
AutoSTEP

UM AMBIENTE DISTRIBUÍDO PARA A INTEGRAÇÃO DE SISTEMAS

AutoSTEP

A DISTRIBUTED ENVIRONMENT FOR SYSTEM INTEGRATION

Eliane Gomes GUIMARÃES*
Rogério Almeida BARRA**

ABSTRACT

This paper presents a proposal for the design and implementation of a distributed, heterogeneous and cooperative environment for supporting the integration of systems in the scope of the automotive industry. This task will demand an open architecture where heterogeneous systems can interact dynamically. The needs for information sharing and interoperability between software systems encourage the introduction of standards like STEP ("Standard for the Exchange of Product Model Data") and CORBA ("Common Object Request Broker Architecture"). AutoSTEP is an open distributed environment being developed at CTI where the CORBA architecture is used as infrastructure for integrating cooperative applications, mainly those related to the STEP standard.

KEY WORDS: Open Distributed Systems, Cooperative Applications, STEP, CORBA Architecture.

RESUMO

Este artigo apresenta uma proposta de especificação e implementação de um ambiente distribuído, heterogêneo e cooperativo para o suporte à integração de sistemas no âmbito da indústria automobilística. Isto requer uma arquitetura aberta onde sistemas heterogêneos interajam dinamicamente. A necessidade de compartilhamento de informações e interoperabilidade entre os sistemas de software motiva o surgimento de padrões, tais como STEP ("Standard for the Exchange of Product Model Data") e CORBA ("Common Object Request Broker Architecture"). AutoSTEP é um ambiente distribuído aberto em desenvolvimento no CTI onde a arquitetura CORBA é utilizada como infra-estrutura de integração de aplicações cooperativas, notadamente aquelas relacionadas com o padrão STEP.

PALAVRAS-CHAVE: Sistemas Distribuídos Abertos, Aplicações Cooperativas, Padrões STEP, Arquitetura CORBA.

1. INTRODUÇÃO

A necessidade do compartilhamento de informação e outros recursos motivou o surgimento de sistemas

computacionais de natureza distribuída. Em tais sistemas um conjunto de componentes de hardware e software

(*) Pesquisadora da Fundação Centro Tecnológico para Informática (CTI).

(**) Pesquisador da Fundação Centro Tecnológico para Informática (CTI) - Professor do Programa de Mestrado em Informática da PUCAMP.

interagem de forma harmoniosa e cooperativa. Via de regra, os sistemas distribuídos (SDs) escondem do usuário os detalhes inerentes a distribuição do processamento, tais como localização geográfica dos componentes, replicação de dados e funcionalidades, e ocorrência de falhas. Esta propriedade é denominada transparência e constitui um ponto crítico no desenvolvimento de sistemas distribuídos.

Historicamente, os primeiros SD's utilizavam a troca de mensagens como elo de comunicação entre os seus componentes. Este mecanismo emprega apenas os recursos oferecidos pela interface de programação dos protocolos de transporte (por exemplo, a interface de sockets comumente associada à pilha de protocolos TCP/IP).

Um marco importante no desenvolvimento de SD's foi a extensão da programação estruturada para ambientes distribuídos, através do modelo cliente/servidor. Neste modelo os componentes assumem papéis de clientes (requisitantes de serviços) ou servidores (provedores de serviço). Assim sendo, a aplicação é estruturada segundo serviços (implementados em procedimentos) que servidores colocam à disposição de clientes. Clientes evocam serviços através de um mecanismo de interação denominado Chamada de Procedimento Remoto (RPC: Remonte Procedure Call). Este mecanismo permite que um componente evoque procedimentos definidos fora do seu espaço de endereçamento, passando parâmetros e recebendo resultados, de forma similar a evocação de procedimentos locais. Esta propriedade torna a programação de sistemas distribuídos similar à programação de sistemas centralizados e estruturados.

Da mesma forma que o paradigma da programação estruturada foi estendido para ambientes distribuídos, o mesmo vem ocorrendo com o paradigma de programação orientada a objetos. Nesta extensão, os objetos se localizam em diferentes espaços de endereçamento e são capazes de conduzir computações em paralelo e de forma autônoma (isto é, os objetos são ativos). Nesta linha, vários modelos de referência e arquiteturas estão sendo propostos com o intuito de facilitar a implementação de objetos em ambientes distribuídos. Cita-se aqui o modelo ODP ("Open Distributed Processing") proposto pela ISO ("International Organization for Standardization") e as arquiteturas CORBA ("Common Object Request Broker Architecture") da OMG ("Object Management Group"); DSOM ("Distributed System Object Model") da IBM Corporation e COM ("Component Object Model") da Microsoft Corporation.

A maior motivação no emprego destes padrões é permitir que produtos complexos sejam integrados aos sistemas distribuídos sem a necessidade de adaptações (que inevitavelmente se tornam focos de problemas). Exemplo de tais produtos são: bases de dados, sistemas gráficos de visualização e pacotes de CAD ("Computer-Aided Design"). Para interoperarem, tais produtos necessitam aderir a um ou mais padrões de interoperabilidade. Neste aspecto, a arquitetura CORBA vem despertando grande interesse como elemento de interoperabilidade entre aplicações, sendo que produtos para o desenvolvimento de aplicações aderentes ao CORBA já estão disponíveis comercialmente.

A arquitetura CORBA [4] define uma arquitetura orientada a objeto que permite às aplicações interagirem sem qualquer conhecimento da infra-estrutura de comunicação (redes, sistemas operacionais, etc). Objetos definem suas interfaces através de uma linguagem comum e, com base nestas definições, um sistema de runtime denominado Agente de Requisições de Objetos ("Object Request Broker - ORB") intermedia a comunicação entre estes objetos. Esta intermediação inclui a localização e instanciação de servidores, a passagem dos parâmetros estipulados na interface e a coleta do resultado da evocação. A interação via ORB provê um nível de abstração maior na comunicação quando comparado aos mecanismos baseados em passagem de mensagem ou RPC.

Aplicações de interesse neste trabalho são da classe CIME ("Computer Integrated Manufacturing and Engineering"). Tais aplicações tipicamente requerem a habilidade para compartilhar e trocar dados em um ambiente distribuído, aberto, heterogêneo e cooperativo. Neste sentido, modelos de informação como proposto pelo padrão STEP ("Standard for the Exchange of Product Model Data"), de interoperabilidade como proposto pelos padrões do OMG e de acesso a informação como proposto pelo ODMG ("Object Data Management Group") devem ser considerados. Este trabalho examina algumas propostas de ambientes para desenvolvimento de aplicações CIME e introduz o AutoSTEP, um protótipo de ambiente computacional com esta finalidade.

O projeto AutoSTEP tem por objetivo especificar e implementar o protótipo de um ambiente computacional que integre aplicações no âmbito da indústria automobilística permitindo o compartilhamento de dados entre as mesmas. Além disto, o ambiente deve suportar a coordenação entre os agentes ao longo do ciclo de vida de um automóvel.

A Seção 2 deste artigo discute aspectos do problema que o projeto procura resolver. Na Seção seguinte, os

padrões relevantes para o AutoSTEP são introduzidos. Os padrões STEP e CORBA são caracterizados e os seus principais componentes são apresentados. A Seção 4 caracteriza o modelo de informação em que o projeto é baseado e discute sua relevância quanto à prática industrial no setor automobilístico. Uma vez que a arquitetura de sistema, metodologias e ferramentas computacionais a serem adotadas no projeto AutoSTEP terão como referência esforços anteriores, estes são apresentados na Seção 5. Em especial, será considerado o projeto CoConut ("Computer Support for Concurrent Design Using STEP") [5]. Outros esforços relevantes incluem o projeto PISA ("Platform for Integration of CIME Applications") [6] e a abordagem orientada a objetos para a implementação de sistemas de visualização baseados em STEP proposta em [1]. A Seção 6 caracteriza áreas em que o projeto AutoSTEP trará contribuições a estes esforços.

2. CARACTERIZAÇÃO DO PROBLEMA

Tecnologia de informação tem sido empregada pelas empresas do setor automobilístico em diversas funções do sistema produtivo, abrangendo, entre outros, sistemas CAD, CAE, CAM e aplicações para gerenciamento de configuração de produtos. Tais sistemas geram e utilizam dados de produtos em forma digital. Tipicamente, a comunicação de dados de produtos entre funções se dá via papel ou interfaces específicas. Tais interfaces se fazem necessárias, uma vez que aplicações requerem dados de produtos em diferentes formatos. A incompatibilidade de formatos conduz a uma redução dos benefícios advindos da introdução de tecnologia de informação devido a barreiras de comunicação entre aplicações.

Um dos principais fatores que contribuem para a situação descrita acima é o fato dos sistemas serem desenvolvidos de forma isolada, impossibilitando o compartilhamento de dados entre as aplicações. É importante salientar os custos associados a estas barreiras de comunicação entre os sistemas, os quais incluem:

- reentrada de dados no sistema produtivo;
- erros na transferência de dados entre sistemas;
- necessidade de compatibilização de dados presentes em mais de um lugar;
- dificuldade de acesso a informações requeridas por agentes do sistema produtivo.

Uma questão fundamental relativa a estes custos é que eles não agregam valor ao produto.

A principal característica de tal ambiente computacional é a ausência de um modelo de informação que capture a semântica comum às várias aplicações executando tarefas no sistema produtivo. Devido à inexistência deste modelo, vários fatores contribuem para a ineficiência global do sistema produtivo, entre os quais podem ser citados:

- a conversão de dados consome muitos recursos;
- a necessidade de conversão implica em operação seqüencial do sistema produtivo, impedindo atividades concorrentes;
- a necessidade de desenvolvimento de interfaces implica em resposta ineficiente a mudanças nos sistemas computacionais, requeridas por novas situações de negócios;
- recursos humanos são dispendidos para a localização e compatibilização de dados de produtos;
- a necessidade de conversão implica em redução da qualidade dos dados, conduzindo a decisões de qualidade inferior.

Em síntese, do ponto de vista dos objetivos estratégicos de uma empresa, a ineficiência de comunicação de dados de produtos entre sistemas resulta em:

- maiores custos;
- menor qualidade dos produtos;
- maior prazo para introdução de novos produtos no mercado;
- dificuldade de adaptação do sistema produtivo a novas situações de negócios;
- sistemas computacionais que provêm suporte deficiente aos negócios das empresas.

2.1 TECNOLOGIAS PARA DADOS DE PRODUTOS

A busca de soluções para as questões mencionadas no item anterior, exige uma análise sistemática da aplicação de tecnologia de informação a todos os aspectos concernentes ao desenvolvimento, manufatura e operação de produtos. Tecnologia para dados de produtos (TDP), como um conjunto de técnicas, métodos e ferramentas visando a efetiva comunicação de dados de produtos no âmbito de uma empresa e entre esta e seus fornecedores, surge como resposta a esta necessidade. Em última análise, TDP objetiva prover informação

correta, no lugar correto e em forma correta, no instante em que esta se faça necessária.

Conforme ilustrado na Figura 1, o componente central em TDP é um modelo de produto integrado, de tal forma que aplicações enxergam umas às outras através deste modelo, ou seja, enquanto o modelo descreve todas as informações ao longo do ciclo de vida do produto, diferentes aplicações representam visões particulares das informações sendo integradas.

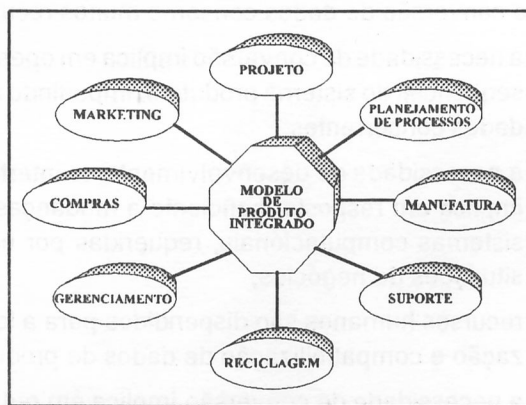


Figura 1 - Modelo de produto em um sistema produtivo

Uma importante característica do problema que compõe TDP é a heterogeneidade do sistema produtivo, tanto no que diz respeito a hardware quanto a software. Esta heterogeneidade é causada pela necessidade de otimização das funções suportadas por aplicações computacionais, o que implica na escolha de um sistema segundo critérios econômicos e funcionais, de forma a atender os processos de negócios de uma empresa. Além disto, a crescente interação entre clientes e fornecedores, no âmbito da indústria automobilística, requer aplicações aptas, a priori, a se comunicarem entre si.

A abordagem mais efetiva, em termos de custos e tempo requerido para a integração de sistemas, consiste na utilização de padrões. A Seção seguinte descreve STEP e CORBA, componentes centrais da arquitetura aberta proposta neste projeto.

3. PADRÕES RELEVANTES AO PROJETO AutoSTEP

3.1 PADRÃO STEP

A necessidade de trocar dados de produto em forma digital entre diferentes sistemas computacionais

que suportam as tarefas executadas ao longo do ciclo de vida de produtos, particularmente sistemas CAD, foi identificada no final dos anos setenta. Em consequência, vários padrões para troca de dados de produtos, tais como, IGES ("Initial Graphics Exchange Specification" - EUA), SET ("Standard d'Echange et de Transfert" - França) e VDA-FS ("Verband der Deutschen Automobilindustrie - Flachschnittstelle" - Alemanha), foram desenvolvidos durante os anos oitenta.

Deficiências identificadas através da utilização destes padrões, tais como as ambigüidades de suas definições, as restrições referentes ao escopo de dados de produtos representados e a falta de requisitos formais para verificação de conformidade, conduziram à formação, em dezembro de 1983, de um comitê no âmbito da ISO ("ISO 10303 - Technical Committee 184 - Subcommittee 4") com o objetivo de especificar um padrão internacional para modelos de dados de produtos. Os objetivos deste esforço, informalmente conhecido como STEP, podem ser sintetizados da seguinte forma:

- * prover a descrição completa, não-ambígua e processável por computador das características físicas e funcionais de produtos ao longo de todo o seu ciclo de vida;
- * prover mecanismos que possibilitem troca de dados de produtos e compartilhamento destes entre funções de um sistema produtivo.

Uma discussão mais detalhada dos requisitos impostos a STEP e da metodologia utilizada em sua especificação pode ser encontrada em [2] e [3]. Cabe, no entanto, salientar que enquanto a especificação do padrão envolve várias questões em aberto, do ponto de vista de pesquisa e desenvolvimento, seu elemento catalizador tem sido a demanda industrial por efetiva comunicação entre aplicações computacionais concernentes ao ciclo de projeto, manufatura e operação de produtos. Neste sentido, STEP conta com a participação de mais de 500 especialistas de dezenas de empresas líderes em seus ramos de atuação.

FORMAS DE IMPLEMENTAÇÃO DO PADRÃO

Entre possíveis métodos para implementação de STEP, podem ser citados:

- * **Troca de Arquivo:** nesta abordagem, a sintaxe e estrutura de um arquivo físico usado para armazenar ou transferir dados de produtos são especificadas. A codificação desta estrutura em texto legível é definida pela parte 21 de STEP [10].

• **Interface para Programação de Aplicações:** neste método de implementação, o programador de aplicações tem à sua disposição uma interface para programação, que dá a ele a ilusão, que os dados sendo manipulados, são representados no sistema responsável por seu armazenamento na forma como eles são descritos no modelo de informação.

SDAI (“Standard Data Access Interface”) [11] provê um mecanismo de acesso a dados cuja estrutura é definida usando uma linguagem denominada EXPRESS. Uma implementação SDAI envolve a representação da especificação da interface em uma linguagem de programação, por exemplo C ou C++. Utilizando uma implementação SDAI, programas de aplicação podem ser desenvolvidos independentemente da tecnologia empregada para armazenamento dos dados.

• **Bancos de Dados:** desde que um modelo de informação, relativo a uma aplicação, é especificado usando uma linguagem para modelagem de dados em alto nível, EXPRESS, ele pode ser mapeado para o esquema lógico de diferentes tipos de bancos de dados, tais como relacionais e orientados a objetos. Ferramentas para o mapeamento de modelos EXPRESS em SQL já se encontram disponíveis.

3.2 O PADRÃO CORBA

CORBA [4] é baseado no modelo de objeto OMG, que considera, tanto a informação quanto as aplicações, como objetos que implementam um conjunto bem definido de operações. As operações definem o conjunto de possíveis manipulações de informação, ao passo que, para as aplicações, as operações definem o comportamento que caracteriza as aplicações. Em ambos os casos, o algoritmo que implementa uma operação é considerado um método, que por sua vez é implementado por uma ou mais rotinas. Isto permite que se agreguem múltiplas implementações numa única definição de método. Métodos e suas implementações são invocados através do envio de mensagens ao objeto definindo o método.

A Figura 2 mostra os quatro elementos principais da arquitetura proposta pelo OMG. O OMG fornece um barramento comum de interconexão que conterá compo-

nentes clientes, serviços básicos necessários para esses componentes e recursos comuns para a colaboração entre componentes.

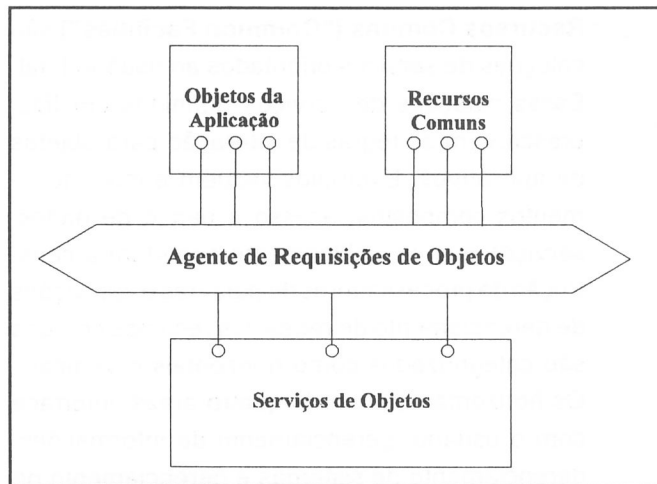


Figura 2 - Arquitetura proposta pelo OMG

Os quatro elementos da arquitetura do OMG são:

• **Agente de Requisições de Objetos - ORB** (“Object Request Broker”) é o meio que estabelece relações cliente/servidor entre objetos. É o mecanismo que permite que um objeto cliente possa invocar de forma transparente um método em um objeto servidor (local ou remoto). O ORB intercepta a chamada e procura um objeto que possa atender o requisito, fornecendo os parâmetros, invocando o seu método e retornando os resultados. Os clientes ignoram os mecanismos utilizados pelo ORB para comunicar, ativar ou armazenar objetos servidores, bem como quaisquer outros aspectos do sistema que não são parte de uma interface do objeto. O CORBA 1.1 definiu uma linguagem genérica de definição de interface IDL (“Interface Definition Language”), utilizada para definir as interfaces dos objetos que se comunicam dentro de uma implementação específica de ORB. O CORBA 2.0 especifica como os ORBs de diferentes fornecedores podem interoperar.

• **Serviços de Objetos (“Object Services”)** são coleções de serviços com interfaces de objeto especificadas em IDL e proporcionam funções básicas para uso e implementação de objetos. Esses serviços ampliam as capacidades dos ORBs e incluem serviços de nomes, notificação de eventos, persistência de objetos, gerenciamento de

ciclo de vida, transações, controle de concorrência e relacionamentos. Alguns serviços estarão disponíveis em breve: consultas, licenciamento, propriedades e segurança.

• **Recursos Comuns (“Common Facilities”)** são coleções de serviços orientados ao usuário final. Essas coleções de serviços definidos em IDL, prescrevem as regras de interação para objetos de aplicativos. Exemplos incluem e-mail, documentos compostos, acesso a banco de dados, serviços que especificamente suportam a construção de processadores de palavras e aplicações de gerenciamento de redes. Os recursos comuns são categorizados como horizontais e verticais. Os horizontais tratam de quatro áreas: interface com o usuário, gerenciamento de informações, gerenciamento de sistemas e gerenciamento de tarefas. Os verticais fornecem interfaces definidas em IDL para suportar a interação de coleções de objetos especializados para a saúde, varejo, finanças e outros domínios. A linha divisória entre recursos comuns e serviços de objetos é imprecisa. Os serviços de objetos precisam ser implementados em todo ORB, enquanto que os recursos comuns são opcionais.

• **Objetos da Aplicação (“Application Objects”)** são objetos específicos para aplicações do usuário final. Estes objetos necessitam ser definidos em IDL a fim de participar em interações mediadas pelo ORB. Interfaces IDL são compiladas resultando na geração de STUBs: elementos utilizados na manipulação de dados entre os objetos e o ORB. Os objetos das aplicações, naturalmente, expandem os serviços proporcionados pelo ORB, pelos recursos comuns e pelos serviços de objetos. Uma aplicação é tipicamente construída de um número grande de classes de objetos básicos; alguns destes podem ser proporcionados pelos serviços comuns do OMG.

A especificação CORBA mais recente (2.0) especifica alguns protocolos de transporte (como o TCP/IP) para permitir a interoperabilidade entre ORBs de fornecedores distintos. A solução, no momento, para este problema de comunicação é a seleção de uma única implementação que suporte todas as plataformas do sistema, ou o uso de objetos gateways, como recomendado na especificação CORBA.

4. UM MODELO DE INFORMAÇÃO PARA A INDÚSTRIA AUTOMOBILÍSTICA

Como parte de STEP, vem sendo definido um modelo de informação que suporta diferentes fases do processo de desenvolvimento de automóveis. Este modelo, especificado no protocolo de aplicação 214 (“Core Data for Automotive Mechanical Design Processes”), descreve informações relativas aos seguintes processos [15]: definição do produto, estilo, projeto, avaliação, planejamento da produção, projeto e construção do maquinário, e controle de qualidade.

Os seguintes aspectos de automóveis são capturados pelo modelo de informação sendo especificado pelo protocolo de aplicação 214:

- peças, montagens de peças, ferramentas e montagens de ferramentas produzidas por fabricantes de automóveis e seus fornecedores;
- dados de definição de produtos e dados para controle de configuração relativos à fase de projeto de um automóvel;
- dados de modificações de projetos e dados relativos à documentação do processo de modificação;
- vários tipos de representação geométrica da forma de peças e ou ferramentas;
- dados relativos à apresentação visual de produtos;
- representação de peças e ou ferramentas por “Form Features”;
- propriedades de materiais utilizados nos produtos;
- dados que descrevem condições superficiais e tolerâncias;
- dados para análise cinemática;
- dados para análise por elementos finitos;
- dados relativos à liberação e aprovação de produtos e versões de produtos;
- dados que registram as modificações de versões de um produto.

O comprometimento da indústria automobilística mundial com este esforço tem sido inequivocamente demonstrado. Os grandes fabricantes europeus e norte-americanos assinaram um documento com o intuito de caracterizar o STEP como o padrão a ser utilizado para a representação e troca de dados de produtos no âmbito da indústria automobilística.

5. PROJETOS CORRELATOS

Nesta seção são discutidos projetos relevantes para os propósitos de AutoSTEP. Enquanto a arquitetura a ser especificada para o ambiente AutoSTEP terá como referência estes esforços, principalmente o projeto CoConut, na Seção subsequente serão evidenciadas áreas em que AutoSTEP trará contribuições.

5.1 O PROJETO CoConut

CoConut ("Computer Support for Concurrent Design Using STEP") [5] é um projeto sendo empreendido no Instituto Fraunhofer-IGD ("Institut für Graphische Datenverarbeitung"), em Darmstadt, Alemanha. CoConut provê um ambiente aberto para integração de aplicações envolvidas na fase de projeto de produtos. Permitindo efetiva comunicação entre agentes em um ambiente heterogêneo e provendo mecanismos para coordenação de suas atividades, o ambiente CoConut objetiva a redução do tempo e custos relativos à fase de projeto de um produto.

Um dos aspectos relevantes da arquitetura proposta em CoConut é suportar a integração de aplicações previamente existentes, como, por exemplo, sistemas CAD disponíveis comercialmente. Além disto, o ambiente inclui aplicações que suportam a cooperação entre agentes, utilizando técnicas CSCW ("Computer-Supported Cooperative Work").

Em CoConut, a integração de dados de produtos é provida por um sistema gerenciador de dados distribuído, cujo modelo conceitual é especificado em conformidade com o protocolo de aplicação 203 ("Configuration Controlled Design") [14] do padrão STEP. Este protocolo suporta, além da descrição da forma do produto (geometria e topologia), a caracterização de sua estrutura e o controle de configuração. Em termos de implementação, um sistema gerenciador de bancos de dados orientado a objetos provê a funcionalidade requerida para gerenciamento de dados distribuídos em um ambiente heterogêneo, incluindo controle de concorrência e processamento de transações.

A Figura 3 apresenta a arquitetura para integração de um sistema CAD com o ambiente CoConut. Linhas contínuas representam transferência de dados, enquanto linhas tracejadas representam comunicação, permitindo troca de mensagens entre aplicações. Uma das aplicações suportadas pelo ambiente é um sistema de visualização cooperativo para modelos de produtos em

conformidade com o protocolo de aplicação 203. A extensão desta aplicação com a finalidade de suportar modelos de produtos definidos pelo protocolo de aplicação 214 é uma das áreas em que o projeto AutoSTEP trará contribuições ao ambiente CoConut.

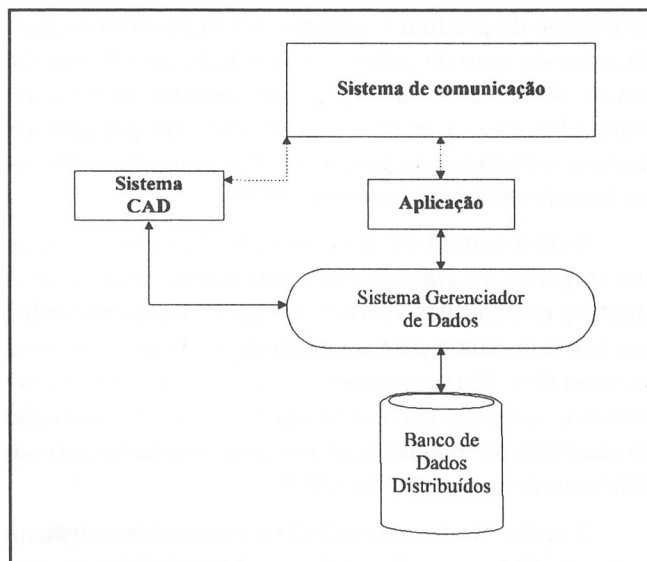


Figura 3 - Arquitetura para integração de sistema CAD em CoConut [5]

5.2 O PROJETO PISA

PISA ("Platform for Information Sharing by CIME Applications") [6] é um projeto sendo empreendido por uma conjunto de empresas e instituições de pesquisa européias, como parte do projeto ESPRIT. O objetivo do projeto PISA é contribuir para o estabelecimento de uma infra-estrutura para o compartilhamento e troca de informações de produto e processo entre aplicações CIME ("Computer Integrated Manufacturing and Engineering"). Para alcançar este objetivo, o projeto PISA considera padrões emergentes, tais como, STEP, CORBA e ODMG na área de bancos de dados.

5.3 A ESTRUTURA DE SOFTWARE DO SVwork

SVwork é uma estrutura de software que explora o paradigma de orientação a objetos, com o objetivo de facilitar a implementação de sistemas de visualização em conformidade com STEP [1].

O modelo de apresentação em STEP, como parte dos recursos integrados, provê informações para a especificação de aspectos visuais de propriedades de

produtos, tais como forma e tolerâncias [13]. Dados de apresentação, em conjunção com dados que descrevem propriedades do produto, possibilitam a geração de imagens em conformidade com STEP.

Sistemas de visualização baseados em STEP devem gerar uma imagem correspondente a uma instância de modelo de produto e suportar interação com o usuário, a fim de permitir a definição e edição de informação visual. Além disto, as ações do usuário devem ser mapeadas para a instância do modelo de produto, de forma que os dados de apresentação resultantes possam ser transferidos para outros sistemas.

A idéia central em SVwork consiste em especificar um conjunto de classes (no contexto de orientação a objetos), que capture a semântica de recursos integrados que são relevantes para a visualização de propriedades de produtos. Estas classes provêm uma estrutura de software que pode ser reutilizada para a implementação de sistemas de visualização em conformidade com um protocolo de aplicação de STEP.

O protocolo de aplicação 214, em contraposição ao protocolo 203, especifica informações relativas a apresentação de produtos, através da interpretação de recursos integrados que constituem o modelo de apresentação em STEP. Sendo assim, para a implementação de sistemas de visualização baseados no protocolo 214, mostra-se oportuna a utilização do conjunto de classes definido em SVwork.

6. O PROJETO AutoSTEP

O CORBA é utilizado no projeto AutoSTEP para proporcionar a infra-estrutura básica necessária para a integração de sistemas distribuídos no âmbito da indústria automobilística. Um dos aspectos relevantes para alcançar este objetivo é a utilização de Serviços de Objetos ("Object Services") e Recursos Comuns ("Common Facilities").

Algumas propostas para a implementação de classes diferentes de Serviços de Objetos estão sendo estudadas para ampliar as capacidades do Agente de Requisições de Objetos (ORB).

Os Recursos Comuns terão um papel fundamental no projeto de novos sistemas CIM ("Computer Integrated Manufacturing"), mais ágeis (rápida resposta a mudanças), distribuídos e reusáveis. Os Recursos Comuns proporcionarão funcionalidades que podem ser incorporadas através de muitos domínios da aplicação, tais

como: manufatura, projeto, análise, simulação e práticas de negócios. Os Recursos Comuns juntamente com os Serviços de Objetos do OMG suportarão implementações modulares, baseadas em padrões que promovem a agilidade de desenvolvimento e modificação de sistemas.

Os trabalhos começaram com a definição de recursos verticais para manufatura [7]. Atualmente, foram identificadas três áreas específicas de especialização: Especialização de Serviços de Dados do Produto ("Product Data Services Specialization"), Especialização de Gerenciamento de Políticas Diversas ("Policy Variable Management Specializations"), Especialização de Gerenciamento de Históricos ("History Management Specializations").

A área de especialização de Serviço de Dados do Produto suporta o compartilhamento de informação em um ambiente distribuído que pode ultrapassar os limites da empresa. A integração de tecnologias de informação intra/inter-organizacional depende da criação de um modelo conceitual baseado em padrões. Entretanto torna-se necessário coordenar a integração e interoperação de padrões em vez de desenvolvê-los de forma isolada. Especificamente, o Serviço de dados do produto deve ser completamente aderente com os padrões STEP e métodos de implementação. Deve ainda utilizar o SDAI ("Standard Data Access Interface") para manter o Serviço independente de tecnologia de armazenagem de dados e para satisfazer os requisitos de compartilhamento de dados de produto, identificados pelos grupos que desenvolvem o SDAI. O Serviço de dados do produto de STEP necessita concordar com um padrão para implementação de tecnologia de dados distribuídos. O OMG proporciona um padrão de consenso internacional para esta tecnologia.

A área de especialização de Gerenciamento de Políticas Diversas suporta a configuração e gerenciamento de variáveis de gerenciamento de sistemas.

A área de especialização de Gerenciamento de Históricos suporta o armazenamento e acesso de dados armazenados em arquivos.

Estes recursos ainda estão em fase de estudos. No caso do projeto AutoSTEP, é relevante para este projeto a área de especialização de Serviços de Dados de Produtos e, como ainda estes serviços não estão disponíveis, propõe-se sua utilização via Adaptadores de Objetos ("Object Adapters") que conectam ferramentas de desenvolvimento STEP ao ORB (Figura 4).

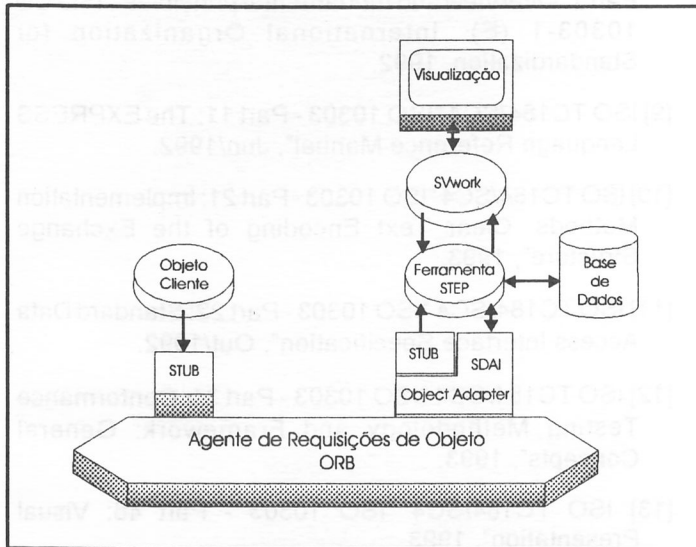


Figura 4 - Arquitetura do Projeto AutoSTEP

O projeto AutoSTEP deve ser visto como uma extensão aos esforços descritos na seção anterior. Os principais aspectos do projeto são:

- o modelo de produto a ser suportado, especificado no protocolo de aplicação para a indústria automobilística, estende aquele referente ao protocolo suportado no projeto CoConut, abrangendo, além da fase de projetos, outras fases do ciclo de vida do produto;
- a abordagem utilizada no projeto PISA para interoperabilidade entre aplicações, baseada no padrão CORBA, deve ser incorporada caracterizando, com isto, uma arquitetura aberta;
- a utilização da estrutura de software SVwork permitirá a implementação de sistemas de visualização de produtos baseados em STEP mais flexíveis que aquele provido pelo projeto CoConut. O acesso ao SVwork se processará via ORB através de um adaptador de objeto específico;
- o desenvolvimento do SDAI Object Adapter servirá para possibilitar a integração de servidores de sistemas, que podem somente ser acessados via uma interface SDAI (ele traduz as mensagens recebidas do ORB em mensagens semanticamente equivalentes às chamadas SDAI), conforme ilustrado na Figura 4.

Numa primeira etapa o projeto AutoSTEP irá utilizar os seguintes componentes de software disponíveis comercialmente:

- uma implementação da arquitetura CORBA (por exemplo, o ORBIX da Iona Technologies);
- um conjunto de ferramentas de desenvolvimento STEP (por exemplo, o da STEP Tools Inc.). Estas ferramentas propiciam uma interface para programação de aplicações que torna transparente a forma como os dados são armazenados;
- uma base de dados orientada a objetos (por exemplo, o ObjectStore da Object Design) para armazenamento das estruturas de informação prescritas pelo STEP e manipulação destas por programas de aplicação.

7. CONCLUSÕES

Este artigo apresenta a motivação e os aspectos mais relevantes dos estudos preliminares para o projeto AutoSTEP. A importância do padrão STEP para o setor automobilístico é caracterizada e a necessidade da harmonização de STEP com o padrão CORBA é examinada.

Além disto, esforços correlatos aos objetivos de AutoSTEP são sumarizados.

Além de permitir a integração de aplicações, tais como sistemas CAD e sistemas para controle de configuração de produtos, o ambiente AutoSTEP deverá oferecer uma plataforma em que aplicações CSCW podem ser desenvolvidas, com o intuito de promover cooperação entre agentes envolvidos no projeto de um automóvel, o que vem de encontro a um dos pressupostos básicos da abordagem para o desenvolvimento de produtos conhecida como engenharia simultânea.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Barra, R. A., "On the Implementation of Systems for Realistic Visualization of STEP Models", tese de doutorado, Universidade Técnica de Berlim, Dez/1994.
- [2] Barra, R. A. e J. M. Adán, "STEP como um Componente Estratégico para a Integração de Sistemas de Manu-

- fatura", Congresso sobre CAE/CAD/CAM e Computação Gráfica, SOBRACON, Abr/1995.
- [3] Barra, R. A. e Guimarães, E. G., "AutoSTEP: um Ambiente Integrado para a Indústria Automobilística", III Simpósio Exposição CAE/CAD/CAM, SAE Brasil, Ago/1995.
- [4] "The Common Object Request Broker: Architecture and Specification" (Version 1.2 Revision Draft), OMG Document Number 93.12.43, The Object Management Group, Dec/1993.
- [5] Jasnoch, U., H. Kress, K. Schroeder e M. Ungerer, "CoConut: Computer Support for Concurrent Design Using STEP", IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Abril/1994, EEE Computer Society Press.
- [6] Koethe, M., A. Nieva e F. Schoenefeld, "Product Data Exchange in Open Systems: The PISA Approach", Autofact, EUA, Dez/1993.
- [7] "Common Facilities Architecture". OMG Document 95-1-2; Revision 4.0; Jan. 3, 1995.
- [8] "Product Data Representation and Exchange. STEP Part 1: Overview and fundamental principles", ISO CD 10303-1 (E), International Organization for Standardization, 1992.
- [9] ISO TC184/SC4 "ISO 10303 - Part 11: The EXPRESS Language Reference Manual", Jun/1992.
- [10] ISO TC184/SC4 "ISO 10303 - Part 21: Implementation Methods: Clear Text Encoding of the Exchange Structure", 1993.
- [11] ISO TC184/SC4 "ISO 10303 - Part 22: Standard Data Access Interface Specification", Out/1992.
- [12] ISO TC184/SC4 "ISO 10303 - Part 31: Conformance Testing Methodology and Framework: General Concepts", 1993.
- [13] ISO TC184/SC4 "ISO 10303 - Part 46: Visual Presentation", 1993.
- [14] ISO TC184/SC4 "ISO 10303 - Part 203: Application Protocol: Configuration Controlled Design", 1993.
- [15] ISO TC184/SC4 "ISO 10303 - Part 214: Application Protocol: Core Data for Automotive Design Processes", 1994.