Modelagem Conceitual de Sistemas de Tempo-Real — Um Estudo de Caso —

Maria Luiza d'Almeida Sanchez
UFF Departamento de Engenharia de Telecomunicações
CAA - Pós Graduação em Computação e Automação Aplicada
e-mail: mluiza@caa.uff.br

Bruno Maffeo[1]
UFG Instituto de Matemática e Física
Departamento de Estatística e Informática
e-mail: bmaffeo@dei.ufg.br

PUC-Rio Inf MCC 18/96 junho,1996

Abstract

This paper presents a summary of results concerning a controlled experiment related to employing concepts, tools and techniques for systems modeling to construct the conceptual specification of the real-time control portion of a realistic process-control system. The control subsystem, **COMONEL**, is intended for **CO**ntroling and **MON**itoring **EL**ectrons Beam Litography process, in future use by the Department of Materials Science and Metalurgy (DMSM) of the Pontifical Catholic University of Rio de Janeiro (PUC-Rio).

This work emphasizes an experimental approach to Software Engineering. It follows a model of research inspired on methods of empirical investigation used by traditional engineering disciplines and by the sciences of nature. Specifically aiming at the conceptual modeling of software systems, we elaborated on tools and techniques proposed for the essential requirements specification of software for process-control real-time systems as extensions of those in use by practitionners for developing conventional information systems. This choice reflects a concern related to reducing the costs associated with the transfer to practice of the proposed method. These tools and techniques were tested and refined within the context of a controlled empirical process: the development of a realistic system, COMONEL, with size and intrinsic complexity seemingly adequate to the scope of the experiment. Specifically, the paper focuses:

- the results of Requirements Analysis and Specification restricted to the construction of the Conceptual Model;
- an overview of a deduction technique based on the Essential Model for generating an object-like oriented design, emphasizing the reuse possibilities of that model.

Keywords:
Conceptual Modeling, Real-Time Systems, Essential Model, Desing Deduction Technique

---

[1] Professor visitante licenciado pela PUC-Rio, Departamento de Informática.

Resumo

Este trabalho apresenta um sumário dos resultados referentes a um experimento controlado relativo ao emprego de conceitos, ferramentas e técnicas para modelagem de sistemas visando construir a especificação conceitual da porção de controle em tempo-real de um sistema processo-controle realista. O subsistema de controle, **COMONLIFE**, deve estabelecer o **CO**ntrole e **MON**itoramento de um processo de **LI**tografia por **FE**ixe de Elétrons, a ser usado futuramente pelo Departamento de Ciência dos Materiais e Metalurgia da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio).

O trabalho enfatiza uma abordagem experimental para a Engenharia de Software. Ele segue um modelo de pesquisa inspirado nos métodos de investigação empírica empregados nas disciplinas de engenharia tradicionais e de ciências da natureza. Visando especificamente à modelagem conceitual de sistemas de software, foram desenvolvidas ferramentas e técnicas para especificar os requisitos essenciais de software para sistemas processo-controle de tempo-real baseadas em extensões daquelas utilizadas na prática durante a construção de sistemas de informação convencionais. Essa escolha reflete a intenção de reduzir os custos associados ao processo de transferência da tecnologia relativa ao método proposto. Essas ferramentas e técnicas foram testadas e refinadas no contexto de um trabalho experimental controlado: o desenvolvimento de um sistema realista, COMONLIFE, cuja complexidade intrínseca e porte foram considerados adequados ao escopo do experimento. Especificamente, o trabalho focaliza:

- os resultados da Análise e Especificação de Requisitos restritos à construção do Modelo Conceitual;
- a apresentação de uma técnica de dedução a ser aplicada ao Modelo da Essência visando gerar um design com características de orientação a objetos, enfatizando as possibilidades de reúso desse modelo.

Palavras Chave:
Modelagem Conceitual, Sistemas de Tempo-Real, Modelo da Essência, Técnica de Dedução do Design

Sumário

1. Introdução

A motivação principal da linha de pesquisa na qual este trabalho está inserido relaciona-se ao **desenvolvimento da porção de controle de sistemas processo-controle** (“process-control systems”) **de grande porte, distribuídos e de tempo-real**. Quando comparados a aplicações comerciais convencionais (sistemas de informação), (sub)sistemas de controle geralmente apresentam menor tolerância associada a requisitos de desempenho e a falhas, maior tempo de desenvolvimento e maior dimensionamento da equipe técnica. Seu processo de desenvolvimento deve abordar ampla e profundamente, em nível elevado de abstração, questões como eficácia, eficiência e confiabilidade de sistemas.

Os rígidos requisitos — primordialmente referentes a desempenho e tolerância a falhas — a que deve satisfazer essa classe de sistemas não somente aumentam sua complexidade intrínseca mas também limitam o espaço de soluções de uma implementação aceitável para o sistema processo-controle como um todo e, em particular, para a fração sob a responsabilidade do subsistema digital de controle, no que diz respeito ao hardware e ao software. Esse subsistema de tempo-real possui características técnicas importantes que influenciam seu processo de desenvolvimento. São elas:

- ♦ o processamento deve ter elevada disponibilidade confiável e deve satisfazer continuamente requisitos de tempo, não somente relacionados a tempo de resposta mas também a sincronismo com eventos programados ou aleatórios que ocorrem no ambiente externo (processo controlado e operador);
- ♦ a execução ocorre em ambiente externo imperfeito;
- ♦ a segurança na presença de falhas de software e hardware deve ser assegurada.

Na Engenharia de Software, o domínio da complexidade do processo de desenvolvimento de um sistema é obtido através de uma sistemática de segmentação do processo de desenvolvimento em etapas com resultados bem definidos. Durante cada etapa o domínio da complexidade é obtido através da segmentação do sistema em subsistemas, de acordo com a abordagem característica dessa etapa. Tecnicamente, a operação “caixa-preta” desse conjunto de componentes — subsistemas — que trabalham cooperativamente para um propósito comum pode ser descrita por uma relação matemática envolvendo entradas (presentes e passadas), saídas e tempo. Portanto, o processo de segmentação deve basear-se na identificação de partes coesas dessa relação, independentes ou interagindo fracamente (baixo acoplamento). Infelizmente, essa identificação não é trivial e, no caso de sistemas complexos, dificilmente é bem-sucedida quando o processo utiliza apenas uma perspectiva de modelagem — e isso ocorre qualquer que seja a etapa considerada —. É indispensável o uso de diferentes perspectivas não-ortogonais para entender-se o problema a ser tratado — o processo controlado — bem como para gerar-se uma proposta de solução — o (sub)sistema de controle —.

Está cada vez mais difundida na área de Informática a convicção de que o processo de desenvolvimento de software só terá a capacidade de gerar produtos eficientes de alta confiabilidade e eficácia comprovada na medida que esse processo utilize conceitos, linguagens, ferramentas e técnicas de modelagem que abranjam também suas etapas de mais alto nível de abstração em relação à tecnologia empregada na implementação do sistema:

- ♦ **etapa de análise**: modelagem conceitual, visando à eficácia, foco principal do presente trabalho;
- ♦ **etapa de design**: modelagem da implementação, visando à eficiência/confiabilidade.

Essa convicção é mais vigorosa quando a classe do software a ser construído incorpora caracterísiticas de sistemas de tempo-real. Nesse caso, pesquisas recentes [Ward85, Maffeo91, Maffeo92, Barbosa92, Barbosa95, Clemente92, Sanchez96, Sanchez95a, Sanchez95b, Sanchez95c] em Engenharia de Software dirigiram-se no sentido de definir um conjunto de instrumentos — conceitos, linguagens, ferramentas e técnicas de modelagem — que possam constituir apoio eficaz ao processo de desenvolvimento nessas etapas que antecedem logicamente (não necessariamente cronologicamente) a etapa de codificação.

Como em princípio deve ocorrer em relação a qualquer sistema artificial construído para atender a propósitos previamente estabelecidos, a implementação de um subsistema de controle deve estabelecer um comportamento desejado. Esse comportamento é especificado nos requisitos, os quais devem conter informações suficientes para que seja possível distinguir entre os comportamentos desejados e os não desejados de forma completa e não ambígua. Linguagens de especificação que não exponham corretamente as distinções de comportamentos relevantes são inadequadas [Jaffe91]. Ambigüidade ou imprecisão nos requisitos afeta a capacidade de determinar a correção de sistemas.

Por outro lado, a correção de sistemas possui dois aspectos:

- o sistema implementado deve satisfazer os requisitos especificados;
- os requisitos especificados devem corresponder ao comportamento desejado e satisfazer as restrições impostas pelo ambiente externo ao sistema.

Nessas condições, os modelos a serem empregados no processo de desenvolvimento não podem focalizar apenas a validade da implementação em relação a uma especificação de requisitos. Para garantir eficácia, segurança ("safety"), disponibilidade, confiabilidade e desempenho, os modelos empregados em todas as fases do desenvolvimento, em especial na formulação do enunciado do problema a ser resolvido pelo subsistema de controle, devem:

- ser coerentes com o comportamento (normal e anormal) do ambiente externo (processo controlado e operador) sob todas as circunstâncias relevantes;
- explicitar as restrições impostas pelo processo controlado, levando em conta o fato de que a introdução de um sistema baseado em computadores pode causar mudanças nesse comportamento.

Caracteristicamente, a pesquisa em Engenharia de Software tem atribuído ênfase insuficiente à avaliação prática das hipóteses, ferramentas e técnicas de trabalho por ela gerados. Isso decorre certamente da complexidade e dos grandes custos associados a projetos de pesquisa orientados para esse tipo de experimentação e provavelmente a uma percepção ainda mal definida em relação a qual deva ser o método a ser empregado em uma atividade experimental nessa área. Em particular, percebe-se que as analogias possíveis com a prática experimental associada a outras disciplinas mais antigas — tais como a Física e a Química — ainda não foram completamente exploradas. De todo modo, qualquer que seja a origem desse afastamento entre a pesquisa e a prática [Glass94], antecipa-se que, permanecendo, ele pode provocar um retardamento significativo no progresso do conhecimento técnico-científico relativo ao processo de desenvolvimento de software.

Visando contribuir para a diminuição do afastamento mencionado no parágrafo precedente, a linha de pesquisa da qual resulta o presente trabalho tem buscado privilegiar a vertente experimental da Engenharia de Software. O método empregado baseia-se no entendimento do que tem sido o método de pesquisa aplicada empregado nas engenharias tradicionais, no qual, dado um problema, novo ou antigo, empreende-se:

- a análise das soluções existentes, nela incluindo-se a busca de analogias para o problema e para as soluções;
- a proposta de melhores soluções;
- a construção ou o desenvolvimento referente à proposta;
- repetitivamente, a análise comparativa qualitativa/quantitativa dos resultados obtidos, gerados pelo processo de construção ou desenvolvimento, até que não seja mais possível/desejável melhorar a proposta.

Certamente, empregar um método baseado em experimentação controlada implica aumentar o custo financeiro da pesquisa em Engenharia de Software, bem como o prazo para obtenção de resultados válidos e a dificuldade para divulgá-los. Não empregá-lo induz a uma pesquisa cujo compromisso primordial é com a lógica da formulação teórica e não com a realidade da prática, afastando-a dos modelos de pesquisa (sobretudo em engenharia) onde a validação e o refinamento da teoria fundamenta-se na construção ou no desenvolvimento de artefatos com características realistas.

Dentro dessa filosofia e partindo-se de modelos testados na prática da construção de sistemas convencionais comerciais, foram elaboradas propostas para o processo de desenvolvimento da porção de controle de sistemas processo-controle em tempo-real. Essas propostas foram testadas e refinadas através de um processo de experimentação controlada — o desenvolvimento de um sistema realista dotado de complexidade e porte necessários à abrangência da experimentação —. Visando reduzir os custos de transferência da tecnologia desenvolvida, as propostas introduzidas baseiam-se em ferramentas conceituais e técnicas de modelagem próximas àquelas já empregadas com sucesso na prática.

Foi nesse contexto que se estabeleceu a proposta de desenvolver o (sub)sistema de controle/monitoramento **COMONLIFE** — **CO**ntrole e **MON**itoramento de **LI**tografia por **FE**ixe de Elétrons —. O sistema processo-controle correspondente é parte de um projeto desenvolvido em conjunto com o Departamento de Ciência dos Materiais e Metalurgia e o Departamento de Física da PUC-Rio.

A **etapa de análise** do processo de desenvolvimento focaliza primordialmente a modelagem conceitual [ISO695] do (sub)sistema de controle a ser construído. Nessa etapa, empregam-se as ferramentas conceituais, técnicas de modelagem e conceitos de modelagem pertinentes à elaboração do **Modelo da Essência** [Maffeo92, Ward85]**,** que corresponde a uma evolução da Análise Estruturada tradicional [DeMarco78, Gane79]. Esse modelo define o problema a ser resolvido pelo sistema e especificando uma solução abstrata (**requisitos essenciais**) totalmente despida de viés de implementação. Ao adotar uma abordagem de múltiplas perspectivas sobre o problema a ser resolvido pelo sistema e empregar um conjunto consistente de ferramentas conceituais e técnicas para modelagem "caixa-preta" do sistema, a modelagem da essência, conforme descrito em [Maffeo92], aprimora significativamente o processo de geração do enunciado desse problema. Analogamente, a estratégia de modelagem paralela e simultânea de atividades e informações [Ward85, Maffeo92] facilita a elaboração de uma solução abstrata correta e consistente com o problema a ser resolvido. Versões preliminares dessas ferramentas, técnicas e conceitos já foram empregadas com sucesso em [Barbosa92, Clemente92, Sanchez95b]. Essa modelagem conceitual utiliza linguagens de representação definidas rigorosamente que permitem a distinção entre comportamentos desejados e não desejados.

Muitas dessas ferramentas/técnicas de modelagem já possuem formalização matemática [Richter92]. As definições de sintaxe e semântica [Ward85, Maffeo92] propostas para o Modelo do Comportamento permitem afirmar que esse é um modelo rigoroso e totalmente formalizável matematicamente. Visando melhorar seu poder de comunicação, o modelo utiliza anotações informais compreendendo descritores sugestivos associados a elementos formais de modelagem, cuja interpretação ligada ao mundo "real" restringe-se à definição dos dados presente no Dicionário de Dados. Essa definição será sempre, qualquer que seja o instrumento de modelagem usado, informal.

A modelagem conceitual de um sistema de tempo-real para controle/monitoramento de processos constitui desafio bastante atual para a Engenharia de Software. Empregá-la neste trabalho contribuiu para aumentar o entendimento do processo a ser controlado, aí incluindo possíveis comportamentos anormais, bem como para assegurar de modo preciso a eficácia da solução informatizada proposta. Para isso, o experimento realizado englobou a construção do Modelo da Essência para o **COMONLIFE** [Sanchez95b] através do qual pretendeu-se evidenciar a pertinência da estrutura adotada para a modelagem da essência.

Dessa forma, visou-se:

- estabelecer as condições para realizar uma modelagem conceitual que especifique requisitos de forma a garantir a confiabilidade e a segurança do sistema a ser construído;
- obter uma considerável redução de erros no produto final e de custos nas fases de design, implementação, implantação e manutenção do sistema.

2. O Processo de Desenvolvimento

A estrutura de modelagem associada ao processo de desenvolvimento completo, preconizada por [Ward85, Maffeo92], inclui três modelos, com graus de abstração diferentes, que detalham toda a concepção e implementação do sistema. Existe uma relação de precedência lógica entre esses modelos que, entretanto, não define necessariamente a ordem cronológica mais adequada para sua construção. São eles:

- **Modelo da Essência** (ME), representando o resultado da elicitação, análise e **especificação dos requisitos essenciais** (isto é, independentes de qualquer alternativa de implementação) do sistema em questão. É o objeto principal deste trabalho.
- **Modelo da Implementação** (MI), sucessor lógico do ME incorporando os requisitos essenciais ao design do sistema, caracterizado sumariamente neste trabalho;
- **Modelo da Automação** (MA), sucessor lógico do MI incorporando o design de alternativa de implementação selecionada aos recursos tecnológicos de nível de abstração mais baixo (linguagem de programação, processadores, ...). Não é tratado neste texto.

Os produtos gerados nas diferentes etapas do processo de desenvolvimento estão indicados na figura a, onde o símbolo "+" indica alternativa e cada modelo engloba o conteúdo do modelo que o precede logicamente.

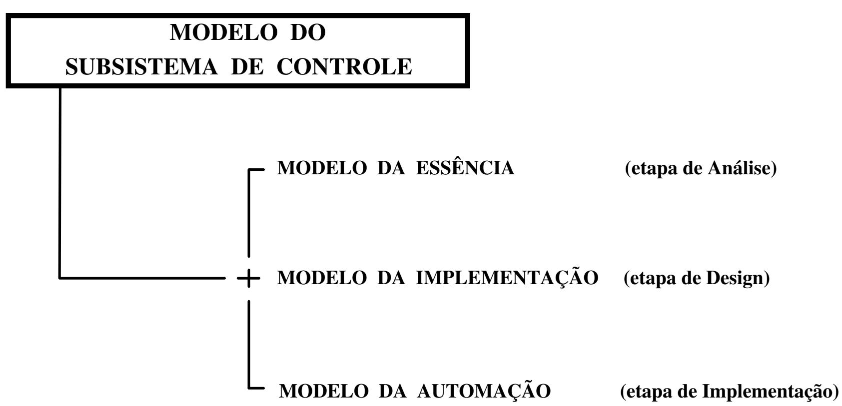

Figura A- Visão Estrutural do Processo de Modelagem

3. O Modelo da Essência

O Modelo da Essência utiliza ferramentas conceituais e técnicas de modelagem rigorosas, primordialmente gráficas, com sintaxe e semântica bem definidas, que facilitam significativamente a comunicação entre pessoas com níveis de conhecimento distintos em relação ao uso da tecnologia de computação digital. A modelagem esquemática das atividades do sistema, através de gráficos legendados realçando o papel dos elementos funcionais ativos (funções), baseia-se numa notação que amplia e aperfeiçoa aquela usada para construir Diagramas de Fluxo de Dados. Essa notação, cuja semântica está parcialmente formalizada em [Richter92], foi denominada ESML (Extended Systems Modeling Language) [Bruyn88, Maffeo91]. Sua eficácia fica refletida na clareza e precisão da representação obtida, como pôde ser constatado em [Ackerman91, Clemente92, Barbosa92, Sanchez95b] onde são apresentados os resultados de experimentação controlada relativa a seu emprego na modelagem de sistemas de tempo-real. A modelagem esquemática da informação armazenada no sistema baseia-se na notação dos bem conhecidos Diagramas de Entidades e Relacionamentos [Chen76].

O domínio da complexidade de um problema, de modo análogo à concepção de um sistema, exige necessariamente algum esforço de segmentação/abstração. Mas, como em todo projeto de engenharia, esse esforço deve obedecer a critérios bem estabelecidos que não deixem dúvidas sobre o caminho a seguir.

A técnica aqui apresentada tem como primeiro ponto de segmentação/abstração o critério de "Essência versus Implementação" [McMenamin84, Ward85, Maffeo92]. A modelagem da essência abstrai a forma de implementar o sistema — solução concreta de um problema computacional —. Para representar adequadamente a solução abstrata, novamente utiliza-se o processo de segmentação/abstração, agora baseado em um critério de "pertinência ao sistema". Esse critério segmenta o Modelo da Essência em dois submodelos, o Modelo do Contexto e o Modelo do Comportamento, que representam, respectivamente, a definição das necessidades do ambiente externo — o enunciado do problema computacional — e uma solução possível para esse enunciado, abstraída de referências a formas específicas de implementação.

Cada um desses dois modelos é composto de submodelos complementares, construídos a partir de enfoques diferentes sobre o ambiente externo (modelagem do contexo) e sobre o sistema (modelagem do comportamento). Esses submodelos são interdependentes e necessidades do ambiente externo, registradas no Modelo do Contexto, podem vir a ser descobertas durante a construção do Modelo do Comportamento. Isso determina alterações no Modelo do Contexto, visando manter a consistência global do Modelo da Essência.

Enfoques distintos e complementares sobre o contexto do sistema (Descrição Sumária, Descrição de Objetivos, Descrição das Operações, Lista de Eventos Externos, Esquema Transacional, Esquema Semântico, Esquema da Dinâmica e Dicionário de Dados) permitem uma visão abrangente sobre o domínio do problema a ser tratado e a identificação precisa das necessidades do ambiente externo a serem atendidas pelo sistema. A redundância controlada introduzida por esses enfoques, complementada por indispensável disciplina de modelagem, permitem assegurar a consistência do enunciado do problema.

Analogamente, enfoques distintos e complementares sobre o estado (Esquema e Detalhe da Informação) e o comportamento (Esquema e Detalhe das Atividades) do sistema permitem modelar rigorosamente uma solução abstrata para o problema enunciado.

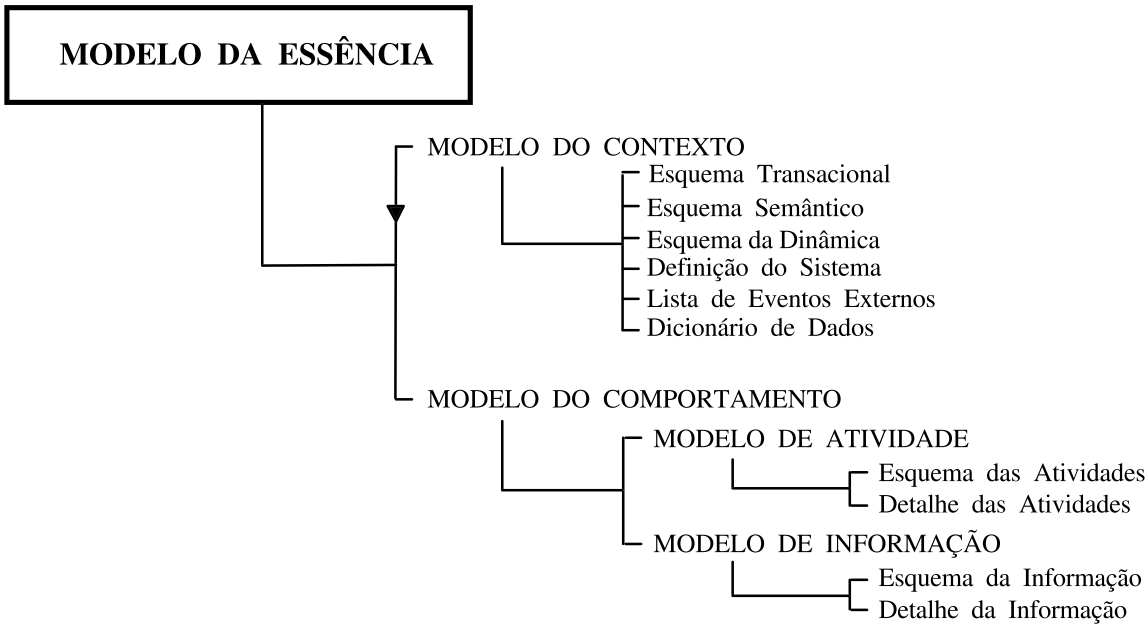

Figura B- Visão Estrutural da Modelagem da Essência

Assim, o **Modelo da Essência** [Maffeo90, Ritto91a, Maffeo92, Clemente92, Sanchez95b] apresenta a seguinte estrutura, representada esquematicamente na figura b (a seta indica precedência lógica):

Modelo do Contexto, composto de:

- **Definição de Sistema**, onde são descritos os objetivos (a nível estratégico, tático e operacional) e a operação “caixa-preta” do sistema. Para essa descrição, emprega-se Linguagem Natural, com estrutura e vocabulário controlado;
- **Lista de Eventos Externos**, identificando os eventos que ocorrem no ambiente externo ao sistema de controle/monitoramento — eventos externos — e que suscitam reações planejadas por parte desse sistema. Essa lista constitui um instrumento básico de modelagem das necessidades do ambiente externo a serem atendidas pelo sistema. Para elaborá-la, emprega-se Linguagem Natural, com estrutura (descritores numerados seqüencialmente e submetidos a classificação) e vocabulário controlado;
- **Esquema Transacional**, que utiliza a ESML (Extended Systems Modeling Language), modelando a interface do sistema com as entidades externas com as quais interage diretamente;
- **Esquema Semântico** [Maffeo90], que modela cada transação presente no Esquema Transacional em termos de um Diagrama de Entidades e Relacionamentos (DER) [Chen76],contendo hierarquias semânticas do tipo “é-um” e “é-composto”, explicitando a natureza e a composição da interface — constituída por fluxos e depósitos externos — entre o (sub)sistema de controle e o ambiente externo;
- **Esquema da Dinâmica** [Clemente92, Sanchez95b], que utiliza a linguagem gráfica dos Diagramas de Estados e Transições ou das Redes de Petri Compactas [Richter92], modelando a dinâmica dos eventos externos de interesse para o sistema.

Modelo do Comportamento, composto de:

- **Esquema da Memória Essencial**, onde o instrumento de modelagem é o DER incluindo toda a parte esquemática da modelagem conceitual dos elementos funcionais passivos do sistema (Tipos de Entidade, seus Atributos e Tipos de Relacionamento entre entidades);

- **Esquema das Atividades Essenciais**, que utiliza a ESML, modelando todas as atividades essenciais em um único diagrama de rede, não hierárquico;
- **Organização Hierárquica das Atividades**, que utiliza a ESML, modelando uma hierarquia de atividades até o nível das atividades primitivas, estas representando as funções do sistema que reagem a ocorrências de eventos externos;
- **Diagramas de Estados e Transições**, um para cada atividade de controle primitiva, constituindo sua especificação formal;
- **Listas de Pré- e Pós- Condições**, uma para cada atividade operacional primitiva (isto é, uma atividade operacional que não tem decomposição funcional gráfica a ela associada), constituindo sua especificação formal.

A abordagem empregada na modelagem do comportamento é "outside-in": o Esquema de Atividades Essenciais (uma porção "in") é parcialmente deduzido a partir da Lista de Eventos Externos (uma porção "outside") e do Esquema da Dinâmica (uma porção "outside") construídos durante a modelagem do contexto. A Organização Hierárquica das Atividades constitui a representação "top-down" dessa modelagem funcional. O Esquema da Memória Essencial (uma porção "in") é parcialmente deduzido a partir do Esquema Semântico (uma porção "outside") e do Esquema Transacional (uma porção "outside") construídos durante a modelagem do contexto.

Complementando os dois modelos, constrói-se um **Dicionário de Dados** que define a informação presente na parte esquemática desses modelos.

Esses submodelos são baseados em perspectivas diferentes sobre o ambiente externo e sobre o sistema. Isso introduz uma redundância na representação global, controlada por imposição da consistência que deve garantir a precisão e não ambigüidade do modelo resultante. A verificação da consistência interna do modelo é facilitada pela construção de uma **Tabela de Verificação de Consistência** [Maffeo92] que relaciona as entradas, reações e respostas do sistema, associadas a cada evento externo.

Neste trabalho não serão apresentados todos os detalhes referentes ao Modelo da Essência, que podem ser encontrados, por exemplo, em [Ward85, Maffeo92].

4. O Modelo da Implementação

O Modelo da Implementação, apenas delineado neste trabalho, consiste em construir a planta que agrega características não-essenciais do sistema aos requisitos essenciais definidos no Modelo da Essência. Esse modelo resulta de uma reorganização e expansão do Modelo da Essência, adicionando-se a este:

- atividades e informações que direcionem o uso de tecnologias de implementação específicas
- especificações procedimentais das ações do sistema.

Dessa forma são completados os requisitos necessários ao atendimento das necessidades do usuário.

O Modelo da Implementação é composto de três porções distintas:

- a especificação dos **Requisitos de Concepção de Sistema**, que agrega requisitos não-essenciais relacionados primordialmente a desempenho e tolerância a falhas. Essa definição impõe restrições à tecnologia a ser empregada na implementação e é precondição para a escolha das arquiteturas de hardware e software básico;
- a especificação da **Interface Usuário-Máquina** (IUM), que determina as características interativas do sistema e sua forma de operação. Implica selecionar e/ou construir os equipamentos de interface com o(s)

operador(es) e/ou com o processo e o software a ser incorporado ao sistema para estabelecer sua interatividade;

- o **Design do Sistema,** que o segmenta em subsistemas obedecendo a um padrão de cooperação para atender aos requisitos essenciais e não essenciais, aqui incluídos os requisitos para a comunicação com operadores.

A estratégia empregada divide o processo de design do sistema em quatro etapas:

1. construção do **Projeto Básico**, que evidencia os **subsistemas básicos** (SB)**,** suas **portas** e **conexões**, bem como a eventual necessidade de concorrência e sincronismo entre os serviços alocados a cada SB;
2. **Especificação Detalhada** de cada **subsistema básico**, que permite seu desenvolvimento autônomo e facilita sua recuperação para reúso;
3. **Design** de cada **subsistema básico** como unidade independente de desenvolvimento;
4. **Configuração do Sistema**, determinando-se a **configuração de software** (definição das portas e conexões associadas aos SBs**)** e a **configuração de hardware** (mapeamento dos SBs e seus processos no hardware).

A figura c representa esquematicamente a estrutura proposta para o Modelo da Implementação.

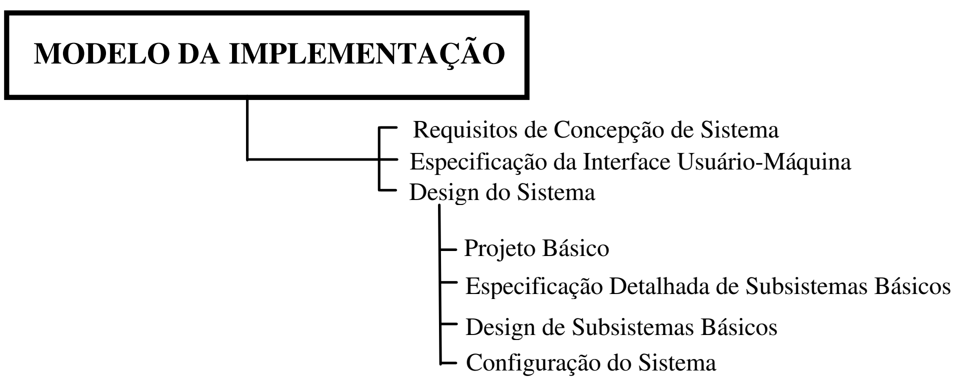

Figura C - Visão Estrutural da Modelagem da Implementação

O **Projeto Básico**, resultado de uma ação de síntese aplicada sobre o Modelo da Essência, contém a definição dos **subsistemas básicos**, que incorpora a especificação dos serviços prestados pelo sistema associados a cada um desses subsistemas. Um **serviço** é um grupo de funções inter-relacionadas que compartilham propósitos comuns. Esse grupo constitui o análogo ao conjunto de operações de uma classe numa abordagem orientada a objetos pois, em particular, gerenciam uma estrutura de dados originária da estrutura de informação encapsulada pela classe da qual deriva o subsistema básico. O conceito de Serviço está fortemente relacionado ao objetivo de reúso e realça o processo de encapsulamento que agrega as atividades do sistema que reagem diretamente à ocorrência de eventos externos.

Subsistemas básicos, definidos a partir de critérios de síntese aplicados ao Modelo do Comportamento e baseados no Encapsulamento de Informação e operações (encapsulam Tipos de Entidade e atividades essenciais), englobam um conjunto resumido e bem definido de serviços passível de reúso em outros sistemas. Serviços prestados por um subsistema básico podem ser concorrentes e, conseqüentemente, um subsistema básico engloba dois ou mais processos. Toda a cooperação (comunicação e sincronismo) existente entre subsistemas básicos se dá através de suas **portas** pelo envio/recebimento de mensagens.

Os **subsistemas básicos** são **unidades de implementação** e **unidades de reúso**. Unidades de implementação que:

- ♦ são codificáveis, compiláveis e testáveis independentemente;
- ♦ prestam-se ao design, gerenciamento, produção e manutenção de grandes sistemas;
- ♦ encapsulam informação e operações de modo a garantir a inexistência de áreas de dados compartilhadas.

Cada subsistema básico é uma unidade de código fonte desenvolvida e mantida por uma equipe pequena e, por definição, é indivisível quando reusado.

5. Conclusão

O trabalho de pesquisa e desenvolvimento que fundamentou o presente documento envolveu:

- ♦ a definição de um método integrado de modelagem, da essência e da implementação, de sistemas de tempo-real para controle de processos
- ♦ e uma experimentação controlada, baseada no desenvolvimento de um sistema realista, visando à validação das características básicas do método proposto.

Neste texto, focalizou-se predominantemente a modelagem conceitual conduzida durante o experimento controlado realizado e as conclusões referentes a essa parte são apresentadas no que se segue.

Construir o Modelo da Essência completo do COMONLIFE visou demonstrar a adequação do Modelo da Essência ao objetivo de elicitar, analisar, especificar, documentar e comunicar os requisitos funcionais e informacionais do subsistema computacional responsável pelo controle/monitoramento. Entretanto, tal demonstração só pode ser evidenciada dispondo-se da versão integral do ME [Sanchez95b]. Trata-se aqui de um ônus a ser incorrido por esse tipo de pesquisa em Engenharia de Software: a demonstração convincente da validade da utilização de qualquer método de desenvolvimento de software em projetos de sistemas complexos que realmente serão colocados em produção só pode ser realizada através de documentação maciça, pouco compatível com os meios tradicionais de divulgação técnico-científica.

O emprego das ferramentas conceituais, técnicas e estrutura de modelagem propostas em [Ward85, Maffeo92] na construção do **Modelo da Essência do COMONLIFE** mostrou que:

- ♦ Ao enunciar o problema a ser resolvido por esse subsistema de controle/monitoramento, no Modelo do Contexto:
    - Na **Definição do Sistema**, o primeiro aspecto a ser tratado é a **Descrição Sumária** do sistema, que dve apresentar uma visão geral concisa dos elementos que constituem o ambiente externo (equipamentos do processo a ser controlado, sensores e atuadores) e das tarefas principais do subsistema de controle/monitoramento. Sua elaboração permitiu a compreensão preliminar do funcionamento básico do processo, o aprendizado da terminologia específica a ele associada e um melhor entendimento das necessidades do ambiente externo.
    - Na **Definição do Sistema**, o segundo aspecto a ser tratado é a **Descrição dos Objetivos** do sistema. O enunciado explícito e, sempre que possível, quantitativo desses objetivos (estratégicos, táticos e operacionais) visa apoiar a análise de custo x benefício x importância do sistema a ser construído, primeiro passo no sentido da tomada da decisão quanto a realmente desenvolver um sistema.
    - Na **Definição do Sistema**, o terceiro aspecto a ser tratado é a **Descrição de Operações** do sistema. Essa descrição, na forma de texto estruturado expresso em linguagem natural empregando vocabulário controlado e facilmente legível, evidenciou rigorosamente, sem antecipar soluções específicas, o problema a ser resolvido pelo subsistema de controle/monitoramento. Essa

evidência é obtida focalizando-se a relação "estímulo x resposta" que caracteriza a interface entre o subsistema de controle/monitoramento e os elementos do ambiente externo com os quais esse subsistema interage diretamente. Trata-se de um instrumento de modelagem fundamental para o processo de validação pois o enunciado do problema a ser resolvido por um sistema deve ser validado pelo cliente/usuário. Essa linguagem de representação facilita a comunicação ao situar-se em nível intermediário de formalização e não empregar conceitos relacionados à técnica de processamento de informação.

- A construção de uma **Lista de Eventos Externos** (LEE) permite a modelagem, sob a forma de eventos, das necessidades do ambiente externo a serem atendidas pelo (sub)sistema de controle/monitoramento. Também constituindo um texto estruturado expresso em linguagem natural empregando vocabulário controlado e facilmente legível, a **LEE** evidenciou rigorosa e completamente essas necessidades de modo a proporcionar outra perspectiva de validação, pelo cliente/usuário, do enunciado do problema a ser resolvido por esse (sub)sistema. Essa perspectiva é obtida focalizando-se a funcionalidade do processo a ser controlado e permite enfatizar, mais concisamente do que a **Descrição de Operações**, o comportamento do ambiente externo. É importante ressaltar que todas as ocorrências relevantes foram explicitamente levadas em conta: tanto aquelas associadas a reações "normais" do ambiente externo quanto as que correspondem a reações "anormais" desse ambiente que impliquem riscos ou mau funcionamento. A **LEE** pode também ser usada como base de um planejamento orientado para a adoção de uma Estratégia gerencial de Desenvolvimento Incremental [Sanchez94, Sanchez93b]. A **LEE** reflete concisamente a funcionalidade do processo a ser atendida pelo subsistema de controle/monitoramento e, portanto, subconjuntos dessa lista podem definir as partes dessa funcionalidade que cada versão incremental desse (sub)sistema deve atender. A **LEE** também constituiu um guia importante para a construção do plano de testes do (sub)sistema a ser implementado.
- O **Esquema Transacional** permitiu a visão integrada das transações entre o (sub)sistema de controle/monitoramento e os elementos relevantes do ambiente externo — seja aqueles que caracterizam o processo a ser controlado seja os dispositivos sensores/atuadores seja o operador — pois evidencia os estímulos e respostas que definem a interface que conecta o artefato de controle a esses elementos e define a fronteira conceitual do artefato. Trata-se de ferramenta útil para efetuar a validação de consistência dos diversos submodelos.
- O **Esquema Semântico** permitiu visualizar graficamente, para cada transação específica, a relação entre estímulos e respostas. Essa visualização facilita a leitura e complementa a visão textual proporcionada pela **Descrição de Operações**. Trata-se de ferramenta poderosa pois auxilia o modelador a determinar se os estímulos considerados são ou não suficientes para dar origem à resposta esperada, que pode ser externa ao sistema (por exemplo, a emissão de um relatório ou de uma saída de controle) ou interna (por exemplo, a internalização de uma informação ou o início do processamento de uma atividade de controle). O **Esquema Semântico** apóia a validação da **Lista de Eventos Externos** e dá subsídios para a construção da Memória Essencial representada no Modelo do Comportamento. A construção do Esquema Semântico permite evidenciar claramente, para cada transação, a relação entrada x saída. Essa explicitação é gerada abstraindo-se o conhecimento específico dos elementos funcionais ativos e passivos que comporão a solução abstrata representada no Modelo do Comportamento. Dessa forma, aprimora-se a compreensão do modelador relativamente às operações "caixa-preta" a serem executadas pelo (sub)sistema de controle/monitoramento, sem haver a necessidade de expressar, juntamente com a especificação da interface entre esse artefato de controle/monitoramento e o ambiente externo, a estrutura e a dinâmica da solução abstrata, características internas ao artefato. Essa compreensão facilitou muito a construção dos Esquemas de Atividades Essenciais e da Memória Essencial, componentes da

solução abstrata de um problema para cujo enunciado preciso o Esquema Semântico contribui significativamente.

- O **Esquema da Dinâmica** permitiu visualizar a interdependência dos eventos externos e suas relações de causa e efeito. Abstrai completamente a existência do subsistema de controle/monitoramento privilegiando a representação do processo, o que permite ao modelador um melhor entendimento do comportamento do processo a ser controlado.

♦ Ao apresentar uma proposta de solução abstrata, no **Modelo do Comportamento**:

- A construção do **Esquema das Atividades Essenciais** permitiu uma reflexão focalizada sobre uma solução abstrata para o problema, privilegiando a representação dos elementos funcionais ativos do COMONLIFE. Esse esquema modela as atividades internas ao sistema que devem reagir à ocorrência de cada evento externo e, portanto, especifica requisitos para estabelecer-se a relação “estímulo x resposta” presente no Modelo do Contexto. O particionamento do conjunto dessas atividades em dois subconjuntos interagentes disjuntos, um contendo atividades operacionais e o outro atividades de controle, aliado ao emprego de Diagramas de Estado e Transição para especificar as atividades de controle, permitiu detalhar a representação da dinâmica do (sub)sistema de controle e, em particular, a concorrência entre suas atividades operacionais.
- A construção do **Esquema da Memória Essencial** permitiu uma reflexão focalizada sobre a informação a ser internalizada e armazenada, necessária à construção da solução abstrata. Privilegiando dessa forma a representação de elementos funcionais passivos do COMONLIFE, puderam ser definidas quais estruturas de informação e quais relacionamentos entre essas estruturas deveriam ser armazenadas internamente ao sistema de modo a representar completa e acuradamente o comportamento do (sub)sistema de controle. A visão das estruturas de informação e de seus relacionamentos complementou o entendimento da solução expressa parcialmente através do esquema de atividades.
- A **Organização Hierárquica das Atividades** do COMONLIFE estabeleceu uma representação mais inteligível para a estrutura da solução abstrata. Construída através de um processo de agregação/segmentação em subsistemas (atividades agregadas) baseado em critérios já usados na organização do Modelo do Contexto, facilitou bastante o processo de simular a execução da especificação e consolidou o entendimento do propósito do artefato de controle/monitoramento.
- As **Listas de Pré- e Pós- Condições**, constituindo especificações declarativas textuais, definiram precisamente, para cada atividade operacional primitiva, a relação entre entradas e saídas a ser estabelecida pela implementação do COMONLIFE sem, no entanto, prescrever uma solução concreta, procedimental. A elaboração dessas listas suprimiu ambigüidades remanescentes na especificação gráfica dos requisitos essenciais.

O **Dicionário de Dados** permitiu especificar rigorosamente a **semântica** associada a cada elemento funcional passivo tratado pelo sistema, a **estrutura** da informação agregada e o **tipo** de cada informação elementar.

Finalmente, o emprego da **Tabela e Verificação de Consistência** contribuiu para garantir a consistência entre os diversos submodelos.

No processo de desenvolvimento do **COMONLIFE**, a modelagem da essência observou um procedimento de experimentação controlada, visando estabelecer a importância de cada um dos submodelos construídos, estimar a consistência de seu uso complementar em conjunto e, em particular, avaliar qualitativamente a utilização das ferramentas e técnicas de modelagem conceitual no processo de desenvolvimento do software associado a sistemas processo-controle. Essa experimentação foi considerada bem-sucedida em virtude da completeza e precisão do Modelo da Essência resultante.

Foi possível concluir que o emprego de instrumentos de modelagem complementares é determinante para garantir a completeza do modelo em relação às necessidades do cliente/usuário a serem atendidas pelo (sub)sistema de controle/monitoramento. Isto é, durante o processo de modelagem, aspectos não identificados em um submodelo tornavam-se claros durante a construção de um outro submodelo com perspectiva complementar à do primeiro. O requisito de consistência imposto ao conjunto de submodelos obriga a revisão do primeiro, refinando-o, e o processo garante a integridade do modelo global.

O experimento permitiu depurar o método original de modelagem visando aperfeiçoar a representação de aspectos importantes relacionados a sistemas processo-controle. Por exemplo, o resultado obtido conduziu a um entendimento não ambígüo das condições normais e anormais impostas ao funcionamento do (sub)sistema de controle/monitoramento a ser implementado. Por outro lado, analisando comparativamente os modelos gerados, foi possível identificar vários aspectos comuns à modelagem de sistemas processo-controle através do método empregado. Essas constatações podem ser melhor apresentadas através de conclusões específicas extraídas da experimentação e apresentadas a seguir, relativas a alguns dos instrumentos de modelagem incorporados ao método.

A **Lista de Eventos Externos,** expressando facilmente a necessidade de reagir a comportamentos "anormais" do ambiente externo, credenciou-se como instrumento indispensável para garantir a segurança do sistema processo-controle resultante. Além disso, permitindo estabelecer correta e completamente o conjunto de necessidades a serem atendidas pelo (sub)sistema de controle/monitoramento, confirmou-se como um instrumento válido para garantir a eficácia desse artefato ao estabelecer um enunciado de problema sucinto e de fácil validação pelo cliente/usuário.

O **Esquema Semântico** foi instrumento indispensável para a construção consistente da **Lista de Eventos Externos**, mostrando muitas vezes a necessidade de acrescentar/suprimir/alterar eventos. Dessa forma, esse esquema permitiu uma compreensão correta do problema a ser tratado pelo subsistema de controle/monitoramento a ser implementado. Para ampliar o poder expressivo desse esquema houve necessidade de adicionar um tipo de resposta interna caracterizado como "resposta interna de controle". Isso deve-se a que, no caso de sistemas processo-controle, a reação à ocorrência de um dado evento externo pode dar início a um curso de ação (comandado por uma ou mais atividades de controle) que manterá o subsistema em funcionamento por um determinado período de tempo ou até a ocorrência de um evento externo específico. É importante ressaltar que, durante a execução de uma resposta interna de controle pode também dar-se o tratamento da ocorrência de outros eventos externos que dão continuidade a uma execução "normal" ou interrompem sua execução em caso de "anormalidade". Concluiu-se, portanto, a existência de dois tipos de reação interna a serem referidos, mas não modelados, no Esquema Semântico:

♦ a **resposta interna de dado**, que também ocorre em sistemas de informação convencionais, caracterizando a necessidade de acesso à Memória Essencial do subsistema de controle/monitoramento, para armazenamento associado a informação recebida como entrada ou para a supressão/alteração de informação armazenada;
♦ a **resposta interna de controle**, que ocorre apenas em sistemas de tempo-real (não necessariamente do tipo processo-controle), caracterizando a necessidade de processamento comandado por atividades de controle, adicional ao processamento estritamente associado à ocorrência do evento externo que provocou inicialmente a reação do sistema.

O **Esquema da Dinâmica** utilizando Redes de Petri, ao complementar a modelagem mais intuitiva proporcionada pelo uso de Diagramas de Estado e Transição, permitiu uma avaliação mais precisa e completa do comportamento do processo a ser controlado, em particular da concorrência de ações realizadas pelo processo que devem ser representadas internamente ao (sub)sistema de controle/monitoramento. A forma de emprego de Redes de Petri introduzida em [Sanchez95.1], proporcionando visões mais abrangentes do processo obtidas através da utilização de Lugares e Conexões Cheias e Lugares Externos, permitiu uma modelagem de alto nível de abstração representando precisamente a dinâmica envolvida. A extensão introduzida pelo conceito de Ramo Determinante permitiu a modelagem simples da causalidade presente no comportamento do processo a ser controlado.

Sumarizando, é possível concluir que a construção do modelo conceitual completo do **COMONLIFE** constituiu um experimento bem-sucedido pois o resultado obtido — o Modelo da Essência — atende ao objetivo para ele fixado: constituir uma especificação completa e precisa dos requisitos essenciais para o sistema a ser construído. Na modelagem do contexto, o fator determinante para a completeza em relação ao levantamento das necessidades do cliente/usuário foi o emprego de instrumentos de modelagem complementares, cada um deles com a função de abordar, sob uma perspectiva bem definida e com a linguagem de representação apropriada, um aspecto específico do enunciado do problema. A precisão decorre da complementaridade desses instrumentos os quais, por não adotarem perspectivas ortogonais, permitem o uso de redundância controlada capaz de garantir a ausência de falhas. Na modelagem do comportamento, de onde provém a solução abstrata do problema enunciado, a completeza decorre do processo dedutivo que é possível utilizar, fundamentando-o em elementos bem definidos do Modelo do Contexto. A precisão do Modelo do Comportamento decorre do emprego de ferramentas conceituais, com sintaxe e semântica rigorosamente definidas, e de técnicas de modelagem também rigorosas. A especificação dos requisitos essenciais que resulta dessa modelagem conduz a um entendimento não ambígüo do comportamento desejado para o sistema a ser implementado.

A partir dessa modelagem conceitual foi possível definir um conjunto de técnicas de transição que permite uma evolução lógica suave para o Modelo da Implementação. Assim, foi possível derivar um **design** baseado no Modelo da Essência, caracterizado por subsistemas básicos que implementam classes fundamentadas no encapsulamento de Tipos de Entidade e Atividades Essenciais. Cada classe trata um conjunto bem definido de eventos externos e, devido às características de rastreamento dos diversos modelos a partir da Lista de Eventos Externos, conclui-se que esses elementos de modelagem — os eventos externos — poderão ser utilizados sistematicamente em um método destinado a garantir reúso de especificações, integrando-se de forma natural a métodos baseados em Análise de Domínios. A implementação dessas classes em termos de subsistemas básicos em nada alterou sua generalidade, se comparada à alternativa de design orientado a objetos.

Com base no trabalho de pesquisa e desenvolvimento realizado, novas pesquisas podem ser empreendidas no sentido de estabelece-ser uma técnica de recuperação de modelos, com apoio automatizado, baseada em coincidência/analogia no tratamento de eventos externos. Essa técnica deverá facilitar o reúso do subconjunto de modelos — gerados segundo a estrutura de modelagem proposta para a essência e para o design — que se referem ao tratamento de um grupo previamente definido de eventos externos.

Referências Bibliográficas

Akerman91 Akerman C. L.; Modelagem da Essência de Sistemas de Tempo-Real - Estudo de Caso para um Sistema de Controle e Monitoramento de um Processo de Litografia por Feixe de Elétrons; Trabalho de Disciplina de Mestrado na Área de Engenharia de Software; 1991.

Barbosa92 Barbosa E., Maffeo B.; Modelagem da Essência de um Sistema de Controle para uma Área de Linhas de Engarrafamento; Monografias em Ciência da Computação, nº 29/92; Departamento de Informática, PUC- Rio.; 1992

Barbosa95 Barbosa E., Maffeo B.; Modelagem da Implementação (Design) de um Sistema de Controle para uma Área de Linhas de Engarrafamento de um Composto Químico; Monografias em Ciência da Computação, nº 11/95; Departamento de Informática, PUC- Rio; 1995.

Bruyn88 Bruyn W., Jensen R., Keskar D. e Ward P.T.; "ESML: An Extended System Modelling Language Based on Data Flow Diagram"; ACM Sigsoft, Software Engineering Notes, 13(1): 58-62; janeiro de 1988.

Chen76 Chen P.P.S.; The entity-relationship model - toward a unified view of data; ACM Transactions on Database Systems, 1(1): 9-36; 1976.

Clemente92 Clemente K.; Modelagem de Sistemas Sócio-Técnicos, Estudo de Caso de um Piloto Automático para Automóvel; Tese de Mestrado, Departamento de Engenharia Elétrica, PUC-Rio; abril de 1992.

Glass94 Glass R. L.; The Software-Research Crisis; IEEE Software; novembro de 1994

ISO695 Griethuysen J.J.(editor); Concepts and Terminology for the Conceptual Schema and the Information Base; International Standartization Organization (ISO), ANSI, Secretariat ISO/TC97/SC5; New York, nº 695.

Jaffe91 Jaffe M.S. et al.; Software Requirements Analysis for Real-Time Process-Control Systems; IEEE Transactions on Software Engineering; março de 1991.

Maffeo90 Maffeo B., Ritto A.C.A.; O Esquema Semântico no Modelo do Contexto de um Sistema Computacional; XXIII Congresso de Informáica da SUCESU; 1990.

Maffeo91 Maffeo B.; ESML: Uma Revisão de Apresentação, Estrutura, Notação e Conteúdo; Monografias em Ciência da Computação 1/91, Departamento de Informática, PUC-Rio;1991.

Maffeo92 Maffeo B.; Engenharia de Software e Especificação de Sistemas; Editora CAMPUS; 1992.

Magee90 Magee J., Kramer J. e Sloman M.; An Overview of the REX Software Architecture; apresentado para IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems, Cairo; outubro de 1990

McMenamin84 McMenamin M., Palmer J. F.; Essential Systems Analysis; Yourdon Press; 1984.

Richter92 Richter G., Maffeo B.; Towards a rigorous interpretation of ESML - Extended System Modeling Language; IEEE Transactions on Software Engineering, vol. 19 nº 2; 1993, p.165.

Ritto91a Ritto A.C.A, Maffeo B. ; Definindo o Problema a ser Tratado por um Sistema Computacional - Modelo do Contexto; Monografias em Ciência da Computação 11/91, Departamento de Informática, PUC-Rio; 1991.

Sanchez94 Sanchez M. L., Maffeo B., Leite J. C. S. P. ; Ferramentas e Técnicas para a Modelagem da Essência de Sistemas de Tempo-Real para Controle e Monitoramento de Processos; Anais do 10º Congresso Brasileiro de Automática e do 6º Congresso Latino Americano de Controle Automático, Rio de Janeiro; setembro de 1994.

Sanchez95a Sanchez M. L., Maffeo B; Sistema de Tempo-Real para Controle de Processos - Design Orientado a Encapsulamento de Dados e a Troca de Mensagens entre Subsistemas Autônomos; Série Monografias em Ciência da Computação 13/95, Departamento de Informática, PUC-Rio; 1995.

Sanchez95b Sanchez M. L., Maffeo B; Modelagem da Essência de um Sistema de Litografia por Feixe de Elétrons - COMONLIFE; Série Monografias em Ciência da Computação 29/95, Departamento de Informática, PUC-Rio; 1995.

Sanchez95c Sanchez M. L., Maffeo B; Design Baseado em Troca de Mensagens entre Subsistemas Autônomos de um Sistema para Controle e Monitoramento de um Processo de Litografia por Feixe de Elétrons - COMONLIFE; Série Monografias em Ciência da Computação 42/95, Departamento de Informática, PUC-Rio; 1995.

Ward85 Ward P.T. e Mellor S.J.; Structured Development for Real-Time Systems; Yourdon Press; 1985.

Sanchez96 Sanchez M.L.; Um Metodo de Design de Sistemas de Tempo-Real Baseado em Experimentacao; Thesis in partial fulfillment of the Doctoral Degree in Informatics, Pontifical Catholic University of Rio de Janeiro (PUC-Rio); 1996

Gane79 Gane, C., Sarson T.,; Structured Systems Analysis: Tools and Techniques; Fourth Edition; Englewood Cliffs, N.J., Prentice Hall; 1979.

DeMarco78 DeMarco T.; Structured Analysis and System Specification; Englewood Cliffs, N.J., Prentice Hall; 1978.