

ESTUDO COMPARATIVO DE METODOLOGIAS DE DESENVOLVIMENTO DE SISTEMAS ESPECIALISTAS

Douglas Yeong Ping Hwang *

AKROP106.PM5

Resumo

Atualmente, as áreas comerciais e industriais estão tendo lucros significativos na utilização da Tecnologia de Sistemas Especialistas. Este trabalho tem como objetivo estudar metodologias para o desenvolvimento de Sistemas Especialistas e apontar suas diferenças em relação ao desenvolvimento de Sistemas Convencionais. Ao final, será feita uma breve comparação entre as metodologias estudadas.

Abstract

Nowadays, the expert system technology applied to commercial and industrial activities has generated significant gains for the users. The purpose of this paper is to study development expert system methodologies. It will be showed the main differences between expert system and conventional system development. At the end it will be compared in few words the methodologies seen.

Introdução

A Inteligência Aplicada (IA) nasceu como uma atividade acadêmica sem finalidade comercial imediata. Mas, com os bons resultados que os programas de IA têm dado nas áreas comerciais e industriais, ela tornou-se um tema de crescente importância. A meta dos cientistas da computação, que trabalham neste campo, é a produção de programas que imitem o desempenho humano em uma ampla variedade de tarefas “inteligentes” [You92].

Uma área já bem conhecida da IA é aquela que trata dos chamados *Sistemas Especialistas* (SE) ou *Sistemas Baseados no Conhecimento* (SBC). Tais sistemas foram criados para solucionar problemas que apenas especialistas humanos eram capazes de resolver. SE e SBC são programas que possuem conhecimento e capacidades embutidas para agirem como especialistas. Na maioria das vezes, são capazes de explicar seus passos até chegarem a uma solução, ou então explicar porque rejeitaram certas linhas de raciocínio e escolheram outras.

Os SE são desenvolvidos para solucionarem problemas não convencionais ou que sejam difíceis de serem *algoritmizados*. Eles se destacam nas áreas de auditoria, diagnóstico de falhas, configuração de computadores, consultoria

* Bacharelado em Ciência da Computação.

financeira, dentre outras. Por se tratar de problemas de risco e ter pouca previsão de um resultado final satisfatório, apenas empresas de grande porte tinham condições suficientes para darem suporte a essas pesquisas. Mas, com os primeiros resultados e os fundamentos estabelecidos por elas, as médias e pequenas empresas adquiriram condições mais seguras e favoráveis para se lançarem nesta nova tecnologia [Kel91].

Um projeto de sistema baseado no conhecimento deve ser administrado como qualquer outro projeto de sistema, uma vez que exige considerações sobre especificação, orçamento e tempo. Assim, estratégias e metodologias adequadas são necessárias na construção de sistemas que se utilizem da IA.

O propósito deste trabalho é discutir uma metodologia para o desenvolvimento de SE, comentando as mudanças necessárias que são introduzidas nas atividades em relação ao desenvolvimento de sistemas convencionais.

1. A Arquitetura de um Sistema Especialista

Um Sistema Especialista é constituído de três elementos básicos: a base de conhecimento, o mecanismo de raciocínio e a *interface* homem máquina.

1.1. A Base de Conhecimento

Para poder solucionar problemas complexos, é preciso que o SE possua uma grande quantidade de conhecimento sobre o domínio do problema e mecanismos que manipulem esses conhecimentos. Esse conjunto de informações fica armazenado em uma **Base de Conhecimento**, que se responsabiliza pelo correto uso dessas informações. E são estas informações que dão a característica de um sistema ser ou não especialista no domínio em questão.

Ao se discutir a Representação de Conhecimento, na base de conhecimento, lidamos com duas entidades diferentes:

Os fatos: que são as verdades do mundo real que se quer representar.

Os mecanismos de representação: que são as formas como os fatos serão representados. São essas formas que identificam o conhecimento que realmente se pode manipular.

Há diferenças importantes entre sistemas convencionais e SE. Estas diferenças se encontram nos conceitos de certeza, completude e dinamismo [Kel91]. Em sistemas convencionais, usam-se informações apresentadas de forma **determinística**, onde há segurança de que, se as condições forem verdadeiras, então a conclusão será verdadeira. Por outro lado, as informações (conhecimento) de SE podem não ser absolutamente verdadeiras, sendo melhores descritas como diretrizes ou normas práticas. Tal conhecimento é **heurístico**, constituindo-se em um conjunto de informações, que se adquire ao longo da experiência e que pode ser aplicado ao se tomarem decisões em situações similares mas não exatamente iguais.

Existem várias formas de representação de conhecimento, dentre elas, as que têm dado bons resultados, são:

- Representação Lógica, onde deduzimos novas informações a partir de fatos conhecidos;
- Representação Procedimental, onde as informações são representadas em forma de uma seqüências de transformações, codificadas em procedimentos;
- Sistemas de Produção: o conhecimento é codificado em regras da forma: *se*, condição *então*, ação;
- Redes Semânticas e Frames: são formas gráficas de representação de conhecimento;
- Scripts ou Roteiros, são estruturas que descrevem uma seqüência estereotipada de eventos em um determinado contexto [RK93].

1.2. O Mecanismo de Raciocínio ou Inferência

O mecanismo de raciocínio é o mecanismo da linguagem que leva o conhecimento implícito através do raciocínio sobre as informações da base. O mecanismo de inferência envolve diferentes processos que devem trabalhar em conjunto [Kel91]:

- 1) Identificar a regra como relevante à situação (premissa) do problema. Esse passo pode resultar na identificação de mais de uma regra da base de conhecimento;
- 2) Resolver o conflito entre regras, resultando na seleção de uma apenas;
- 3) Executar a regra. Isto é, podemos chegar à conclusão, partindo de suas premissas.

Caso ocorra um conflito, é preciso um mecanismo de Resolução de Conflitos, que é o processo de decisão sobre a seqüência no uso das regras. Qualquer técnica para a resolução de conflitos deve ser realmente *aplicável ao domínio de conhecimento em que estivermos trabalhando* [Kel91].

A maneira mais comum de utilizarmos regras para chegar a uma conclusão é o encadeamento de regras (Rule Chaining). Existem dois tipos básicos de protocolos de encadeamento de regras: *encadeamento para frente (forward chaining)* e *encadeamento para trás (backward chaining)*. Sistemas executando encadeamento para trás são melhores em situações onde se aceita uma conclusão como verdadeira e, através das regras, chega-se à prova dessa conclusão. Já os sistemas executando encadeamento para frente são utilizados quando se conhecem as premissas de uma situação e, por meio do encadeamento de regras, obtém-se uma conclusão.

1.3. A Interface Homem-Máquina

Todos os problemas relacionados com o usuário em programas convencionais são tratados da mesma forma em um sistema especialista. É importante dar mais atenção à força e à flexibilidade da interface com o usuário final [SMS92].

O tipo de usuário do sistema define as capacidades de sua interface. Um empregado, um especialista de domínio, um programador tradicional e um profissional em IA, cada um tem necessidades e prioridades diferentes em relação à interação com o sistema.

Algumas *Shells* oferecem ao usuário capacidade para interagir com o sistema especialista através de menus, com instruções simples e diretas. Todos os *displays* devem ser compreendidos pelo usuário, sendo simples e atrativos. O usuário deverá ser capaz de interromper o sistema, salvando o programa, e recomeçar do ponto da interrupção.

O sistema deverá prover um ambiente simples, passo a passo, para novatos, contendo uma fácil compreensão, *help* sensível ao contexto, etc. Por outro lado, para usuários especialistas, o sistema deve ter macros, passos combinados, usando caminho mais curto e capacidade para seleção de menus. Em um sistema, são importantes também mensagens amigáveis de erros e métodos alternados de entradas. A habilidade para mudar respostas sem ter que recomeçar a sessão é também muito importante para projetistas e usuários. Uma interface ideal deveria ser capaz de capturar e mostrar imagens gráficas [SMS92].

A *interface* em linguagem natural é uma tecnologia que deve revolucionar a parte de interface com o usuário final do sistema. A linguagem natural tem como objetivo permitir ao usuário *falar* com o computador em uma linguagem humana e fazer com que o computador responda na mesma língua [LDB88], possibilitando ao usuário ser tão vago ou preciso

quanto achar necessário. Isto significa que o usuário pode deixar de fornecer informações que intuitivamente julga ser do conhecimento de um computador.

2. O Desenvolvimento de Sistemas Convencionais

O desenvolvimento de sistemas convencionais é composto por algumas atividades essenciais. Essas atividades serão descritas sucintamente a seguir.

2.1 O Levantamento ou estudo de viabilidade.

Esta atividade inicia-se quando o usuário pretende automatizar partes de sua atividade. Seus objetivos principais são:

- a) identificar os usuários responsáveis e desenvolver um “escopo” inicial do sistema [You92]. Isso pode ser realizado através de entrevistas com os usuários envolvidos e pode gerar um diagrama de fluxo de dados (DFD) simples;
- b) especificar as deficiências do sistema atual;
- c) estabelecer metas e objetivos para um novo sistema;
- d) determinar a possibilidade de automação do sistema atual e
- e) preparar uma previsão do projeto que será usada para conduzir o restante do projeto.

2.2 A Análise do sistema.

Visa criar uma *especificação* dos resultados obtidos no levantamento. Essa atividade é uma das mais importantes de todo o desenvolvimento de sistema, sendo crucial para sua correção e confiabilidade.

2.3 O Projeto.

É uma atividade de alto nível. Ocupa-se em organizar o sistema em subsistemas e alocar esses subsistemas aos componentes de hardware e software [Rum94]. Visa, também, escolher uma estratégia para a transformação do modelo de dados em estruturas de dados, arquivos e bancos de dados. Preocupa-se com a interface homem-máquina em que o usuário será submetido pelo sistema.

2.4 A Implementação.

Responsabiliza-se pela codificação e integração dos módulos para uma primeira instância do sistema final.

2.5 A Geração de Testes.

Verifica se os resultados do sistema estão de acordo com as especificações do usuário. O propósito é garantir que o sistema apresente um nível apropriado de qualidade. Desse modo é preciso que os testes sejam realizados em todas as atividades iniciais para alcançarmos o nível desejado de qualidade.

2.6 A Descrição dos Procedimentos.

Preocupa-se em gerar uma descrição formal das atividades manuais e das atividades automatizadas do novo sistema. O seu resultado será um *manual do usuário*.

3. O Desenvolvimento de Sistemas Especialistas

O desenvolvimento de sistemas, baseados no conhecimento, utiliza a mesma metodologia dos sistemas convencionais. Mas é preciso fazer mudanças em algumas das atividades e acrescentar outras atividades relevantes ao desenvolvimento de um sistema especialista.

3.1 O Levantamento

O Levantamento, em um projeto de SE, pode requerer mais tempo do que nos desenvolvimentos de sistemas convencionais. Isso ocorre devido à necessidade de se reverem as várias aplicações possíveis dessa tecnologia, antes de começar alguma. Ela deve fornecer algumas metas gerais e estimativas de custo do projeto.

O levantamento deve propor estratégias de curto e longo prazo para uma suave integração da tecnologia de IA. Ele deve determinar quais domínios são apropriados para a aplicação da IA e, através desses domínios, preparar uma proposta detalhada para executar um projeto de desenvolvimento, identificando áreas onde a IA forneceria benefícios substanciais em termos de aumento de rendimentos, redução de custos, etc. Dentro de cada domínio de aplicação o relatório deve conter:

- Uma visão geral da importância dos sistemas baseados no conhecimento (DFD's);
- Uma análise de custo/benefício da aplicação do domínio;
- Uma análise de risco de desenvolvimento proposto [Kel91].

O estudo de viabilidade analisa três importantes fatores que são: Valor, Risco e Cultura Corporativa.

O Valor:

Os projetos de SE procuram automatizar as tarefas no nível de decisão profissional e de planejamento, visando maior rapidez ou segurança nas tarefas que serão executadas ou na habilidade de se executar corretamente as tarefas. A aplicação, baseada no conhecimento, tende a considerar problemas que não são tratados previamente, por causa da dificuldade de manuseá-los com métodos tradicionais, exigindo profissionais especializados

e caros. Os sistemas baseados no conhecimento têm o mesmo objetivo dos sistemas convencionais: *o lucro*. Mas os tipos de tarefas e a magnitude, tanto do retorno como do custo, tendem a ser diferentes dos projetos de sistemas convencionais.

O Risco:

O Risco significa a probabilidade de não ser capaz de completar o projeto até um ponto onde os resultados justifiquem os custos até aquele período [Kel91]. Os principais fatores que levam ao risco, podem ser: a complexidade da base de conhecimento, a complexidade da técnica de inferência e a disponibilidade contínua dos especialistas de domínio. Em qualquer área, de uma possível aplicação de SE, os fatores ligados à natureza do conhecimento envolvem confiabilidade, integridade, ambigüidade e estabilidade. Os SE têm uma melhor aplicação em ambientes onde os dados são, de certa forma, incompletos, não confiáveis, ambíguos e dinâmicos. Mas os SE têm dificuldades em resolver problemas onde o especialista utiliza *porções* de bom senso e intuição, isto é, os SE ainda não possuem mecanismos para “raciocinar” de forma não dedutiva.

A Cultura Corporativa:

Uma “cultura organizacional” significa estruturas, valores e componentes de um grupo, isto é, uma empresa. Os resultados esperados na construção de um sistema especialista devem ser consistentes com uma cultura corporativa particular. A empresa deve ter uma diretriz em mente: *escolha de aplicações para as quais a aceitação do sistema pelos funcionários seja positiva* [Kel91]. Quando o conhecimento especializado é muito difícil de aprendizagem, a aplicação de sistema especialista é contra-indicada. Por outro lado, situações onde existe alta rotatividade de especialistas, como corretores de

títulos, tais aplicações são mais adequadas aos SE.

3.2 A Análise de Sistema Especialista

Uma vez escolhida a aplicação, é hora de especificar completamente os requisitos do usuário para a forma final do sistema. Os primeiros resultados de uma análise de IA são: especificação do sistema e um protótipo da base de conhecimento.

A especificação representa o processo analítico em que visualizamos o completo sistema especialista. Um sistema baseado no conhecimento, independente do tamanho, é muito difícil de se compreender como um todo. Assim, a Análise divide esta grande tarefa em mini tarefas ou mini-sistemas mais fáceis de se especificar e implementar.

A especificação do novo sistema é um processo em dois estágios. O primeiro, é adicionar qualquer funcionalidade ou dados novos necessários ao novo sistema. Ou, caso não existam funções novas, detalhar o que existe, tornando mais claro possível sobre o que desejamos fazer. A especificação descreve completamente o sistema final e entra para a Atividade de Projeto.

Em todo sistema é necessária uma documentação completa, não ambígua e não redundante, do que está sendo feito no momento, de maneira que as mudanças possam ser integradas facilmente. Nos SE, o enfoque principal é captar o que o especialista faz no momento e simplesmente automatizar esse processo. Uma diferença importante em relação aos sistemas convencionais, é que os SE tentam automatizar processos humanos nos quais há poucas ações observáveis, ou seja, um analista procura caracterizar a ação de raciocinar do especialista que é de difícil observação e descrição.

Segundo Irgon [IZMB90], a documentação descrevendo a identificação do problema e da solução constitui o mais importante requisito para um projeto de sistema especialista.

Existem ferramentas que permitem não só o desenho dos DFD's, mas também uma *prototização* das funções do DFD ("Excelerator", "The Analyst's toolkit", "Kee", "Art"). O desenvolvimento de um *protótipo inicial*, que pelo menos trate dos casos mais simples, auxilia na definição dos requisitos do sistema.

Durante a análise, objetiva-se a aproximação da versão final das estruturas de decisão e função de cada mini-sistema do componente de IA e o desenvolvimento daquelas minibases de conhecimento.

3.3 A Aquisição de Conhecimento

Antes de se discutir a aquisição de conhecimento, é preciso destacar a fonte de conhecimento. É de vital importância que, pelo menos, uma fonte de conhecimento especializado, *um especialista*, tenha vontade e capacidade de desenvolver a base de conhecimento até a sua maturidade, sendo interessante criar uma documentação da área de atuação do sistema para facilitar as discussões com o especialista do domínio. A atitude, do especialista é crucial para o sucesso de um projeto de sistema.

No mundo atual, temos vários especialistas com diferentes opiniões em relação a um mesmo domínio, sendo importante considerar essa variância para não ignorar grande parte do conhecimento especializado realmente utilizado.

É importante construir uma fundação da base de conhecimento para facilitar o significado da discussão com o especialista do domínio [IZMB90].

A aquisição de conhecimento a partir de um especialista é uma aventura excitante tanto para o especialista quanto para o analista. Em geral, os especialistas são classificados como:

Cooperativo e capaz : é a situação mais desejável, com menor gasto de tempo e maior chance de sucesso;

Desinteressado : não acredita no sucesso do sistema, porém não se coloca como obstáculo na construção;

Incompetente ou hostil : tem dificuldade em sua auto-observação e em explicar como a tarefa é feita [Kel91].

A aquisição de conhecimento constitui-se na parte interativa da análise, na qual o analista e o usuário (especialista do domínio) trabalham juntos para construir a lógica através de uma estrutura orientada para negócios, representados por DFD's e suas mini-especificações.

A evolução da base de conhecimento para um estado especialista inicia-se no levantamento, onde é possível criar algumas regras gerais (pré-protótipo da base) para o início da base de conhecimento, continuando ao longo do projeto.

Não podemos separar teoricamente a aquisição de conhecimento do resto da análise do projeto. A aquisição de conhecimento é o conteúdo lógico e funcional do domínio do especialista, enquanto a análise inclui os componentes funcionais das atividades periféricas, tais como interface, componentes físicos e tecnológicos do sistema.

A tecnologia de SE separa física e logicamente o conhecimento do desenvolvimento do software, a fim de dar liberdade para a continuidade da aquisição de conhecimento na análise.

Essa é atividade que levará mais tempo e terá um custo relativamente alto. Ela pode continuar além da fase de instalação. Isto significa que a base de conhecimento pode evoluir por um tempo indeterminado até um ponto em que seja considerado especialista. É importante, porém, levantar uma estimativa de tempo para o usuário de quando ele poderá começar a utilizar-se do sistema, mesmo que o sistema especialista ainda não tenha um conhecimento altamente especializado.

3.4 O Projeto e a Implementação

A atividade de Projeto tem os mesmos propósitos dos sistemas convencionais. O seu objetivo é produzir uma documentação que especifique os módulos de programa e os *algoritmos* que devem ser usados nesses módulos. Também deve ser especificada a arquitetura dos módulos de programa. O analista, no processo de detalhamento da atividade de Projeto, elabora descrições e pseudo-códigos de procedimentos, desenha diagramas para a rede semântica, escreve em linguagem natural as regras de produção, identifica e nomeia *slots* para *frames* e *scripts* [WK 89], entre outros detalhes.

Em um sistema baseado no conhecimento, as atividades de Projeto e Implementação estão tipicamente relacionadas com a definição e construção da parte do sistema que faz o raciocínio e o planejamento lógico - *a máquina de inferência*.

Segundo as experiências de Irgon [IZMB90], o ambiente de desenvolvimento que suporta uma poderosa representação de conhecimento, um raciocínio avançado e programas multiparadigmas são importantes para um desenvolvimento rápido de um robusto sistema especialista.

Uma meta existente há muito tempo no desenvolvimento de sistemas é o que pode ser chamado de programação automática que, em resumo, é a automatizar o projeto e a implementação o máximo possível, reduzindo a dependência do projeto à atividade humana. Já está sendo utilizada uma classe de produtos baseados no conhecimento, que possui sofisticada capacidade de comunicação em linguagem natural, componentes de inferência e uma extensiva base embutida de conhecimento, evitando, assim, os problemas de projeto e implementação. Uma variedade de **Ambientes de Engenharia de Conhecimento** (KEE - Knowledge Engineering Environment) tem sido desenvolvida, para simplificar o desenvolvimento dos sistemas, baseado no conhecimento.

3.5 O Projeto da Base de Conhecimento

Quando da implementação da Base de Conhecimento, o analista escolhe o mecanismo para representar todo o conhecimento adquirido. Algumas técnicas de representação do conhecimento são : *Frames*, *Scripts*, *Redes Semânticas*, *Knowledge Spaces*, entre outras.

Os *Frames* permitem aos objetos serem associados a uma coleção de características (fatos, regras, padrões, valores ativos) contidos em *Slots*. Cada *Frame* é composto por um conjunto de *Slots* relacionados a um objeto específico. Os *Scripts* armazenam os eventos que ocorrem em situações familiares dentro de uma série de *Slots*. Um *Script* é composto por uma série de cenas que, por sua vez, compõem-se de uma série de eventos. As redes semânticas representam os relacionamentos entre os objetos. Os objetos são representados como *nodos* interligados entre si por arcos. Cada arco carrega uma indicação do relacionamento entre os nodos ligados. A rede utiliza agregação, especialização e herança para modelar diferentes tipos de conhecimento. *Knowledge Spaces* são subdomínios ou partições de conjunto de regras que permitem dividir uma base de conhecimento em um número de subproblemas selecionados com o uso de *subframes* [SMS92].

Mas, se a equipe de desenvolvimento decidir utilizar-se de *Shells*, teremos algumas características oferecidas por algumas delas. A maioria das *Shells* é baseada em uma única técnica de representação do conhecimento. Poucas *Shells* oferecem representações alternativas e ainda outras permitem múltiplas representações no mesmo sistema. A maioria das *Shells*, baseada em microcomputadores, usa regras de produção como sua técnica de representação do conhecimento. Algumas *Shells* suportam frames com herança, que é um processo pelo qual objetos de baixo nível assumem as características dos objetos de alto nível a que pertencem.

3.6 Testes

Em sistemas baseados no conhecimento, a atividade de teste é de fundamental importância. Como os SE são aplicados, a áreas onde o usuário requer uma alta confiabilidade do sistema, como exemplo o diagnóstico de doenças, é preciso que os critérios para testes sejam rigorosos em relação ao domínio de aplicação. Alguns problemas nesta atividade estão relacionados à medição do conhecimento armazenado e à definição da completude do conhecimento para garantir a confiabilidade do sistema.

São essenciais dois tipos de testes:

Testes do Raciocínio: os problemas com mecanismo de raciocínio podem ocorrer devido a erros no conhecimento armazenado e a erros no sistema em relação à especificação do usuário;

Testes do Conhecimento: preocupa-se em detectar conhecimento *inválido* ou *ambíguo*. O conhecimento é *inválido* quando os fatos estão incorretos. A *ambigüidade* é detectada quando o sistema escolhe diferentes soluções para casos semelhantes. O analista usa casos reais e artificiais nos testes. Essa tarefa pode gerar “ a aquisição de mais conhecimento “.

Há alguns testes importantes realizados com a execução do sistema:

- Testes rigorosos com o especialista do domínio;
- Testes para outros especialistas do domínio, para garantir a confiabilidade (através do consenso) do sistema;
- Testes com o usuário final.

4. Uma Breve Comparação entre Metodologias

O desenvolvimento de um sistema baseado no conhecimento, proposto por Keller [Kel91], utiliza a análise estruturada. Essa metodologia

auxilia a identificar as partes do sistema, que podem utilizar as técnicas de processamento de dados e as partes que podem requerer a tecnologia de IA. Um ciclo de vida de um projeto de IA é sugerido por Keller, como mostra o DFD a seguir:

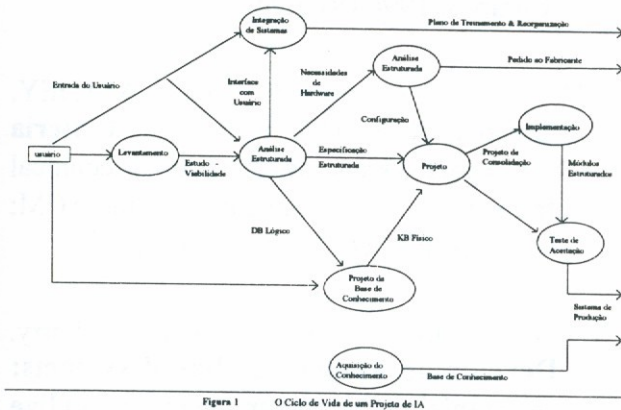


Figura 1 O Ciclo de Vida de um Projeto de IA

A análise estruturada tem o seu enfoque na funcionalidade do sistema, organizando-se em torno dos procedimentos. Um projeto estruturado impõe obstáculos à sua expansão por ser bem definido. No desenvolvimento de um SE, trabalha-se muitas vezes com informações incertas, precisando de várias modificações durante o seu desenvolvimento. Este pode ser um ponto fraco da utilização da análise estruturada no desenvolvimento de sistemas baseados no conhecimento, pois modificações constantes, nas suas estruturas bem definidas, podem ser desastrosas e muito custosas nessa metodologia.

O ciclo de vida proposto acima [Kel91] denota um ciclo de atividades seqüenciais, não demonstrando nenhuma interação entre algumas das atividades. Isso, na realidade, pode não ser verdade. Por exemplo, erros encontrados na atividade de testes podem acarretar mudanças na atividade de análise de projeto ou de implementação.

Uma outra proposta é a de Weitzel [WK89], que sugere a utilização de uma metodologia *Orientado-Objeto*, onde o analista consegue caracterizar o conhecimento em objetos, criar uma relação conhecimento/objeto e descrever processos que manipulem esses conhecimentos através de objetos. Um sistema baseado em objetos coloca sua ênfase na descrição dos objetos e deixa para um segundo plano a funcionalidade do sistema. Desse modo, uma alteração nas funções do sistema não acarretaria danos desastrosos para o desenvolvimento do sistema. E modificações nos objetos não significam alterar os procedimentos que manipulam esses objetos. Uma pessoa que não está envolvida no projeto “enxerga” melhor a definição de um sistema *Orientado-Objeto*.

O conhecimento é fornecido na definição de classe, herdado de superclasses e adquirido durante o tempo de vida do objeto [AC93].

Weitzel [WK89] propõe um outro ciclo de vida, onde se pode notar as várias interações entre os diversos processos(1). Entretanto, não coloca um processo específico para a aquisição de conhecimento, o que deve ser de vital importância para sistemas baseados no conhecimento.

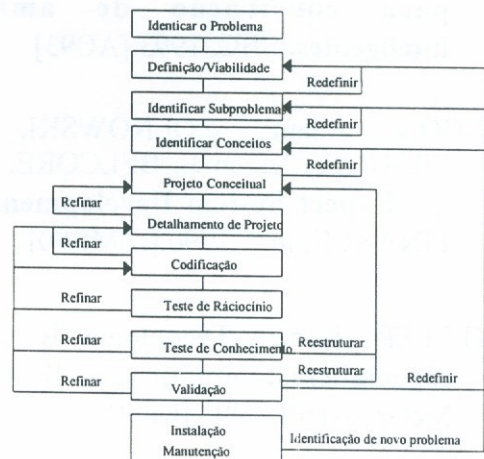


Figura 2 Fluxo da Metodologia de Desenvolvimento de Sistemas Baseados no Conhecimento

(1) Este autor define processos em vez de atividades.

Conclusão

As metodologias já bem difundidas e experimentadas podem ser facilmente adaptadas ao desenvolvimento de um Sistema Especialista, não exigindo a contratação de um engenheiro de conhecimento a preço alto, pois o próprio analista de sistemas pode tornar-se o engenheiro de conhecimento. É preciso investir em novos *softwares* que facilitem a construção de um Sistema Especialista (prototização, criação de um mecanismo de inferência e representação de conhecimento).

Um projeto baseado em objetos sugere melhores condições para um projeto de sistema especialista, oferecendo facilidades para a sua especificação, construção e manutenção do sistema especialista.

Bibliografia

1. AMARAL, Janete P.; CASTRO, Jaelson F. **B. Uma abordagem orientada a objetos para construção de ambientes inteligentes.** SBC, 1993.[AC93]
2. IRGON, Adam; ZOLNOWSKI, Jean; MURRAY, Karen J.; BELCORE, Marvin G. **Expert System Development; MIS/FINANCE;** june 1990.[IZMB90]
3. KELLER, Robert. **Tecnologia de sistemas especialistas: e aplicação.** São Paulo : McGraw-Hill, 1991.[Kel91]
4. LEVINE, Robert I.; DRANG, Diane E.; BARRY, Edelson. **Inteligência artificial e sistemas especialistas.** São Paulo : McGraw-Hill; 1988.[LDB88]
5. RICH, Elaine; KNIGHT, Kevin. **Inteligência artificial.** São Paulo : Makron Books, 1993.[RK93]
6. RUMBAUGH, James. **Modelagem e projetos baseados em objetos.** Rio de Janeiro : Campus, 1994.[Rum94]
7. STYLIANOU, Anthony, C.; MADEY, Gregory, R; SMITH, Robert D. **Criteria for expert systems shells: a socio-technical framework.** Communication of the ACM; V. 35; n. 10; oct., 1992.[SMS92]
8. WEITZEL, John R.; KERSCHBERG, Larry. **Developing knowledge-based systems: reorganizing the system development live cycle.** Communications of the ACM; V.32, n.4; apr., 1989.[WK89]
9. YOURDON, Edward. **Análise estruturada moderna.** Rio de Janeiro : Campus, 1992.[You92]