

4

Biblioteca de Componentes para Simuladores de Radar

Este capítulo apresenta uma solução para se criar simuladores de diferentes sistemas radar utilizando a tecnologia componentes SCS. Na seção 4.1 apresentamos alguns requisitos inerentes ao sistema. Com base nesse estudo e na experiência prática, desenvolvemos um conjunto de componentes para ser usado com o SCS. Na seção 4.2 é apresentado o histórico de evolução da biblioteca. A divisão lógica do sistema é apresentada na seção 4.3. E, por fim, destacamos algumas observações finais na seção 4.4.

4.1

Motivação e Requisitos

A necessidade de se construir um simulador de radar foi percebida durante a concepção do Radar Saber M60 (04). O simulador foi útil na avaliação das funcionalidades e da viabilidade do projeto do Radar M60 e facilitou a realização de ajustes ao longo de seu desenvolvimento. Com o conhecimento obtido com o simulador foi possível definir características, modelos e modos de funcionamento.

O escopo do simulador e os modelos utilizados foram definidos com o objetivo que este tivesse um comportamento próximo ao do radar real, sendo possível a interação entre componentes reais e componentes simulados. A simulação do processamento de sinais e de dados foi desenvolvida para atender aos requisitos técnicos do radar, e todo o seu comportamento está descrito no simulador. O programa de simulação pode fornecer dados de diversos tipos de cenários com os vários tipos de vetores aéreos. Mesmo após a concepção do produto radar todas as alterações propostas para o radar são validadas antes no simulador para enfim serem utilizadas no Radar Saber M60. Assim, procedimentos que poderiam custar grandes recursos sofrem uma economia devido a sua avaliação anterior no simulador. (04)

Durante o desenvolvimento de um radar de Identificação de Alvos Amigo-Inimigo (14), também conhecido como radar secundário, novamente foi implementado um simulador do sistema que proporcionou economia de recursos, agilidade e flexibilidade no desenvolvimento do produto. Um radar de Identi-

ficção de Alvos Amigo-Inimigo possui um processamento e modo de funcionamento diferenciado de um radar normal, não sendo possível a sua simulação com a biblioteca aqui proposta, no entanto serve de exemplo para a motivação à utilização de simuladores. E previamente à concepção dos radares Saber M200 (04), que será um radar com matriz de antenas e de alcance maior que o M60, e do Sentir M20, que será um radar para detecção de pessoas e tropas, foi percebida a utilidade da implementação de simuladores, motivando, assim, o estudo para a criação de uma biblioteca de *software* que permita a implementação de simuladores de sistemas radar.

Tal biblioteca deve permitir a instanciação de diferentes simulações de sistemas radar por meio da conexão e configuração de módulos, a criação de novos módulos visando alterar ou melhorar a simulação, a interação dos módulos da biblioteca com módulos reais de um radar permitindo testes isolados destes módulos mesmo que outras partes do sistema ainda estejam em desenvolvimento. É desejável que a conexão entre os módulos possa ser feita e desfeita em tempo de execução, permitindo assim que no futuro essa biblioteca seja utilizada para testes em gerenciamento de tarefas, possibilitando o radar modificar seu funcionamento criando e delegando tarefas.

4.2

Histórico de desenvolvimento

Iniciamos nossa pesquisa estudando o simulador desenvolvido no projeto do Radar M60 (04). A princípio o principal objetivo foi recriar o simulador do Radar M60 utilizando a plataforma de componentes de *software* SCS (34).

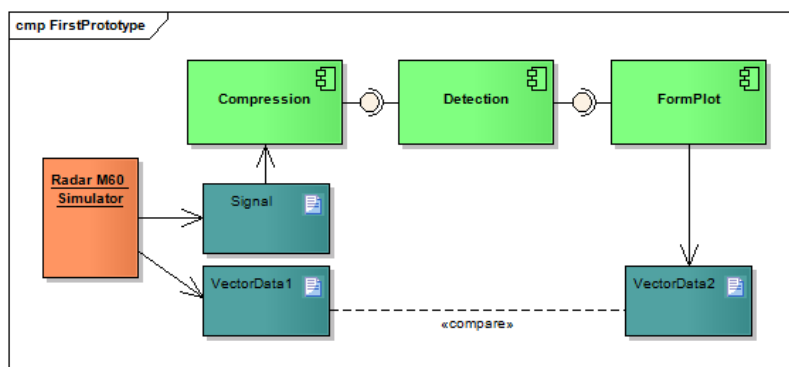


Figura 4.1: Visão do primeiro protótipo implementado

O primeiro protótipo desenvolvido realiza parte do processamento de sinais e processamento de dados, sendo que a camada de processamento de sinais possui os componentes de compressão de sinais e de detecção e a camada de processamento de dados possui os componentes de ordenação dos dados, de cálculo dos parâmetros dos vetores detectados e de rastreamento de

trajetória. A entrada de sinais foi realizada pela leitura de arquivos gerados no simulador do Radar M60 e a saída de dados processados foi comparado com a saída fornecida pelo simulador. Atingido a igualdade de resultados nos dois simuladores, a implementação do primeiro protótipo foi dada como concluída. Na Figura 4.1 é ilustrado o primeiro protótipo.

A partir desse primeiro protótipo ficou comprovada a viabilidade da implementação da biblioteca de utilizando o sistema de componentes de *software* SCS. A próxima etapa foi a implementação completa do simulador do Radar M60, simulando desde a geração de sinais, passando pelo seu processamento e até obtenção de dados de saída do radar.

Tendo o simulador do Radar M60 implementado utilizando a tecnologia de componentes de *software*, o objetivo passou a ser a implementação da biblioteca para simuladores de radar. Para isso esse simulador implementado foi utilizado como ponto de partida, sendo necessário a generalização dos componentes existentes, a criação de novos componentes e a definição das interfaces.

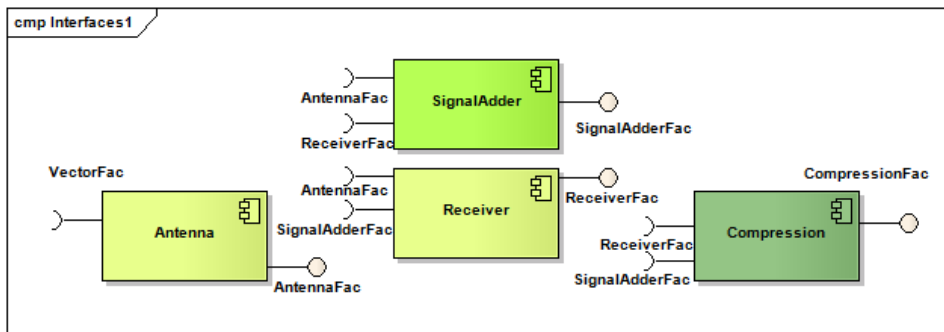


Figura 4.2: Receptáculos para cada faceta dos diferentes componentes

O primeiro desafio foi a definição das interfaces para atender a flexibilidade que a biblioteca se propõe. A primeira abordagem, ilustrada na Figura 4.2, os componentes possuem suas facetas e os componentes que necessitam utilizar os serviços dessas faces possuem receptáculos correspondentes para cada componente.

Existe uma segunda abordagem, ilustrada na Figura 4.3, onde diferentes componentes possuem uma faceta que implementa uma interface comum, assim os componentes que necessitam utilizar os serviços dessa faceta apenas necessitam de um receptáculo, uma vez que as facetas dos componentes que provem o serviço implementam a mesma interface. Nessa abordagem um componente necessita implementar mais de uma interface.

A segunda abordagem foi a escolhida devido a sua maior simplicidade, pois modulariza as interfaces, de implementação da facetas e receptáculos que atendam os requisitos de flexibilidade das conexões entre os componentes que

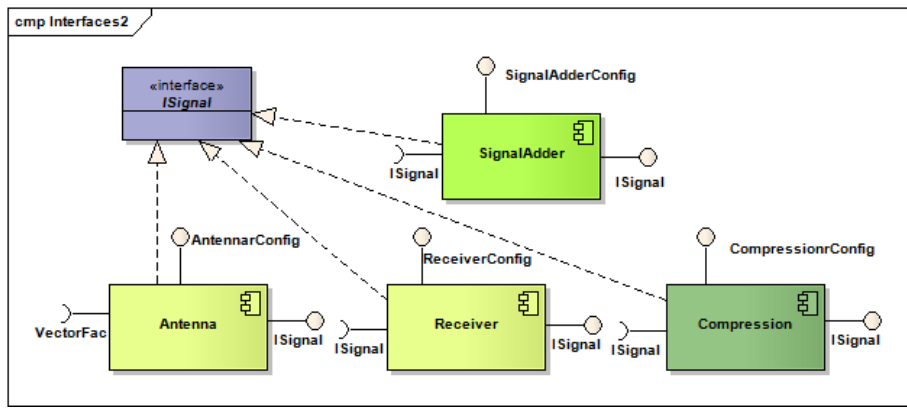


Figura 4.3: Componentes implementam a mesma interface e possuem facetas compatíveis

biblioteca exige. Por exemplo na Figura 4.4 é ilustrado um determinado tipo de processamento de sinais analógico e a posterior digitalização do sinal no componente *receiver*. Para se alterar para uma simulação do mesmo processamento entretanto utilizando um sinal já digitalizado, basta a alternância da ordem dos componentes, antecipando a etapa de digitalização ocorrida no componente *receiver*, ilustrado na Figura 4.5.

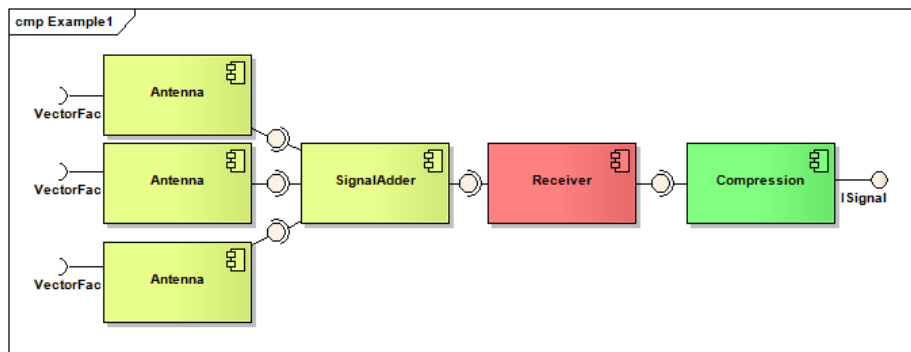


Figura 4.4: Simulação de processamento com digitalização do sinal

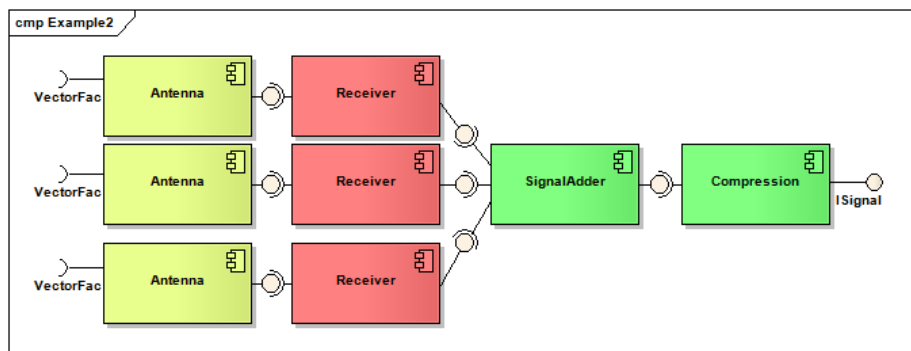


Figura 4.5: Antecipação da etapa de digitalização do sinal

Outra questão interessante foi a tomada de decisão por acumular funções no componente *antena*, tendo como objetivo diminuir o número de componen-

tes, diminuir a complexidade da biblioteca e diminuir o tempo de codificação. Assim, o componente antena acumula funções dos componentes: transmissor, antena de transmissão, canal de propagação da onda e antena de recepção. Podendo ser interpretado como uma composição destes componentes.

Na Figura 4.6 o componente antena é exibido como uma composição dos componentes acumulando suas funcionalidades, mas como a implementação do modelo de componentes utilizado, o SCS, ainda não permite a composição de terceiros para um componente, esta modelagem de composição não foi implementada utilizando técnicas de composição previstas pelo modelo de componentes de *software*.

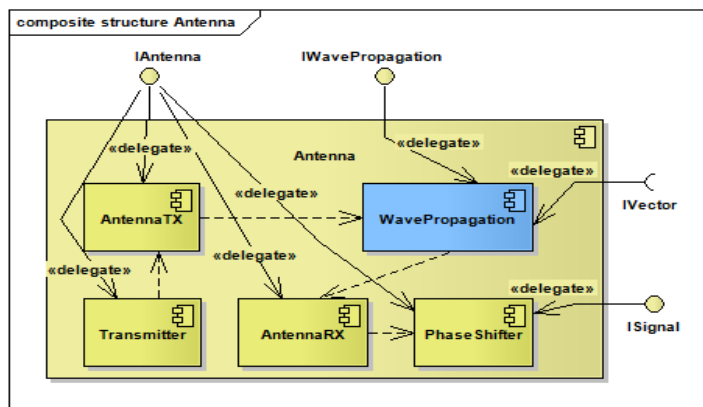


Figura 4.6: Componente antena como uma composição de terceiros

Outra análise interessante é a que diz respeito a sincronização dos componentes. O sentido natural do fluxo de informação nesse sistema é originando-se no transmissor, seguindo pela antena de transmissão, meio de propagação da onda, vetor, e retorna pelo meio de propagação da onda, antena de recepção, receptor, processamento de sinais e por fim processamento de dados.

No entanto podem existir diversos vetores e diversos canais de recepção (um canal de recepção equivale canal de informação gerado pela antena de recepção, pelo receptor e pelo processamento de sinais) que irão gerar sinais independentes que irão compor os sinais de processamento. Tais sinais compostos devem conter sinais de um determinado momento de simulação, sendo necessário um mínimo de coordenação entre os componentes.

Uma solução é a criação de um elemento de controle dos componentes, ilustrado na Figura 4.7 para sincronização do mesmos. Como essa informação espalhada acaba convergindo na etapa de processamento de dados. A solução adotada foi a de inserir esse controle no componente de detecção, ilustrado na Figura 4.8, que se encontra uma etapa antes do processamento de dados. Essa solução foi boa no que tange a simplicidade de programação, no entanto não

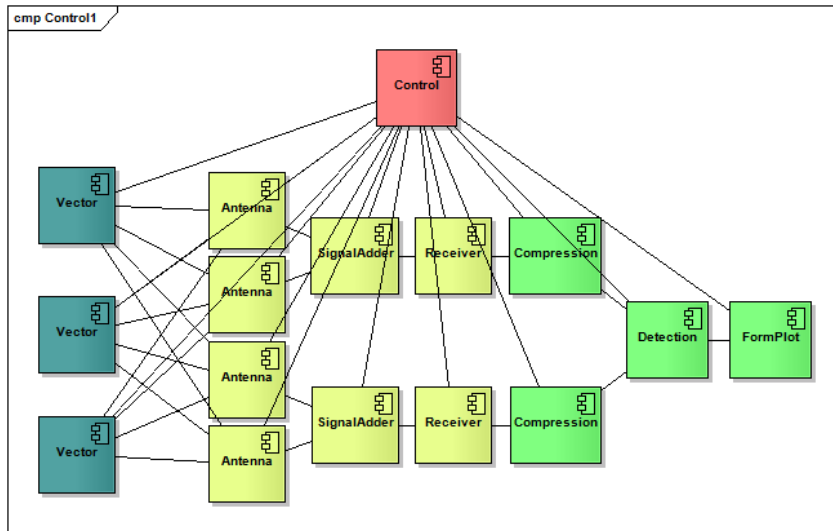


Figura 4.7: Componente de controle para sincronizar os componentes

aproveita as possibilidades de processamento paralelo(observe que não afeta o processamento distribuído) que o sistema de componentes de *software* SCS disponibiliza, pois o controle implementado acaba serializando a geração de sinal nos diversos canais de recepção,

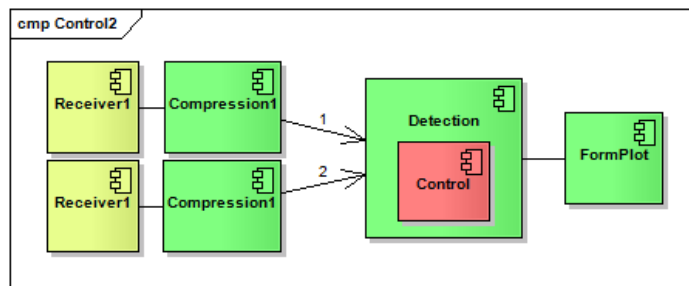


Figura 4.8: Componente de controle interno no componente de detecção

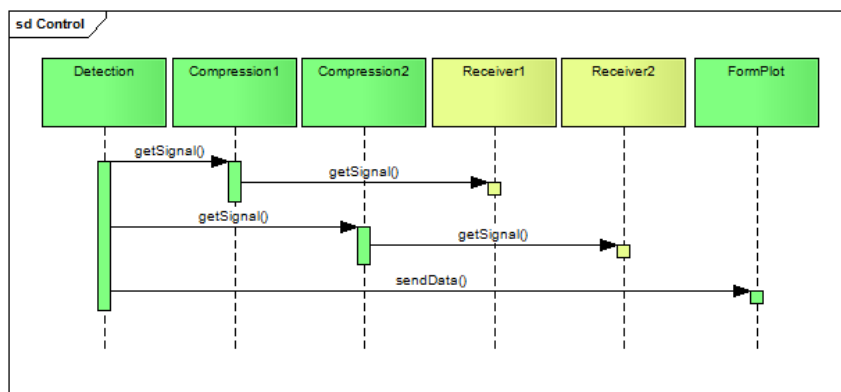


Figura 4.9: Diagrama de seqüência do controle de componentes

Outro ponto interessante é o componente de calculo dos parâmetros dos vetores detectados. Esse componente é responsável por calcular os parâmetros

como distância, azimute, elevação ou velocidade dos vetores detectados. A questão é que um radar pode calcular todos ou um subconjunto desses parâmetros, e existem métodos diferentes para se calcular um mesmo parâmetro. A solução foi criar uma faceta de configuração no componente, onde é possível mapear os canais de recepção nos parâmetros desejados e escolher o método de cálculo desejado.

Uma segunda opção é a criação de um componente específico para cada cálculo de parâmetro e método de cálculo. Sendo que a simples inserção, na simulação, de um componente de cálculo de parâmetro significa a escolha da realização do cálculo deste parâmetro pelo método determinado. A primeira opção foi a implementada devido a sua maior simplicidade.

Na Figura 4.10 é ilustrada a seleção de parâmetros calculados por configuração em comparação com a seleção por conexão de componentes.

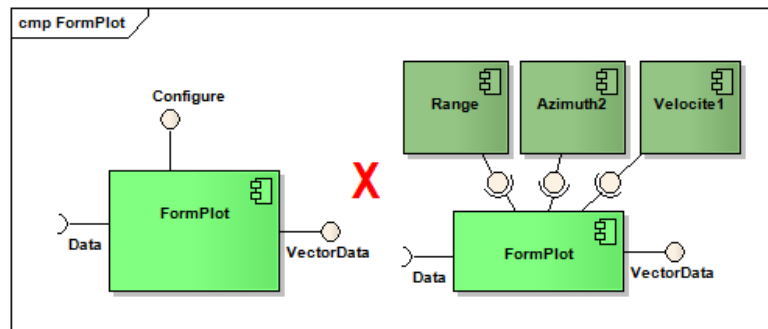


Figura 4.10: Seleção de cálculo de parâmetros: configuração X conexão

Um último ponto interessante foram as tecnologias utilizadas. Inicialmente começamos a implementação utilizando o sistema de componentes SCS sobre o ORBIX (11), que é um *middleware* de comunicação CORBA. No entanto nos deparamos com um problema quando eram carregados diferentes tipos componentes em um mesmo *Container*. Detectado que o problema se encontrava no ORBIX, passamos a utilizar outro *middleware* de comunicação CORBA, o MICO (21), e não mais obtivemos tal problema.

4.3

Divisão lógica da Biblioteca

O estilo proposto para a biblioteca consiste em quatro camadas conforme a Figura 4.11. Estas camadas podem ser diretamente mapeadas nas entidades da Figura 2.2 que mostra uma visão de um sistema radar. Na camada de ambiente foram reunidos os componentes relativos a simulação dos vetores e dos efeitos da propagação eletromagnética no meio. A camada de *hardware* do radar reflete a parte concreta com as entidades referentes a antena, ao receptor e ao motor. A camada de *software* do radar pode ser subdividida em

duas camadas, uma de processamento de sinais e uma de processamento de dados, estas camadas contemplam os componentes abstratos de programação. E por fim, uma camada de suporte para prestar apoio a operações de registro (*log*) e operações de validação, como a análise estatística dos dados processados.

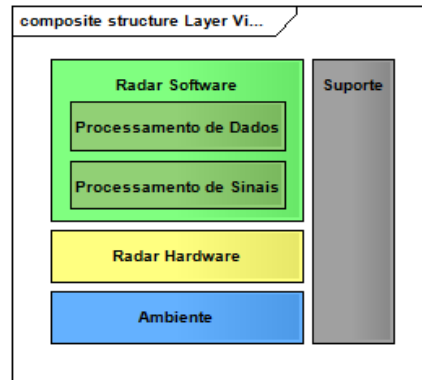


Figura 4.11: Divisão lógica dos componentes em camadas

4.4

Considerações finais

A biblioteca proposta utiliza como base um sistema de componentes que naturalmente atende aos requisitos de modularidade exigidos, o sistema de componentes escolhido permite a sua execução distribuída o que possibilita a interação do simulador com componentes reais de um radar e também permite a conexão em tempo de execução de seus componentes. Para a biblioteca proposta foram implementados diversos componentes que permitem simular diferentes cenários de simulação por meio de suas conexões e configurações.

Ao término deste capítulo é interessante analisar o suporte oferecido por este trabalho para a implantação de um simulador de um sistema radar. A biblioteca proposta oferece uma coleção de componentes que podem ser selecionados para montar um simulador completo de sistema radar. Os componentes podem ser conectados, respeitando suas interfaces, e configurados para agirem conforme a simulação desejada. A biblioteca oferece componentes para registro das operações dos componentes, para validação estatística dos resultados obtidos. Os componentes implementados utilizam algoritmos clássicos da literatura, apresentados no Capítulo 2, no entanto basta a implementação de novos componentes que respeitem as interfaces para se utilizar algoritmos mais aprimorados ou novos algoritmos para testes.