

GerDoc Ábacus – Um Sistema de Gerenciamento Eletrônico de Documentos Técnicos para Ambientes de Engenharia/CAD

Joacir Giaretta

Universidade de Caxias do Sul (UCS)
Campus Universitário da Região dos Vinhedos – Depto. de Engenharia e Informática
Alameda João Dal Sasso, 800 – 95700-000 – Bento Gonçalves – RS - Brasil
joacir@abacus.inf.br

Abstract. *Design environments require support for the cooperation between designers in the project conception, which is not provided by the environments used in conventional applications. To provide such support, several resources and mechanisms for data and project management are presented in the present paper, as a management system called GerDoc Ábacus, designed to support technical documentation in project environments. The main objective of GerDoc Ábacus is to support the project data consistency and the integration of project tasks allocated to designers, which cooperate in a distributed environment. An interactive interface is proposed, and supported by a set of mechanisms able to provide an adequate level of integration between the designers that cooperate in the same project, working together for long periods of time. The system supports also the documentation of all project steps.*

Resumo. *Ambientes de engenharia apresentam a forte característica da necessidade de cooperação entre projetistas na concepção de projetos CAD, o que provoca uma série de problemas em relação aos ambientes usuais encontrados em aplicações convencionais. Na busca de solucionar tais problemas, vários recursos e mecanismos relativos às gerências de dados e do processo de projeto são apresentados nesse artigo, que visa apresentar um sistema gerenciador de documentação técnica para ambientes CAD chamado GerDoc Ábacus, cuja proposta é baseada na busca da solução dos problemas relativos à consistência dos dados de projetos e da integração de tarefas de projetistas que interagem em ambientes distribuídos de projeto. Unindo vários mecanismos, é proposta uma interface interativa, no intuito de manter a harmonia entre projetistas que fazem parte de equipes de projetos que são mantidos em atividade durante longos períodos de tempo, além de documentar todos os passos realizados acerca de cada um desses projetos.*

1. Introdução

Historicamente, projetos de engenharia se caracterizam por serem compostos por objetos com estruturas complexas, criados por grupos de projetistas trabalhando em ambientes de cooperação. A complexidade desses projetos traz a tona uma série de problemas, como as múltiplas representações possíveis para um único objeto de projeto. Comum em *ambientes de projetos de engenharia*, a *modularização* [Wagner, 1994] separa objetos em sub-objetos, garantindo maior facilidade na sua manipulação, mas cria a necessidade de controlar tais módulos, de maneira que o ambiente de cooperação entre os projetistas não seja afetado. Assim, se torna fundamental a integração das tarefas de engenheiros e projetistas através de ambientes distribuídos, controlando e auxiliando nas etapas de desenvolvimento dos objetos de projeto.

A literatura apresenta o conceito de *framework* como sendo uma plataforma que provê um ambiente operacional que disponibiliza facilidades para a manutenção e o gerenciamento dos dados criados, além do conhecimento a respeito do processo de projeto, sendo capazes de conduzir o projetista no desenvolvimento de suas tarefas [Soares, 1996]. Nesse contexto, vários *frameworks* destinados a ambientes de engenharia são encontrados, como o STAR [Wagner, 1991], JESSI-Common-Frame [Kathöfer, Miller, 1992] e NELSI [Bosch, Bingley, Wolf, 1991]. Todos possuem a preocupação de oferecer recursos de *gerência do processo de projeto* e de *gerência de dados* para garantir a consistência dos ambientes integrados de projeto.

Característica presente em *frameworks*, a *documentação dos passos de projeto* visa unir os recursos de gerência de dados e do processo de projeto, objetivando manter o histórico do ciclo de vida dos projetos, controlando atividades de projetistas, não permitindo acessos de projetistas não ligados a projetos particulares, gerenciando a modularização dos projetos e o versionamento desses módulos, além de oferecer uma interface totalmente interativa, garantindo facilidades de operacionalização na busca de versões de objetos específicos e no controle da execução das tarefas necessárias ao cumprimento dos objetivos fixados.

Atualmente, as características que integram um *framework* compõem também os sistemas de Gerenciamento Eletrônico de Documentos – GED específicos para ambientes de engenharia, conhecidos como EDMS (*Engineering Document Management System*). Tais sistemas estão se tornando populares nesses ambientes, exatamente por proverem os recursos de gerência do processo de projeto e de dados. Em [Baldam, Valle, Cavalcanti, 2002] são encontrados conceitos diversos relativos a GED e em [Baldam, 2003] é apresentada uma aplicação de EDMS no setor siderúrgico. Os conceitos e a aplicação de EDMS também podem ser encontrados em [Peltonen, Männistö, Alho, Sulonen, 1993], que apresenta um estudo de caso em uma empresa fabricante de elevadores.

A proposta desse artigo é a de apresentar os conceitos e ferramentas relativos a *gerência de dados* e a *gerência do processo* de projetos complexos de engenharia, para depois apresentar o *GerDoc Ábacus*, que é um protótipo de um sistema gerenciador de documentação técnica para ambientes de engenharia (EDMS), descrito detalhadamente em [Giaretta, 2001]. A finalidade principal do *GerDoc Ábacus* está relacionada com a problemática do gerenciamento de projetos de engenharia em ambientes com características próprias, que possuem, como fatores críticos, a necessidade de modularização dos projetos e a participação de vários projetistas na construção destes módulos. O *GerDoc Ábacus* objetiva organizar equipes de projetistas responsáveis por

projetos específicos, a fim de que seja possível controlar todas as tarefas realizadas por cada elemento, através de uma interface amigável e interativa.

O presente artigo está organizado da seguinte forma: o *capítulo 2* descreve as características encontradas em ambientes de projetos de engenharia e introduz o tópico *framework* e sua relação com o gerenciamento de documentação dos passos de projetos; o *capítulo 3* destina-se a explicitar os recursos possíveis de *gerência de dados* de projetos complexos; o *capítulo 4* aborda recursos fundamentais para o trabalho cooperativo, através de mecanismos de *gerência do processo* de projeto complexos; o *capítulo 5* apresenta o *modelo conceitual do GerDoc Ábacus* e seus sub-modelos; o *capítulo 6* apresenta a interface do *GerDoc Ábacus*, caracterizando cada elemento componente e relacionando com os modelos explicitados no *capítulo 5*, e; o *capítulo 7* apresenta conclusões e considerações finais acerca do estudo realizado e do próprio *GerDoc Ábacus*.

2. Ambientes de Projetos de Engenharia

Projetos de engenharia sempre se caracterizaram pela sua complexidade. O processo de um projeto complexo de engenharia cria múltiplas representações para cada objeto que o compõe, em níveis de abstração diferentes. Torna-se necessária, então, a existência de ferramentas adequadas para manipulação destes objetos, garantindo autonomia a projetistas e total controle sobre o projeto em si. Uma forma usual de dominar o problema da complexidade é a *modularização* [Wagner, 1994], onde cada projeto considerado complexo é separado em diversos módulos, permitindo aos projetistas a manipulação individual de cada um e, através da *hierarquização*, dividir módulos em sub-módulos, até que se atinja um nível de simplicidade satisfatória.

Através de um controle dos módulos e sub-módulos de um projeto específico, é possível dividir o trabalho em *equipe*, onde vários projetistas podem trabalhar em módulos distintos, considerando dois fatores que são importantes para garantir a funcionalidade do projeto como um todo [Wagner, 1994]: (1) os módulos precisam conhecer a sua hierarquia e; (2) os projetistas precisam trabalhar em ambiente de *cooperação*. Baseado no fato de que projetistas, trabalhando em ambiente de cooperação, precisam interagir entre si e de que os módulos são manipulados por vários projetistas, torna-se fundamental a *exploração do espaço do projeto*, garantindo o retorno a estados anteriores e, a partir daí, o uso de um conjunto de recursos de *refinamento de alternativas* para a obtenção de resultados mais próximos dos desejados.

Um ambiente de projeto deve possuir funcionalidades que possam eliminar os problemas verificados nos ambientes usuais [Wagner, 1994] [Soares, 1996]. Ambientes usuais possuem fatores que diminuem a produtividade global do processo de projeto, como a ausência de um modelo único de representação de dados e a ausência de recursos para a *gerência de dados* e do processo do projeto.

Para solucionar os problemas apresentados, na busca da redução do tempo de desenvolvimento de projetos e conseqüente redução de custos, um ambiente de projeto deve [Giaretta, 2001]: (1) possuir uma boa interface; (2) guiar o projetista no seu trabalho, através de recursos de *gerência de projeto*; (3) permitir acessos diferenciados, dependendo da hierarquia de comando dos projetistas, garantindo o bom trabalho cooperativo; (4) gerenciar e eliminar possíveis inconsistências de dados e controlar as informações da hierarquia dos módulos de um projeto, incluindo a *gerência de versões*

e configurações destes, através de recursos de gerência de dados, e; (5) possibilitar o trabalho em ambientes distribuídos.

A procura de alternativas para redução de tempo de desenvolvimento e custo de projetos complexos levou ao surgimento, na década de 80, dos *frameworks*, que são plataformas que provêem um ambiente operacional comum a ferramentas de projeto (os programas utilizados), disponibilizando facilidades para a manutenção e o gerenciamento dos dados criados (os objetos de projeto), e conhecimento a respeito do processo de projeto, sendo capazes de conduzir o projetista no desenvolvimento de suas tarefas [Soares, 1996]. O principal objetivo de um *framework* é potencializar o aumento da produtividade global do processo de projeto obtido através de ambientes integrados e abertos. Ele deve permitir aos usuários finais [Wagner, 1994]: (1) disparar e gerenciar a execução das ferramentas; (2) criar, organizar e gerenciar dados; (3) visualizar (graficamente) todo o processo de projeto, e; (4) realizar tarefas diversas de gerência de projeto, inclusive relativas ao trabalho em equipe.

Em [Heaton, 1992] encontram-se várias funcionalidades desejáveis para obtenção de *frameworks* que garantam ambientes de projeto realmente integrados e abertos. Dentre essas funcionalidades, duas são de grande importância: (1) *gerência de dados* e; (2) *gerência do processo de projeto*. A *tabela 1* resume os serviços fundamentais de gerência de dados e do processo de projeto.

Tabela 1. Serviços das gerências

Gerência de Dados	<i>Controle de acesso aos projetistas</i> <i>Controle de versões e configurações</i> <i>Consultas interativas aos dados dos projetos</i> <i>Organização de espaços de projetos</i>
Gerência do Processo de Projeto	<i>Execução de metodologias de projeto</i> <i>Suporte ao trabalho cooperativo</i>

Vários trabalhos relativos a *frameworks* apresentam técnicas de *documentação dos passos de projeto*, visando resolver problemas ligados às gerências de dados e do processo de projeto de engenharia/CAD, sendo um recurso usado para a manutenção de um histórico de tudo o que acontece no desenrolar do processo de projeto: quais as ferramentas utilizadas, quando, por quem, em que ordem, as opções usadas, quais os dados de entrada, quais os dados de saída, entre outros [Soares, 1996].

Uma das principais carências existentes na área de engenharia é exatamente o controle de documentação na fase de projeto. Para suprir essa carência é necessária uma aplicação que gerencie essa documentação, possuindo um controle eficiente dos dados de projeto e das atividades de projetistas. Nesse contexto, a implementação de um sistema de gerenciamento de documentação de projetos de engenharia (EDMS) deve respeitar as duas áreas já citadas: (1) *gerência de dados*, tratando da forma como os dados são acessados e o próprio gerenciamento das versões de projetos, e; (2) *gerência de projeto*, que será responsável pela administração do trabalho em equipe, controlando o fluxo de execução de tarefas.

Assim, o assunto *frameworks* não é o principal objeto em foco, e sim o gerenciamento da documentação dos passos de projetos, que possui relação direta com *frameworks*, por ser uma técnica que absorve as principais funcionalidades neles encontradas, como a gerência de dados e do processo de projeto.

3. A Gerência de Dados de Projetos Complexos

A *gerência de dados* de projetos complexos deve garantir a integridade do grande volume de dados gerado nos projetos CAD, para que projetistas possam manipulá-los de forma consistente, dentro dos ambientes em que estão inseridos. Três são os recursos fundamentais para cumprir esse requisito [Giaretta, 2001]:

- ① *Controle de acesso*: o compartilhamento da base de dados entre projetistas de equipes de projetos distintos, ou até mesmo da mesma equipe, e a hierarquia existente entre projetistas, onde projetistas-chefes são responsáveis por equipes de projeto, possuindo, portanto, tarefas específicas, leva à necessidade de um controle de acesso preciso, com perfis de usuários e níveis de acesso que garantam o bom andamento dos trabalhos e evitem acessos indesejados. A granularidade dos direitos de acesso varia muito entre os modelos encontrados na literatura. O ambiente AMPLO [Wagner, 1991], por exemplo, possui um controle relacionado às bases de dados existentes;
- ② *Gerência de versões e configurações*: a complexidade dos projetos leva a modularização destes em objetos de projeto, que, por sua vez, podem possuir várias versões, que são estados distintos do mesmo objeto alcançados em tempos distintos. A importância do gerenciamento das versões de objetos de projeto está na possibilidade de retornos a pontos anteriores, na busca de novas alternativas, a fim de alcançar o objetivo traçado. Além disso, várias versões do mesmo objeto podem ser consideradas consistentes, podendo ser utilizadas em configurações diferentes de um mesmo projeto. A área de engenharia é pioneira no estudo de versões e configurações, e vários modelos podem ser encontrados na literatura, como os modelos históricos de Chou e Kim (1986), Dittrich e Lorie (1988) e Katz (1990), além de outros, como os de Cellary, Vossen e Jomier (1991), Rykowski e Cellary (1996) e Goledziner (1995);
- ③ *Mecanismos de navegação e consulta*: projetos de engenharia tendem a ter muitos objetos de projeto, cada um com várias versões, criadas a partir de tentativas de alcançar consistência. Os projetistas responsáveis pela criação e manutenção dessas versões realizam tarefas delegadas por outros projetistas, de forma que, é necessário que exista um mecanismo para visualizar, através de uma interface gráfica e interativa, as próximas tarefas a serem executadas. Da mesma forma, versões específicas devem ser encontradas de maneira rápida e eficiente. Mecanismos de navegação e consulta realizam essas tarefas, para que projetistas possam encontrar dados desejados sequencialmente, com ordenações selecionadas, ou através de consultas específicas, atribuindo critérios de seleção.

Assim, a implementação de um sistema que gerencie essa documentação deve garantir a presença de recursos relativos à gerência de dados, como: (1) uma interface visual, com mecanismos de navegação e consulta, apresentando ao usuário formas interativas de acesso aos objetos complexos que compõem os projetos; (2) um rigoroso controle de usuários e acessos, mantendo a segurança no acesso aos dados e não deixando que ações indesejadas sejam realizadas, além de gerenciar equipes de usuários que interagem em projetos distintos, e; (3) um mecanismo de *controle de versões*, que garanta aos projetistas uma melhor organização nas mudanças ocorridas em projetos, permitindo retornar a pontos anteriores para testes em busca de melhores alternativas.

4. A Gerência do Processo de Projetos Complexos

A *gerência do processo de projeto* deve garantir o total controle das equipes de projetistas que interagem em projetos específicos. Projetos possuem metodologias de desenvolvimento, que devem ser seguidas pelos seus integrantes. Para guiar no cumprimento de tais metodologias, os seguintes recursos são utilizados:

- ① *Controle do fluxo de tarefas*: visando disciplinar as ações dos projetistas, além de guiá-los no cumprimento destas ações, o controle do fluxo de tarefas se torna imprescindível em ambientes de projeto, já que garante que as regras dos projetos sejam rigorosamente seguidas. Destacam-se dois tipos de controles: (1) os *sintáticos*, que estabelecem seqüências pré-definidas de etapas a serem seguidas, não interessando o estado do projeto, e; (2) os *semânticos*, que, dependendo da situação que o projeto se encontra, apontam o melhor caminho a ser seguido, a partir de regras pré-estabelecidas. A literatura apresenta vários modelos, como o modelo histórico de Chiueh e Katz (1990) e o esquema de tarefas do framework Odyssey [Brockman, Cobourn, Jacome, Director, 1992];
- ② *Gerência de cooperação entre projetistas*: aplicações CAD, denominadas não convencionais, possuem transações de longa duração e necessitam de suporte à cooperação entre projetistas, diferente das aplicações tradicionais. A partir desse enfoque, vários requisitos se tornam fundamentais, como os suportes à longa duração, ao trabalho em grupo, a dados complexos e à evolução de dados.

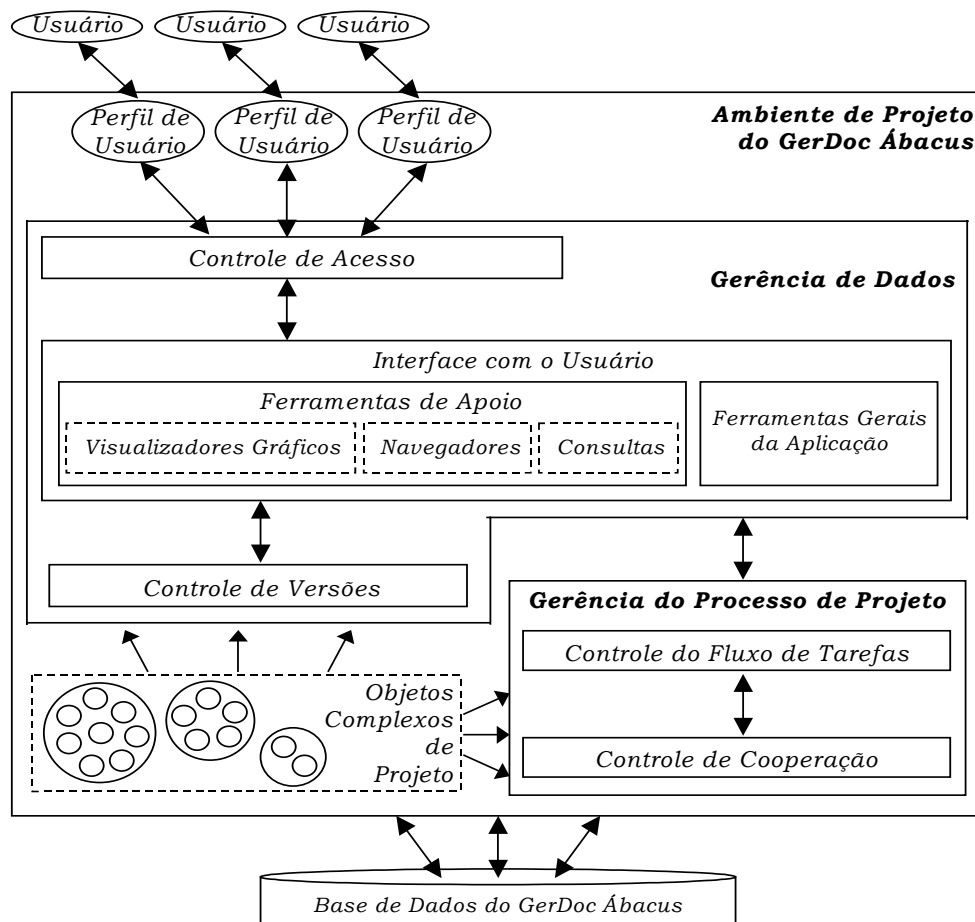


Figura 1. Modelo conceitual do GerDoc Ábacus

Dessa forma, dois mecanismos são fundamentais em sistemas que gerenciam essa documentação: (1) o controle do fluxo de tarefas de projetistas inseridos em equipes de projetos específicos, visando, além de ajudar o próprio projetista em suas ações, auxiliar o projetista-chefe a manter um controle sobre todas as atividades realizadas acerca do projeto, e; (2) a gerência de cooperação entre projetistas, incluindo os suportes ao trabalho em grupo, a dados complexos e a evolução dos dados.

5. O Modelo Conceitual do GerDoc Ábacus

O sistema gerenciador de documentação técnica para ambientes de engenharia/CAD *GerDoc Ábacus* possui a estrutura mostrada na *figura 1*, onde é possível verificar os recursos utilizados a fim de manipular os objetos complexos de projeto, na tentativa de resolver problemas relativos às gerências de dados e do processo de projeto já apresentados.

No modelo, é possível visualizar que o ambiente de trabalho do *GerDoc Ábacus* possui recursos suficientes para garantir uma certa integridade em sua base de dados. Relativo à *gerência de dados*, o *GerDoc Ábacus* apresenta recursos considerados importantes à manutenção da consistência dos dados mantidos para cada projeto.

Tabela 2. Perfis de usuários no GerDoc Ábacus

Perfil	Características e Responsabilidades
Gerente de Ambiente	- configuração geral do ambiente do sistema e sua personalização; - alteração das áreas de acesso disponíveis aos demais perfis; - possui acesso irrestrito ao sistema.
Gerente de Engenharia	- criação de novos projetos e definição dos grupos de usuários participantes e o perfil de cada um. - construção do fluxo de tarefas a ser executado em projetos.
Engenheiro de Projeto	- é o próprio projetista, responsável pela execução de tarefas determinadas pelo Gerente de Engenharia.
Consultor	- não possui acesso à execução de tarefas dentro do ambiente do sistema. Pode apenas visualizar a situação de projetos.

O primeiro deles é o *controle de acesso*, que é baseado em *usuários com perfis* diferentes (*tabela 2*), identificados a partir da função exercida por cada projetista dentro da equipe que ele faz parte, possuindo *áreas de acesso* distintas, controlando acessos indesejados em áreas impróprias. Esses perfis são distribuídos aos usuários através de *grupos de usuários*, segundo o envolvimento de cada um nos projetos de engenharia existentes. Esse mecanismo de acesso possui um nível largo de granularidade, pois define direitos de acesso para grupos de usuários dentro de projetos como um todo.

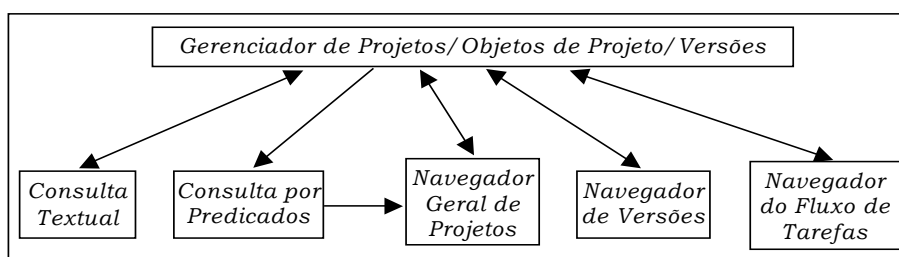


Figura 2. Interface gráfica do GerDoc Ábacus

Os *Mecanismos de Navegação e Consulta* são disponibilizados através do ambiente gráfico do *GerDoc Ábacus*, mostrado na *figura 2*. O principal navegador

existente possibilita visualizar listas sequenciais de dados de projetos, ordenados a partir de várias formas de classificação. Há ainda o navegador de versões, que mostra o grafo de versões de determinado objeto de projeto, e o de fluxo de tarefas, que permite visualizar as próximas tarefas a serem executadas em um objeto de projeto específico. Em relação às consultas, além de uma consulta textual simples, o *GerDoc Ábacus* apresenta uma consulta por predicados, onde o usuário cria e executa suas próprias sentenças de seleção de dados.

Em relação à *Gerência de Versões de Objetos de Projeto*, os dados são armazenados na forma $\langle \text{projeto}, \text{objeto de projeto}, \text{versão do objeto de projeto} \rangle$, onde cada objeto de projeto criado é considerado um objeto versionado. Cada versão criada possui um *documento CAD* associado e pode possuir versões sucessoras. Um *status* indica seu estado atual e apenas uma delas é considerada *versão corrente* do objeto de projeto. A *figura 3* mostra a representação das versões de objeto de projeto seguida pelo *GerDoc Ábacus*.

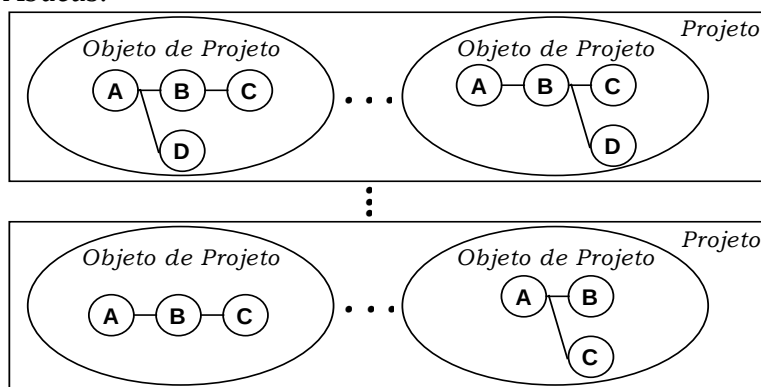


Figura 3. Representação das versões de objeto de projeto

Objetivando facilitar o versionamento de objetos de projeto e apresentar formas de manter a integridade entre as versões, o modelo de versões do *GerDoc Ábacus* engloba as características a seguir descritas, tendo a maioria delas baseada no modelo de Goledziner (1995):

- ① Cada projeto é composto por vários objetos de projeto que, por necessidade de modificações durante seu ciclo de vida, podem ser versionados até atingirem um nível satisfatório;
- ② Um objeto de projeto é considerado um *objeto versionado*, mantendo um grafo de derivação acessível ao usuário, para que seja possível visualizar sua situação nos diversos estágios de desenvolvimento. No momento da criação de um objeto de projeto, dados referentes a ele são armazenados e, automaticamente, é criada a sua primeira versão (A), que se torna o ponto inicial do desenvolvimento do objeto, tendo anexado um documento inicial, que poderá ser versionado posteriormente;
- ③ Cada objeto versionado possui uma versão que é considerada *versão corrente*. O usuário pode especificar a versão corrente que, por *default*, é mantida automaticamente como a mais recentemente criada. No processo de concepção de um objeto de projeto, várias versões podem surgir e a última versão criada se tornará a versão corrente se o usuário não alterar essa propriedade. Assim, em boa parte dos casos, a versão corrente pode não ser a versão final (considerada consistente) do objeto de projeto, se assim for definido pelo usuário.

- ④ Uma versão pode ter várias antecessoras e várias sucessoras. No momento em que uma versão é criada como sucessora de uma já existente, ela passa a possuir seus mesmos dados e, automaticamente, uma cópia do documento CAD é feita.
- ⑤ Cada versão de um documento possui um *status*, que reflete seu estágio de desenvolvimento, podendo ser: (1) *em trabalho*, versão temporária que deve sofrer diversas alterações; (2) *estável*, versão que já atingiu um estado mais consistente, podendo ser compartilhada e não podendo sofrer alterações, ou; (3) *consolidada*, versão no seu estágio final, não podendo mais ser alterada ou removida. No momento da criação, uma versão recebe o *status em trabalho*. As versões utilizadas como antecessoras de novas versões passam automaticamente para *estáveis*, impedindo que alterações sejam feitas em versões-base de outras. Essas promoções automáticas não significam que o usuário não possa realizar as suas próprias promoções, explicitamente, podendo passar uma versão *em trabalho* para *estável* e uma versão *estável* para *consolidada*.

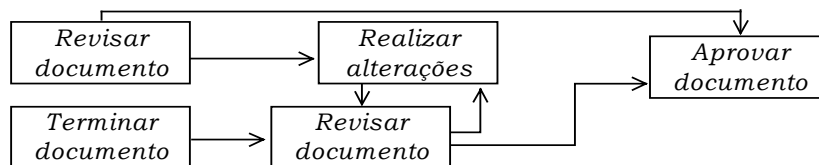


Figura 4. Exemplo de modelo de fluxo de tarefas

Relativo a *Gerência do Processo de Projeto*, o GerDoc Ábacus possui mecanismos para controlar o trabalho cooperativo dos projetistas envolvidos em projetos específicos, como o *controle do fluxo de tarefas* de objetos de projeto, que se trata de um mecanismo meramente *sintático*, guiando projetistas inseridos em projetos nos passos a serem seguidos durante o ciclo de vida de cada objeto de projeto criado. Cada projeto pode ter um *modelo de fluxo de tarefas* distinto, e cada objeto de projeto seguirá o modelo relacionado ao projeto em que está inserido. O GerDoc Ábacus exige que seja seguido o fluxo pré-determinado de tarefas. A *figura 4* apresenta um exemplo de fluxo de tarefas.

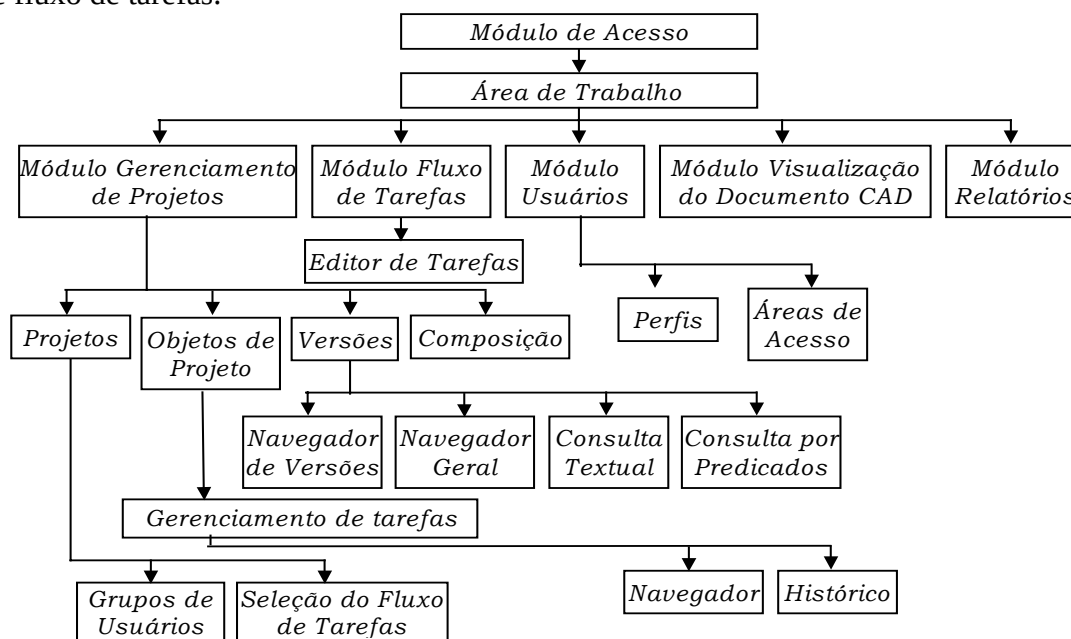


Figura 5. Módulos componentes do GerDoc Ábacus

6. Aplicação do Modelo Conceitual na Interface do GerDoc Ábacus

A interface do sistema gerenciador de documentação técnica para ambientes de engenharia/CAD *GerDoc Ábacus* segue explicitamente o modelo conceitual apresentado. Cada módulo componente possui seu funcionamento calcado nos conceitos do modelo específico que justifica sua existência. Os módulos existentes e a interação entre eles são mostrados na *figura 5*.

O ambiente do *GerDoc Ábacus* pode ser visualizado na *figura 6*. Observa-se na parte superior a *barra de menu* e a *barra de ferramentas*, na parte direita o *visualizador de documentos CAD* e na parte esquerda o *gerenciador de projetos*, que objetiva visualizar a forma de operacionalização real do sistema em termos da organização de sua base de dados na forma <projeto, objeto de projeto, versão do objeto de projeto>, da interação dos usuários quanto aos seus direitos de acesso e da execução do fluxo de tarefas, sendo dividido em quatro gerenciamentos: (1) *de projetos*; (2) *de objetos de projeto*; (3) *das versões dos objetos de projeto*, e; (4) *das versões de objeto de projeto componentes*.

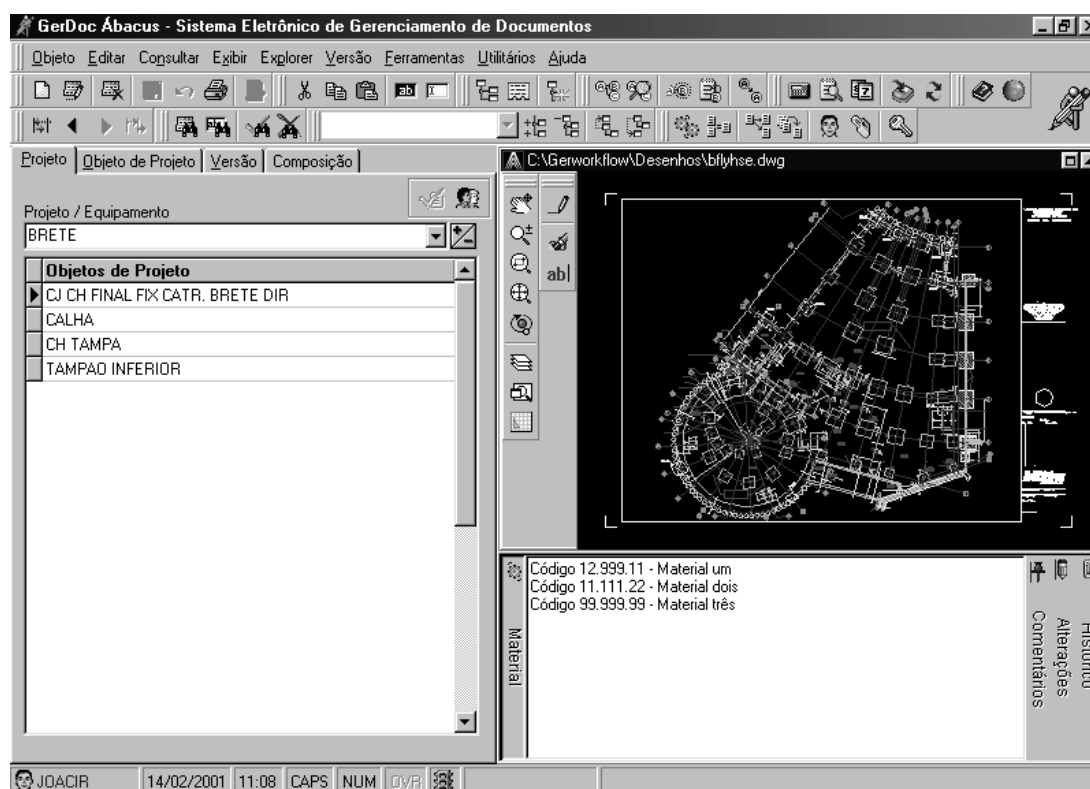


Figura 6. Visão geral do GerDoc Ábacus

Em primeira instância, *projetos* devem ser criados para receberem *objetos de projetos* com suas respectivas *versões*. A criação de novos projetos, que é uma tarefa com exclusividade de execução a usuários com o perfil gerentes de engenharia, inclui a montagem do grupo de usuários que interagirá com o projeto e a determinação do modelo de fluxo de tarefas que será seguido em cada objeto de projeto a ser criado posteriormente, como mostra a *figura 7*.

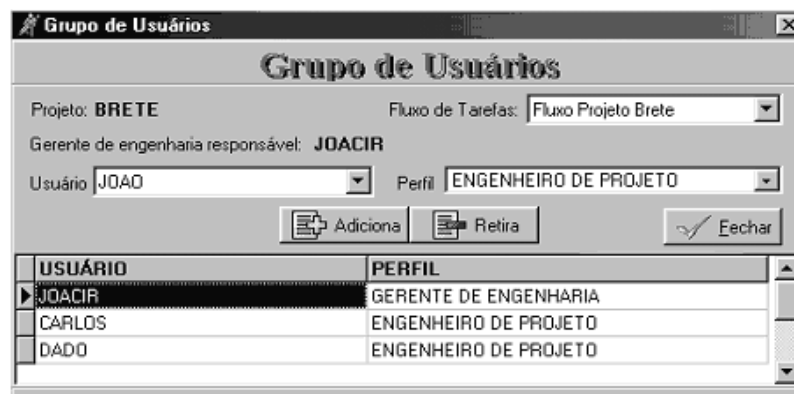


Figura 7. Criação do grupo de usuários de um projeto específico

Os objetos de projeto são criados a partir do gerenciador de objetos de projeto. No instante que um objeto de projeto é criado, sua versão inicial também é criada. É necessário, portanto, que o *objeto versionado* seja cadastrado juntamente com a sua primeira versão, que possuirá o documento CAD associado. Não é possível cadastrar um objeto de projeto sem ter uma versão associada. A *figura 8* mostra os dois gerenciadores.

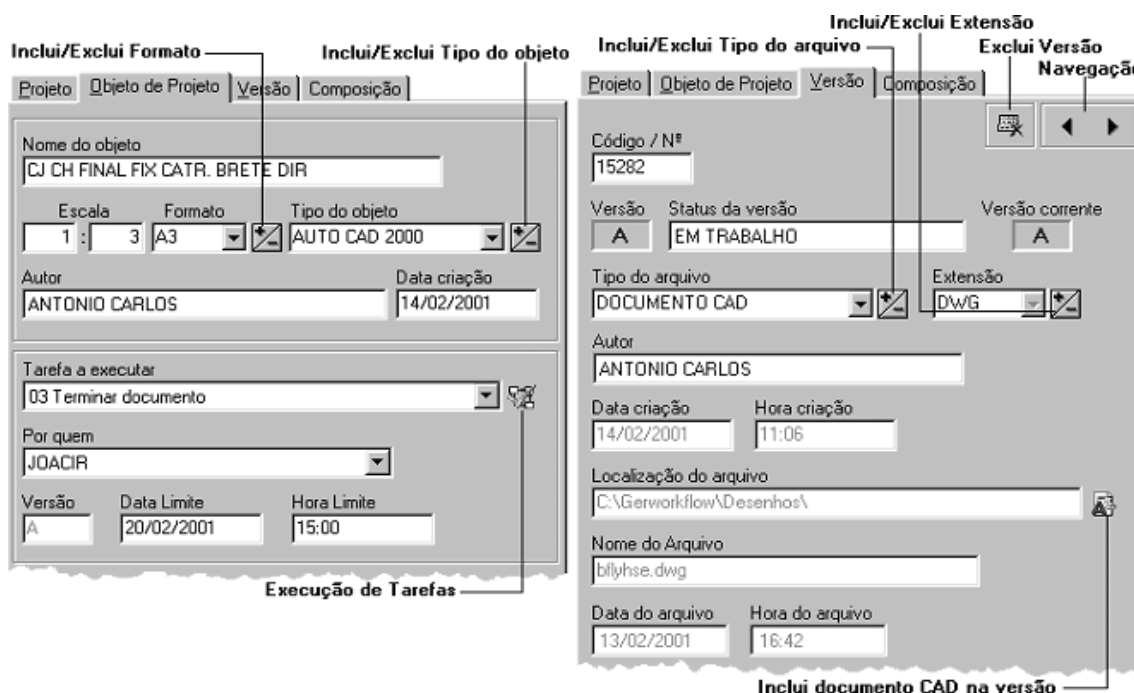


Figura 8. Gerenciador de objetos de projeto e de versões de objetos de projeto

O mecanismo sintático de gerenciamento de tarefas é mantido na mesma aba do gerenciador de objetos de projeto, pois cada objeto criado terá o seu próprio fluxo de tarefas particular, baseado no modelo de fluxo de tarefas determinado na criação do projeto. A tarefa a ser executada é visualizada, indicando o usuário responsável pela execução, a versão-alvo, a data e a hora limite da operação. A realização das tarefas e a determinação da próxima tarefa a ser executada, além da visualização do fluxo de tarefas do projeto são feitas em telas próprias, como mostra a *figura 9*.

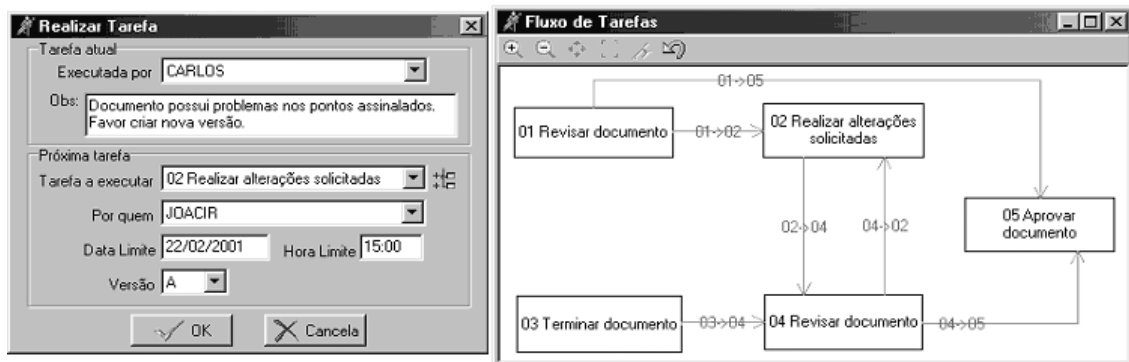


Figura 9. Execução de tarefas e navegador do fluxo de tarefas

É possível, a qualquer momento, visualizar o grafo de versões do objeto de projeto que está sendo trabalhado, através do *navegador de versões*, conforme *figura 10*. Graficamente, o *GerDoc Ábacus* permite percorrer o grafo e visualizar os principais dados das versões, além do próprio documento CAD. Através de um duplo clique do mouse na versão desejada, é possível passar o controle para o gerenciador de versões de objetos de projeto, a fins de alterações ou consultas mais detalhadas.

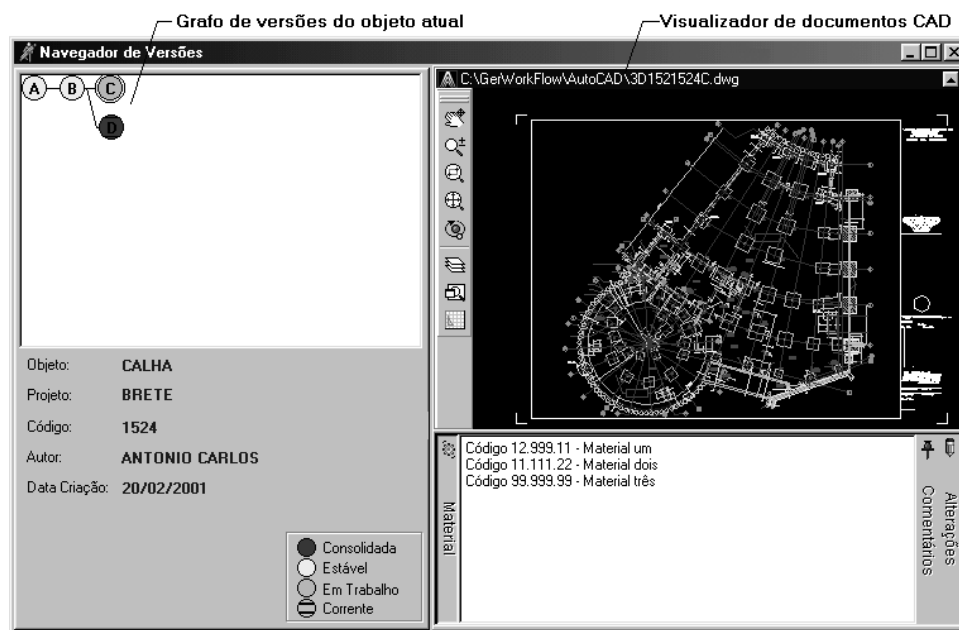


Figura 10. Navegador de versões

Referente aos módulos de navegação e consulta, a interface gráfica apresentada pelo *GerDoc Ábacus* permite que vários mecanismos de navegação e consulta estejam acessíveis aos usuários-projetistas durante a execução de seus trabalhos, objetivando facilitar a interação entre os grupos de projetistas e os diversos projetos existentes.

O *GerDoc Ábacus* possui uma preocupação especial nesse sentido, pois determinadas situações exigem tratamento especial dos dados de cada projeto. Por exemplo, usuários-projetistas precisam saber, diariamente e de forma rápida, quais tarefas são destinadas a ele. O *navegador geral* realiza essa operação, mostrando, de forma gráfica, a lista de tarefas correspondente. Em outro exemplo, um gerente de engenharia responsável por um projeto pode querer saber quais foram os objetos de projeto que determinado usuário trabalhou em um determinado período de tempo. Através da *consulta por predicados* e do *navegador geral* isso é possível. A *consulta*

por predicados apresenta três elementos, conforme mostra a figura 11: (1) os atributos, situados na parte esquerda; (2) os operadores, situados em botões, e; (3) os valores referenciais, situados no canto superior direito.

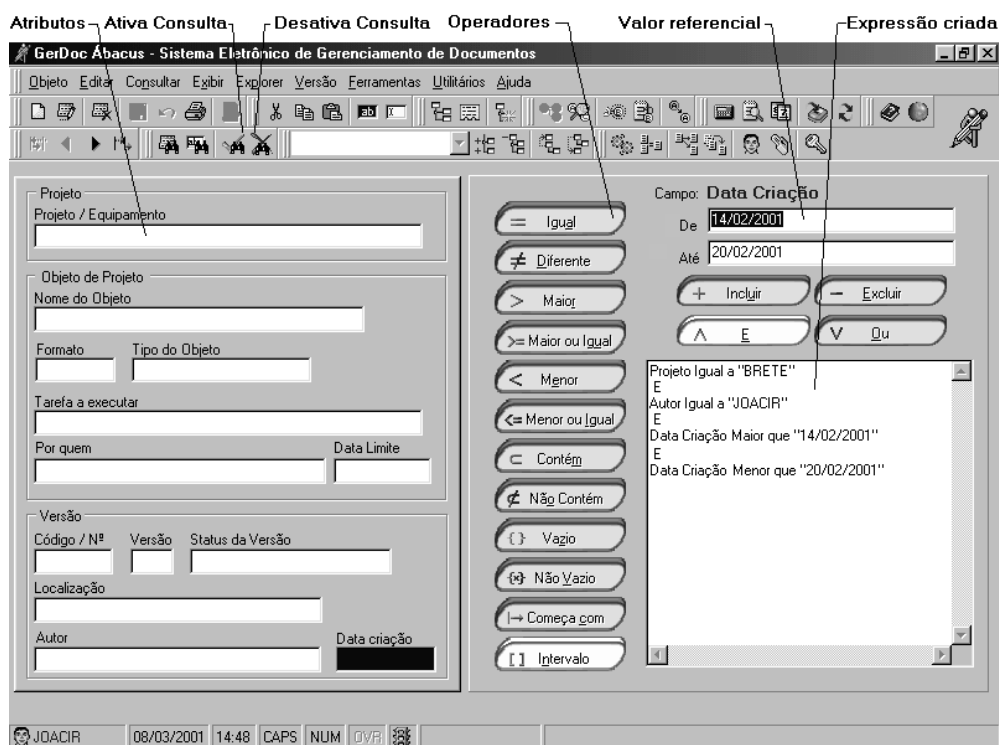


Figura 11. Consulta por predicados

O resultado das consultas feitas através de predicados é mostrado *navegador geral*, onde os dados são apresentados no estilo *explorer*, permitindo abrir e fechar itens título de conjuntos de dados, de uma maneira gráfico-interativa. Conforme mostra a figura 12, o usuário pode selecionar o modo de exibição desejado através de lista própria, sendo que os modos existentes são: (1) Árvore de Diretórios; (2) Autor/Versão; (3) Projeto/Autor; (4) Tipo de Arquivo; (5) Por Data; (6) Lista de Tarefas.

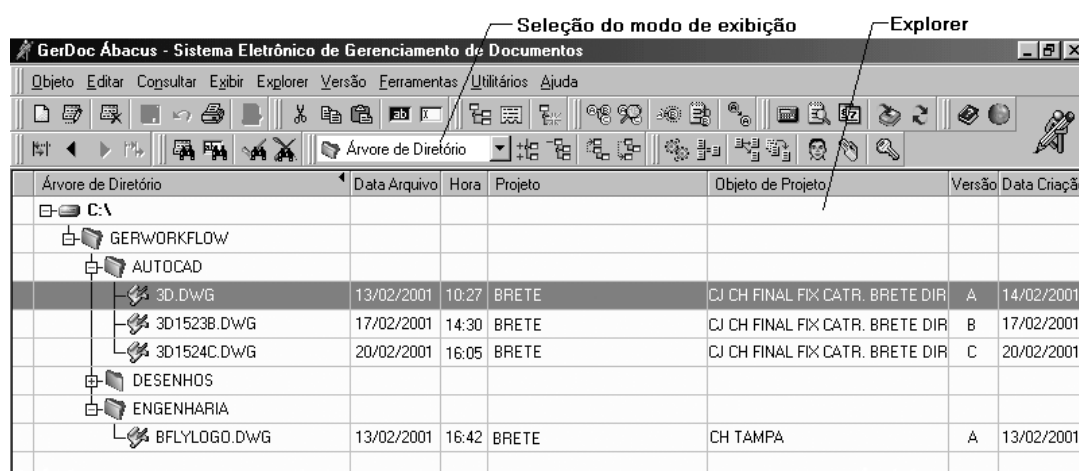


Figura 12. Navegador geral - Árvore de diretórios

7. Conclusões e Trabalhos Futuros

O sistema gerenciador de documentação técnica para ambientes de engenharia/CAD *GerDoc Ábacus* une recursos considerados importantes para garantir a integridade de dados e o trabalho harmonioso entre projetistas que interagem em projetos específicos. Tais recursos são bem definidos, e fazem parte de duas categorias de gerência: a *de dados* e a *do processo de projeto*. Objetivando garantir a integridade dos dados dos projetos cadastrados, os recursos de *gerência de dados* existentes no *GerDoc Ábacus* seguem os exigidos para alcançar bons níveis de consistência. Integrado aos recursos de gerência de dados, os recursos de *gerência do processo de projeto* objetivam garantir total organização no trabalho cooperativo existente entre projetistas. Os recursos existentes apresentam um nível satisfatório de controle alcançado, tornando o *GerDoc Ábacus* uma ferramenta de grande utilidade no controle da documentação dos passos de projetos de engenharia.

A constatação final acerca do *GerDoc Ábacus* é que ele é um sistema que apresenta um bom nível de integridade e consistência nos serviços prestados e, se bem aplicado em algum ambiente de engenharia, pode contribuir muito na organização e manutenção dos dados relativos aos projetos e no controle das tarefas realizadas pelas equipes de engenheiros inseridas nesse ambiente.

Apesar disso, o *GerDoc Ábacus* poderia apresentar alguns recursos adicionais. O principal é o fornecimento automático da composição das versões de objeto de projeto. Duas ferramentas podem ser inseridas no momento que esse recurso se tornar automático, a ferramenta *documentos relacionados*, que permite visualizar as versões de objeto de projeto na qual a que está sendo pesquisada está inserida, e a ferramenta *partes da montagem*, que permite visualizar a própria composição da versão que está sendo pesquisada.

Outra ferramenta de auxílio que poderia ser inserida no ambiente do *GerDoc Ábacus* é a *comparação de dois documentos*, permitindo comparar versões diferentes de um mesmo objeto de projeto, para verificar inserções e exclusões de itens. Para isso, é necessário adquirir algum dos produtos existentes no mercado que permita realizar essa operação e inserir no *GerDoc Ábacus*.

Referências Bibliográficas

- Baldam, R.; Valle, R.; Cavalcanti, M. **GED: Gerenciamento Eletrônico de Documentos**. São Paulo: Érica, 2002. 204p.
- Baldam, R. L. **Gerenciamento Eletrônico de Documentos Técnicos em Departamentos de Engenharia de Projeto e Manutenção do Setor Siderúrgico**. Rio de Janeiro: UFRJ, 2003. 217p. Dissertação de Mestrado.
- Bosch, K. O.; Bingley, P.; Wolf, P. Design Flow Management in the NELSIS CAD Framework. In: ACM/IEEE Design Automation Conference, 28., 1991, San Francisco. **Proceedings...** New York: ACM, 1991. v.11, p.711-716.
- Brockman, J. B.; Cobourn, T. F.; Jacome, M. F.; Director, S. W. **The Odyssey CAD Framework**. In: IEEE DATC Newsletter on Design Automation, IEEE, spring 1992.

- Cellary, W.; Vossen, G.; Jomier, G. **Multiversion Object Constellations for CAD Databases**. Germany: Division of Computer Science, University of Giessen, 1991. (Techn. Report 9105).
- Chiueh, T.; Katz, R. H. A History Model for Managing the VLSI Design Process. In: IEEE International Conference On Computer-Aided Design, 1990, Santa Clara. **Digest of Technical Papers**, [S.l.:s.n.], 1990. p. 358-361.
- Chou, H. T.; Kim, W. A Unifying Framework for Version Control in a CAD Environment. In: International Conference On Very Large Data Bases, 12., 1986, Kyoto, Japão. **Proceedings...** [S.l.:s.n.], 1986. p. 336-344.
- Dittrich, K. R.; Lorie, R. A. Version Support for Engineering Database Systems. **IEEE Transactions on Software Engineering**, New York, v.14, n.4, p.429-437, april 1988.
- Giaretta, J. **Gerenciamento de Documentação Técnica para Ambientes de Engenharia/CAD com Suporte a Versões**. Porto Alegre: PPGC da UFRGS, 2001. 90p. Dissertação de Mestrado.
- Goledziner, L. G. **Um modelo de versões para banco de dados orientados a objetos**. Porto Alegre: CPGCC da UFRGS, 1995. 147p. Tese de Doutorado.
- Heaton, J. The JESSI-Common-Frame Project – SP4 Evaluation: Evaluating the Options. In: IFIP WG 10.2/WG 10.5 Workshop On Electronic Design Automation Frameworks, 3., 1992. **Proceedings...** Amsterdam: North-Holland, 1992. p.271-288.
- Katz, R. Toward a Unified Framework for Version Modeling in Engineering Databases. **ACM Computing Surveys**, New York, v.22, n.4, p.375-408, Dec. 1990.
- Kathöfer, T.; Miller, J. The JESSI-Common-Frame Project – Sub-project Development. In: IFIP WG 10.2/WG 10.5 Workshop On Electronic Design Automation Frameworks, 3., 1992. **Proceedings**: Amsterdam: North-Holland, 1992. p.253-269.
- Peltonen, H.; Männistö, T.; Alho, K; Sulonen, R. An Engineering Document Management System. In: ASME Winter Annual Meeting. New Orleans, Lousiana, 1993.
- Rykowski, J.; Cellary, W. Using Multiversion Object-Oriented Databases in CAD/CIM Systems. In: International Databases And Expert Systems Applications (DEXA), 7., 1996, Zurich, Switzerland. **Proceedings...** Zurich: [s.n.], 1996. p.183-190.
- Soares, S. N. **Sistema Gerenciador de Documentação de Projeto**. Porto Alegre: CPGCC da UFRGS, 1996. Dissertação de Mestrado.
- Wagner, F. R. Data Management Facilities in the AMPLO Design Framework. In: IFIP Workshop On Electronic Design Automation Frameworks, 1991. **Proceedings...** Amsterdam: North-Holland, 1991. p.71-84.
- Wagner, F. R. et al. STAR – Um ambiente para a Integração de Ferramentas de Projeto de Sistemas Digitais. In: Congresso Da Sociedade Brasileira De Microeletrônica, 6., 1991, Belo Horizonte. **Anais...** Belo Horizonte: [s.n.], 1991. p.176-185.
- Wagner, Flávio Rech. Ambientes de Projeto de Sistemas Eletrônicos. In: Escola De Computação, 9., 1994, Recife, Brasil. **Anais...** Recife: UFPE, 1994. 155p.