordagem. No RCCF e ACCORD, a manutenção de requisitos temporais é atribuída ao componente. No Cadena e CIAO/QuO esta tarefa é do *container*, o componente contém apenas a lógica da aplicação. A definição de requisitos temporais da aplicação no RCCF, ACCORD e Cadena só pode ser feita em tempo de projeto. O único trabalho apresentado que gerencia requisitos de QoS em tempo de execução é o CIAO/QuO, que também trata de configuração de requisitos de QoS no lado cliente, o que não ocorre nas outras abordagens.

O tratamento de requisitos temporais no RCCF é realizado por interfaces do próprio componente. A interface de extensão de componentes do RCCF exige um mapeamento das tarefas que compõem a funcionalidade sendo agregada. A prioridade, *deadline*, frequência e mapeamento de relações entre estas tarefas também são parâmetros obrigatórios. No ACCORD a inserção de aspectos é controlada pelo uso de primitivas como blocos de construção. As primitivas são pré-estabelecidos e possuem WCET conhecido. A relação entre as primitivas e entre as operações deve ser precisamente detalhada para permitir a análise

temporal. Interfaces de composição também verificam a compatibilidade temporal do componente com o aspecto sendo inserido.

Embora o CIAO/QuO e Cadena sejam desenvolvidos sobre um modelo de componentes para sistemas de propósito geral, autores de ambas as abordagens declaram que são ferramentas para sistemas de tempo real crítico. Nenhum tratamento com relação a arquitetura interna do componente é relatada no que se refere a previsibilidade temporal, mas ambas as ferramentas trabalham com a configuração da camada subjacente da aplicação. O ORB de tempo real é citado como parte fundamental destas abordagens.

3.8. Interfaces Providas

Embora existam divergências quanto a características de componentes, em todas as abordagens o único meio de interação com o componente em tempo de execução é através de suas interfaces.

O componente RCCF divide suas interfaces em interfaces do espaço de solução, que provêm a parte funcional do componente, e meta-interfaces, comuns a todos os componentes RCCF e responsáveis por implementar a configuração e extensão do componente. As meta-interfaces do componente RCCF implementam o papel do *container* dos modelos tradicionais, e são responsáveis por controlar a adição ou remoção de meta-interfaces, prover serviços para recuperação de informação sobre o componente, gerenciar a configuração do componente, informar quais as propriedades configuráveis do componente, prover reflexão de configuração, especificar e realizar a verificação de compatibilidade do componente, publicar funções para informar a respeito de propriedades temporais.

As interfaces dos componentes RTCOM são classificadas em interfaces funcionais, de configuração e de composição. As interfaces de configuração provêm informações sobre o comportamento temporal de cada componente e refletem aspectos de tempo de execução do mesmo. As interfaces de composição são ativadas somente a partir da configuração do componente, elas são usadas para composição de aplicações e para inserção de aspectos.

Os componentes Cadena e CIAO/QuO possuem interfaces condizentes com o modelo de componentes CORBA. As interfaces destes componentes implementam somente as funções pertencentes à lógica da aplicação.

4. Considerações Finais

Os diferentes objetivos de cada trabalho tornam difícil estabelecer uma comparação de suas abordagens com relação ao tratamento de requisitos temporais. Principalmente na proposta do RCCF e ACCORD, a previsibilidade temporal é bastante enfatizada na concepção de cada componente. Por outro lado, este cuidado com relação ao comportamento temporal desde a arquitetura dos componentes torna a abordagem extremamente dependente de ferramentas. Ao mesmo tempo que trabalhos com boa fundamentação teórica (RCCF e ACCORD) promovem um grande potencial para geração de aplicações de tempo real baseadas em componentes com previsibilidade temporal, deixam a desejar quanto ao estágio de desenvolvimento que se encontram. Não há disponibilidade de ferramentas ou detalhamento quanto aos mecanismos que automatizam a implementação destas abordagens. Não há informações a respeito da continuidade ou não destes projetos ou listas de discussões a respeito do desenvolvimento destes trabalhos.

O CIAO/QuO e Cadena enfocam a comunicação entre componentes e a monitoração do comportamento da aplicação formada por estes componentes em tempo de execução. Ao contrário do RCCF e ACCORD, esses trabalhos não tratam de mapeamento de tarefas internas ao componentes ou implementam qualquer ferramenta de análise de WCET, por exemplo. O componente em sí não recebe qualquer tratamento quanto ao comportamento temporal, somente a parte não funcional do componente, implementado pelo *container* recebe os mecanismos e configurações de tempo real. Por outro lado, as ferramentas destas abordagens estão continuamente sendo atualizadas.

O Cadena já foi submetido a experiências com cenários limitados, pode ser utilizado para análise de componentes CIAO e se encontra disponível através do site de seus desenvolvedores. Apesar de serem projetos em fase mais avançada de desenvolvimento (em comparação com os outros dois trabalhos), o Cadena e CIAO/QuO apresentam um esforço de adaptação de um modelo de componentes padrão para que seja usado para aplicações de tempo real.

Existe uma falta de textos na literatura que façam comparação direta entre abordagens de componentes de tempo real. Este artigo buscou preencher essa lacuna, comparando as características e propriedades dos principais trabalhos.

Referências

- [1] Microsoft component object model com. <http://www.microsoft.com/com/>.
- [2] Sun microsystems, java 2 platform enterprise edition. <http://java.sun.com/j2ee/>.
- [3] Corba component model. <http://www.omg.org/technology/documents/formal/components.htm>, June 2001.
- [4] DEMARTINI, C., IOSIF, R., AND SISTO, R. ds핀: A dynamic extension of spin. In *In Theoretical and Applied Aspects of SPIN Model Checking (LNCS 1680)* (September 1999).
- [5] FREDRIKSSON, J., AKERHOLM, M., SANDERSTROM, K., AND DOBRIN, R. Attaining flexible real-time systems by bringing together component technologies and real-time systems theory. In *29th Euromicro Conference* (Turkey, September 2003).
- [6] HATCLIFF, J., DENG, W., DWYER, M. B., JUNG, G., AND RANGANATH, V. Cadena: An integrated development, analysis, and verification environment for component-based systems. In *ICSE 2003* (Oregon, 2003).
- [7] NOLTE, T., MOLLER, A., AND NOLIN, M. Using components to facilitate stochastic schedulability analysis. 24th IEEE Real-Time Systems Symposium (RTSS'2003), Work in Progress Session.
- [8] SCHMIDT, D., LEVINE, D. L., AND MUNGEE, S. The design and performance of real-time object request brokers.
- [9] SCHMIDT, D., LEVINE, D. L., AND MUNGEE, S. Goal. the openccm platform. <http://corbaweb.lifl.fr/OpenCCM/>, 2002.
- [10] TEŠANOVIĆ, A., NYSTRÖM, D., HANSSON, J., AND NORSTRÖM, C. Aspects and components in real-time system development: Towards reconfigurable and reusable software. *Journal of Embedded Computing* (February 2004).
- [11] URTING, D., BERBERS, Y., BAELEN, S. V., HOLVOET, T., VANDEWOUDE, Y., AND RIGOLE, P. A tool for component based design of embedded software. TOOLS Pacific 2002.
- [12] WANG, N., BALASUBRAMANIAN, K., AND GILL, C. Towards a real-time corba component model. In *in OMG Workshop On Embedded & Real-Time Distributed Object Systems* (Washington, D. C., July 2002).
- [13] WANG, N., SCHMIDT, D. C., AND GOKHALE, A. Total quality of service provisioning in middleware and applications. In *Elsevier Journal of Microprocessors and Microsystems* (January 2003). vol.26, number 9-10.
- [14] YAU, S., AND TAWEPONSOMKIAT, C. Component configuration for object-oriented distributed real-time software development. In *ISORC 2000* (2000).
- [15] YAU, S., AND TAWEPONSOMKIAT, C. An approach to object-oriented component configuration for real-time software development. In *ISORC 2002* (2002).
- [16] YAU, S. S., AND XIA, B. An approach to distributed component-based real-time application software development. In *ISORC'98* (Kyoto, Japan, April, 20-22 1998).
- [17] ZINKY, J. A., BAKKEN, D. E., AND SCHANTZ, R. E. Architectural support for quality of service for corba objects. *Theory and Practice of Object Systems* 3, 1 (1997).