

2 - Fundamentos

Neste capítulo, apresentam-se conceitos necessários para um melhor entendimento desta dissertação. Como este trabalho se propõe ao estudo de uma abordagem para aumentar a fidedignidade de sistemas multi-agentes abertos através da noção de contratos, serão descritas características de sistemas multi-agentes, sistemas abertos e engenharia de software fidedigno.

2.1 Sistemas Abertos

A Teoria Geral dos Sistemas (TGS) foi desenvolvida pelo biólogo húngaro, Ludwig von Bertalanffy, em 1936. Bertalanffy utilizou-se do seguinte conceito para a definição de um sistema:

“Sistema é uma entidade que tem a capacidade de manter certo grau de organização em face de mudanças internas ou externas, composto de um conjunto de elementos, em interação, segundo determinadas leis, para atingir um objetivo específico”.

Sistema pode ser definido como um conjunto de elementos interdependentes que interagem e que possuem objetivos comuns formando um todo, e onde cada um dos elementos componentes comporta-se, por sua vez, como um sistema cujo resultado é maior do que o resultado que as unidades poderiam ter se funcionassem independentemente. Qualquer conjunto de partes unidas entre si pode ser considerado um sistema, desde que as relações entre as partes e o comportamento do todo seja o foco de atenção (Alvarez et al, 1997).

Em sistemas computacionais, o conceito de sistemas abertos tem ganhado grande importância devido ao aumento da demanda industrial por sistemas mais complexos, no qual aplicações baseiam-se na comunicação entre sub-sistemas que são desenvolvidos separadamente e independentemente. (Hewitt, 1986) identificou as seguintes características de um sistema aberto:

- Não existem objetos globais. A única semelhança entre os subsistemas é a capacidade de comunicação de uns com os outros. Isto significa que em sistemas abertos não é possível

presumir nada sobre aspectos internos dos subsistemas, como arquitetura, por exemplo.

- É difícil determinar quais são os objetos que existem no sistema. Essa segunda característica reflete a autonomia que cada subsistema possui para realizar as suas ações. Subsistemas podem entrar e sair do sistema a qualquer momento, seja por uma falha de funcionamento ou por decisão dos seus desenvolvedores.
- Subsistemas têm conhecimento apenas parcial do sistema como um todo. Essa característica reflete a incerteza inerente aos sistemas abertos. Cada subsistema tem somente conhecimento parcial sobre o sistema como um todo e novos conhecimentos podem ser adquiridos ou modificados em função da evolução da interação entre os subsistemas.

Como consequência dessas três noções básicas, pode-se dizer que sistemas abertos são (i) concorrentes, uma vez que são compostos por diferentes subsistemas operando geograficamente distribuídos e utilizam-se de diferentes fontes de informações; (ii) assíncronos; (iii) possuem controle descentralizado, no sentido de que o fluxo de execução do sistema é regido por decisões locais dos subsistemas; (iv) e podem conter informação inconsistente (Hewitt, 1986).

Sistemas abertos são dinâmicos por definição. Geralmente não é possível se prescrever durante a elaboração do sistema quais serão os componentes (e em quais plataformas estarão sendo executados) que comporão o sistema futuro.

Diversas tecnologias tem sido desenvolvidas para facilitar a organização de sistemas abertos, tais como web-services (W2C, 2002) , protocolos de comunicação, como TCP/IP e Contract Net (FIPA, 2002), e paradigmas de software, tal como a engenharia de software orientada a agentes - ESOA (Jennings, 2001), (Choren et al, 2004), e a arquitetura orientada a serviços- SOA – Service Oriented Service (Newcomer et al, 2005]), que possui como componente fundamental o conceito de serviços em ambientes abertos heterogeneos.

2.2 Sistemas Multi-Agentes

Atualmente, há muitas definições para o termo agente (Jennings, 1999) (Ferber, 1999). Basicamente, um agente é uma entidade computacional com um comportamento autônomo que lhe permite decidir suas próprias ações (Ferber, 1999). A decisão de qual ação levar a cabo é determinada pelo agente, tendo em consideração as mudanças acontecidas no ambiente em que atua e o desejo de alcançar seus objetivos. (Wooldridge et al, 1995) usou a seguinte definição para agente:

“Um agente é um sistema computacional encapsulado que está situado em algum ambiente e é capaz de ação flexível autônoma neste ambiente, a fim de alcançar seus objetivos de projeto”.

Ainda seguindo a linha de pesquisa de (Wooldridge et al, 1995), agentes podem possuir as seguintes propriedades:

- **Autonomia:** os agentes operam sem a intervenção direta de qualquer entidade, seja ela humana ou não. Além disso, eles possuem controle sobre suas ações e sobre o seu estado.
- **Reatividade:** capacidade de perceber o ambiente e responder às mudanças deste, de forma a satisfazer os objetivos do agente. É importante em ambientes dinâmicos, pois dado que o ambiente torna-se incerto, assumir posições imutáveis não é razoável, devendo haver adaptação à nova realidade.
- **Habilidade Social:** retrata a capacidade de um agente trabalhar em parceria com outros, característica esta marcante em um sistema multi-agentes. Isto é necessário uma vez que determinadas tarefas são demasiadamente complexas para um agente resolver sozinho. Nestes casos, podem-se trocar dados com agentes que ajudem na solução, particularmente quando estes são especialistas em uma parte do problema a ser solucionado. Naturalmente, esta capacidade só se torna viável se houver comunicação entre eles. Portanto é necessário estabelecer protocolos comuns, para que todos os participantes troquem informações, tendo por base um conhecimento comum da sua ontologia¹. Desta forma o caráter

¹ Ontologia é o estudo do ser, sendo através dela possível descrevê-lo para que seja então conhecido. Na medida em que a ontologia de um ser seja conhecida por outro

semântico de uma mensagem é garantido, pois todos os seres têm um mesmo tipo de conhecimento e, portanto de interpretação do mundo que habitam. Isto denota sociabilidade dos agentes, podendo-se falar de sociedade de agentes. (Bradshaw, 1997), (Weiss, 1999), (Russel et al, 1995).

- **Inferência:** capacidade de poder de agir sob um determinado ambiente, utilizando o conhecimento adquirido, analisar cenários e tomar decisões baseadas em processos “cognitivos” que se aproximam do raciocínio. Com isto o agente torna-se flexível para responder às mudanças do ambiente e satisfazer os seus objetivos. Um agente com esta característica é chamado de agente inteligente, já que possui algum conhecimento e pode adaptá-lo, ou seja, aprender novos itens para adequar-se à realidade do ambiente em que se encontra. É importante frisar que este conhecimento é válido no contexto no qual está o agente; qualquer alteração deste contexto pode invalidar totalmente o conhecimento adquirido.
- **Pró-atividade:** os agentes não agem simplesmente em resposta ao ambiente, mas eles são capazes de “tomar a iniciativa”. Ou seja, além de reagir a mudanças ocorridas no ambiente, o agente ainda pode alterar seus estados internos durante o processamento e tomar uma ação, baseada em seus novos estados. Os agentes pró-ativos podem em função de seus objetivos se antecipar e efetuar ações que não sejam meramente reativas.
- **Continuidade Temporal:** representa a capacidade do agente em permanecer desativado por um certo período, durante o qual mantém seus estados internos, e que ao ser reativado retomará como seu estado inicial o mesmo estado imediatamente anterior à desativação, ou seja, o agente reinicia seu processamento do ponto

torna-se possível ao primeiro entender o segundo e, portanto com ele se comunicar e trocar conhecimento. O eventual uso de diferentes ontologias para um mesmo ser permite sua descrição sob diversos enfoques , ou com diferentes roupagens. Neste caso consegue-se criar interfaces do ser descrito que sejam apropriadas para reconhecimento por interlocutores de diferentes espécies.

exato onde parou. Esta característica é importante, pois garante que o conhecimento adquirido pelo agente seja mantido e possa ser restabelecido quando este for reativado.

- **Noções mentais:** os agentes possuem conceitos que são geralmente aplicados a humanos, como por exemplo: conhecimento, crença, intenção e obrigação.

A distinção entre agente e objeto pode ser entendida em função da autonomia. Em muitos casos um objeto faz parte de um sistema maior, e executa alguma ação, mas sempre sob comando do sistema maior; o objeto em si não executaria uma ação baseada apenas em seus estados internos ou em resposta aos seus objetivos. Ocorre em diversos casos que os estados internos são controlados por outros objetos.

2.2.1 Inteligência dos Agentes

A inteligência de um agente ainda é tema polêmico na comunidade de pesquisa, uma vez que diferentes pesquisadores utilizam características diferentes para definir o comportamento de um agente inteligente. Segundo (Bradshaw, 1997), (Weiss, 1999) e (Russel et al, 1995), a inteligência de um agente pode ser definida como a capacidade de raciocínio e comportamento aprendido. Segundo (Wittig, 1999): “a capacidade de sobreviver dá o grau de inteligência. Um sistema é inteligente tanto quanto sua capacidade de maximizar as propriedades de sobrevivência em um dado ambiente”.

Segundo o artigo publicado pela IBM (Gilbert et al, 1995) três elementos definem as principais capacidades que agentes inteligentes em software devem apresentar:

- Inteligência - grau de raciocínio e aprendizagem;
- Agência - grau de autonomia e autoridade incorporadas;
- Mobilidade - capacidade de tráfego pela rede.

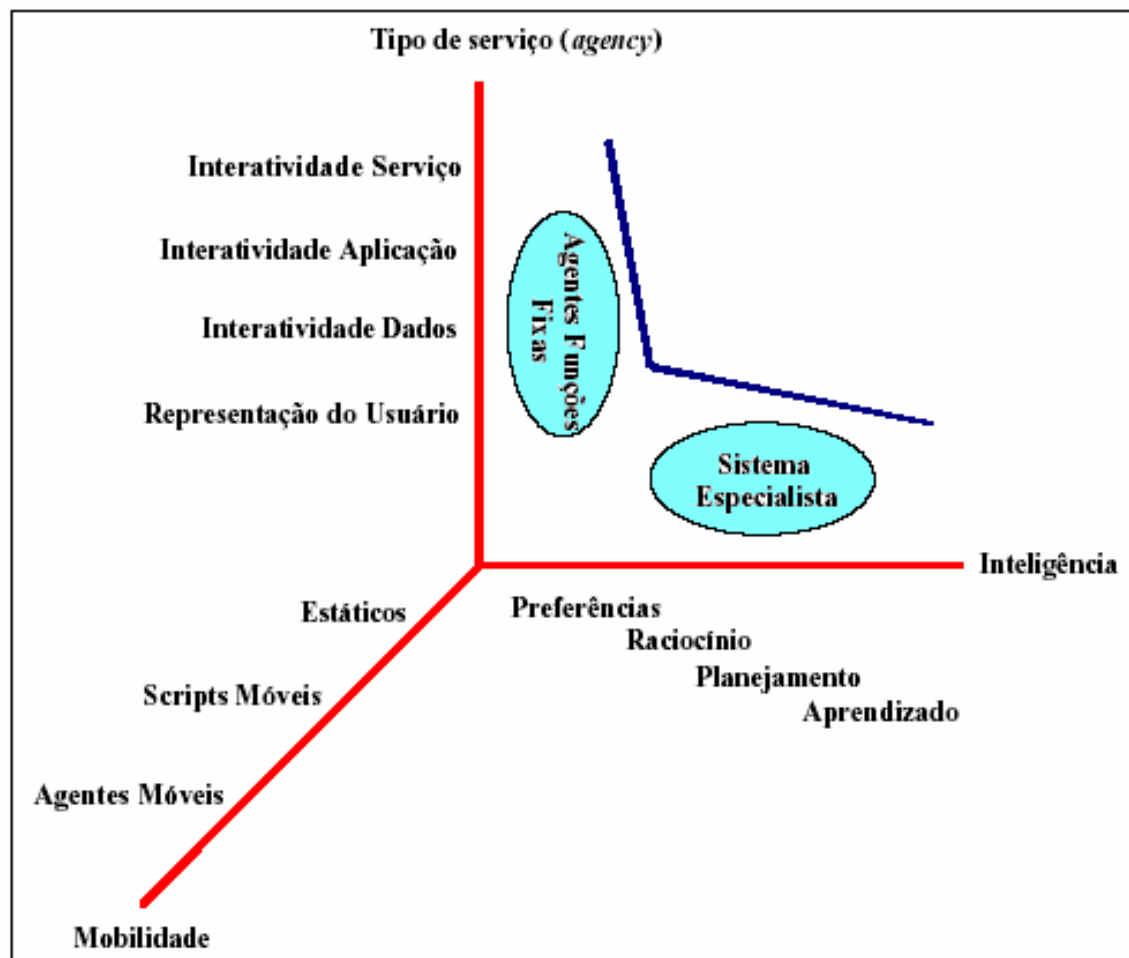


Figura 1: Classificação de um agente (Gilbert et al. 1995) adaptado.

Tipo de Serviço (*Agency*) é o grau de autonomia e autoridade do agente, e pode ser medido qualitativamente através da natureza da interação entre o agente e outras entidades do sistema. No mínimo, um agente deve ser capaz de executar de forma assíncrona (Bradshaw, 1997). O tipo de serviço difere dependendo do ambiente no qual o agente está inserido. Por exemplo, o agente pode prestar um serviço de interação com dados, aplicações ou mesmo representar um usuário.

Inteligência é o grau de raciocínio e aprendizagem do agente: é a habilidade que o agente possui em assumir objetivos e suportar tarefas delegadas a ele (Bradshaw, 1997). A inteligência está fortemente relacionada à capacidade de aprendizado e adaptação do agente.

Mobilidade é definida pela capacidade de tráfego do agente pela rede. Existem diversas plataformas de agentes móveis em software que suportam a criação de processos computacionais autônomos e assíncronos, capazes de execução e migração continuada entre computadores em rede. O aspecto que

mais diferencia entre a migração de agentes e a transferência de código móvel (também chamado de código sob demanda) é que, quando migra entre *hosts*, os agentes transferem não apenas o código, mas também o estado da sua execução. O modelo computacional de agentes móveis pode ser considerado um caso particular de tecnologia de código móvel. No entanto, não é objetivo desta dissertação o estudo mais detalhado sobre agentes móveis.

2.2.2 Características de Ambientes Multi-Agentes

(Weiss, 1999) define ambientes Multi-agentes como ambientes que:

- provêem uma infra-estrutura especificando protocolos de comunicação e interação.
- são geralmente abertos e não possuem apenas um projetista – não são centralizados
- contêm agentes que são autônomos e distribuídos, e podem ser competitivos ou cooperativos.

O ambiente de um agente pode ser definido como a estrutura onde o agente realiza suas ações e percebe (ou recebe) sinais enviados pelas entidades com as quais se relaciona ou que de alguma forma o influenciam.

| Propriedade | Valores |
|---|--|
| <i>Autonomia de Projeto</i> | <i>Plataforma/Protocolo de Interação/Linguagem/Arquitetura Interna</i> |
| <i>Infra-estrutura de Comunicação</i> | <i>Memória Compartilhada (blackboard) ou Baseada em Mensagens, Orientada à Conexão ou Não Orientada à Conexão(e-mail), Ponto-a-Ponto, Multicast ou Broadcast, Síncrono ou Assíncrono</i> |
| <i>Diretório de Serviço (Directory Service)</i> | <i>White pages, Yellow pages</i> |
| <i>Protocolo de Mensagens</i> | <i>KQML, FIPA-ACL, HTTP OU HTML, OLE, CORBA</i> |
| <i>Serviço de Mediação</i> | <i>Baseado em Ontologia ou Transações</i> |
| <i>Serviço de Segurança</i> | <i>Timestamps/Autenticação</i> |
| <i>Operações de Suporte</i> | <i>Armazenamento/Redundância/Restauração/Accounting</i> |

Figura 2: Características de um ambiente Multi-Agente, segundo (Weiss, 1999).

2.2.3 Classificação dos Agentes

Podemos classificar os agentes entre diferentes tipos, levando em consideração suas habilidades de raciocínio, mobilidade e relacionamento com outros agentes ou objetos. A seguir listamos alguns dos principais tipos de agentes encontrados na literatura:

- **agentes situados ou estacionários:** são aqueles opostos aos móveis. Isto é, são fixo em um mesmo ambiente e ou plataforma. Não se movimentam em uma rede e muito menos na Web.
- **agentes móveis:** são agentes que tem a mobilidade como característica principal. Isto é, uma capacidade de mover-se seja por uma rede interna local (intranet) ou até mesmo pelo Web, transportando-se pelas plataformas levando dados e códigos. Seu uso tem crescido devido alguns fatos como uma heterogeneidade cada vez maior das redes e seu grande auxílio em tomadas de decisões baseadas em grandes quantidades de informação.
- **agentes competitivos:** são agentes que “competem” entre si para a realização de seus objetivos ou tarefas. Ou seja, não há colaboração entre os agentes.
- **agentes coordenados ou colaborativos:** agentes com a finalidade de alcançar um objetivo maior, realizam tarefas específicas porém coordenando-as entre si de forma que suas atividades se completem.
- **agentes Reativos:** são agentes que reagem a estímulos sem ter memória do que já foi realizado no passado e nem previsão da ação a ser tomada no futuro. Não tem representação do seu ambiente ou de outros agentes e são incapazes de prever e antecipar ações. Geralmente atuam em sociedades como uma colônia de formiga por exemplo. Muitas vezes baseiam-se na “teoria do caos” a qual afirma que até mesmo no caos existe uma “certa organização”. No caso da formiga, por exemplo, uma única

não apresenta muita inteligência, mas quando age no grupo comporta-se o todo como uma entidade com uma certa inteligência. Ou seja, a força de um agente reativo vem da capacidade de formar um grupo e construir colônias capazes de adaptar-se a um ambiente.

- **agentes cognitivos:** esses, ao contrario dos agentes reativos, podem raciocinar sobre as ações tomadas no passado e planejar ações a serem tomadas no futuro. Ou seja, um agente cognitivo é capaz de “resolver” problemas por ele mesmo. Ele tem objetivos e planos explícitos os quais permitem atingir seu objetivo final. (Ferber, 1999) afirma que para que isso se concretize, cada agente deve ter uma base de conhecimento disponível, que compreende todos os dados e todo o “know-how” para realizar suas tarefas e interagir com outros agentes e com o próprio ambiente. Sua representação interna e seus mecanismos de inferência o permitem atuar independentemente dos outros agentes e lhe dão uma grande flexibilidade na forma de expressão de seu comportamento. Além disso, devido a sua capacidade de raciocínio baseado nas representações do mundo, são capazes de ao mesmo tempo memorizar situações, analisá-las e prever possíveis reações para suas ações.

2.3 Fidedignidade em Sistemas de Software

Nesta seção, apresenta-se os principais conceitos sobre a engenharia de software fidedigno (inglês: *dependable*), sua importância para sistemas multi-agentes abertos e uma breve noção de como a abordagem de contratos pode ajudar a prover software fidedigno. Um software é dito fidedigno quando se pode justificavelmente depender dele assumindo riscos de danos compatíveis com o serviço prestado pelo software (Avizienis et al, 2004).

2.3.1 Fidedignidade e seus atributos

Fidedignidade é definida como o nível de confiança associada à entrega de um serviço por parte de um sistema. Isto implica que o sistema não apresenta erros, não é suscetível a falhas de segurança, se mantém íntegro durante a sua

execução, satisfaz as especificações determinadas, satisfaz os usuários, dentre outras características (Avizienis et al, 2004).

Em (Avizienis et al, 2004b) são estabelecidas as propriedades que sistemas fidedignos devem possuir, baseando-se em (Scott et al, 1987). Entretanto, segundo um estudo realizado por (Staa, 2006), esta lista de propriedades é fortemente focada em sistemas embarcados e não evidencia adequadamente algumas das propriedades relevantes no que toca a evolução do software. Segundo (Staa, 2006), pode-se definir fidedignidade através dos seguintes atributos:

- **Disponibilidade:** estar pronto para prestar serviço correto sempre que se necessite do software.
- **Confiabilidade:** habilidade de sempre prestar serviço correto. Cabe observar que mais de 50% das falhas observadas em sistemas em uso devem-se a erros de operador (Patterson et al, 2004), (Fox, 2002). Ou seja, confiabilidade não é somente assegurar que o sistema esteja em correspondência exata com a sua especificação.
- **Segurança:** habilidade de evitar consequências catastróficas, ou de grande envergadura, afetando tanto os usuários como o ambiente.
- **Proteção:** habilidade de evitar tentativas de agressão bem sucedidas (em (Avizienis et al, 2004b) esta é uma propriedade agregadora no mesmo nível que fidedignidade).
- **Privacidade:** habilidade de proteger dados e código contra acesso indevido. Cabe observar que isto se refere não só aos dados contidos na máquina, mas também aos dados em trânsito entre máquinas, mesmo quando estiverem em uso mecanismos diferentes de redes de computadores.
- **Integridade:** ausência de alterações não permitidas (corrupção de elementos).

- **Robustez:** habilidade de detectar o mais cedo possível eventuais falhas de modo que os danos (as correspondentes consequências) possam ser mantidas em um patamar aceitável (em (Avizienis et al, 2004b) esta é considerada uma propriedade secundária).
- **Recuperabilidade:** habilidade em ser rapidamente repostado em operação fidedigna após a ocorrência de uma falha. Em (Avizienis et al, 2004b) é considerada uma forma de implementação da tolerância a falhas. Já Patterson em (Patterson et al, 2004) e Fox em (Fox, 2002) consideram recuperabilidade uma característica suficientemente marcante para merecer atenção individualizada no nível das demais características da fidedignidade.
- **Manutenibilidade:** habilidade de ser modificado (evoluído) ou corrigido sem que novos problemas sejam inseridos e a um custo compatível com o tamanho da alteração.
- **Depurabilidade:** facilidade de diagnosticar e eliminar possíveis causas de problemas a partir de relatos gerados. Embora a depuração faça parte da manutenção, o diagnóstico da causa de uma falha a partir de um relato de usuário ou de sintomas observados, tem-se mostrado extremamente frustrante e desgastante, principalmente quando se lida com software produto. A manutenibilidade também não explicita a necessidade de se projetar e incluir instrumentação de modo que seja reduzido o tempo necessário para corretamente diagnosticar e eliminar as causas das falhas identificadas (Staa, 2000).

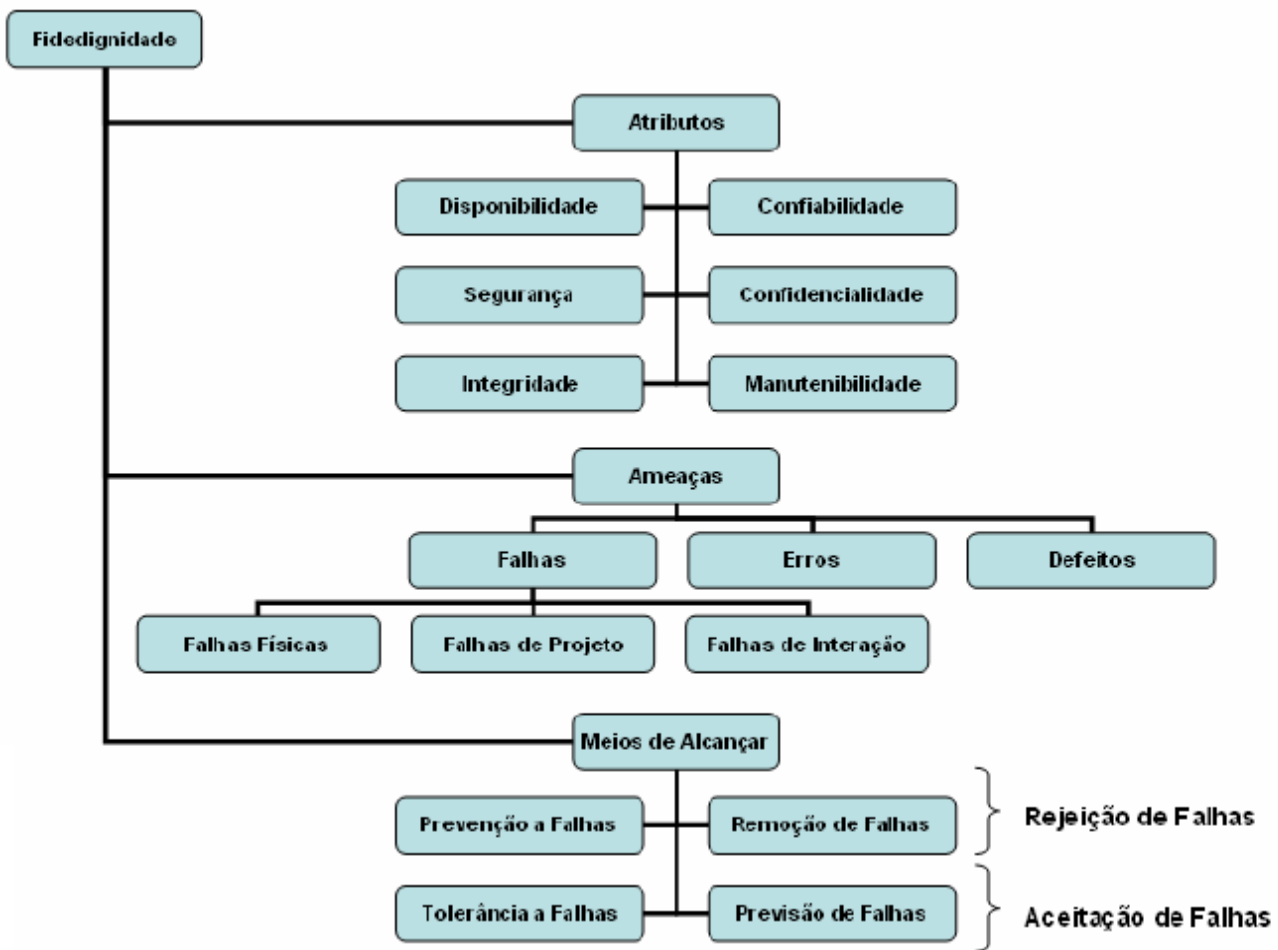


Figura 3: Fidedignidade e seus atributos (Gatti, 2006), adaptado de (Avizienis et al, 2004).

As ameaças para a fidedignidade de sistemas consistem em erros, faltas e falhas. Uma falha no sistema ocorre quando um serviço que é entregue difere do serviço correto. Um erro é o estado do sistema que levou ou pode levar a uma falha. E, finalmente, uma falta é a causa suposta ou real de um erro. Uma falta está ativa/latente quando ela produz erros, senão diz-se que ela está dormente.

Como exibido de forma sucinta na Figura 3, existem quatro técnicas que devem ser empregadas ou implementadas para se alcançar a fidedignidade:

| Técnica | Descrição |
|---------------------|--|
| Prevenção de falhas | Provê meios de prevenir a ocorrência ou introdução de falhas |
| Remoção de falhas | Provê meios de reduzir as falhas em número e severidade |

| | |
|---------------------|--|
| Tolerância a falhas | Provê meios de evitar defeitos em serviços na presença de falhas |
| Previsão de falhas | Provê meios de estimar o número presente, a incidência futura, e as conseqüências das falhas |

Tabela 1: Técnicas de alcançar a fidedignidade (Gatti, 2006).

Como exposto na Tabela 1, as técnicas de prevenção de falhas e tolerância a falhas objetivam prover a habilidade de entregar um serviço que pode ser confiável, enquanto que as de remoção de falhas e previsão de falhas objetivam alcançar a confiança no sentido de que as especificações funcionais e de fidedignidade são adequadas e que o sistema as implementa (Gatti, 2006).

Este trabalho foca especialmente na previsão de falhas, visando identificar agentes que não desempenham o serviço esperado na organização. Não é objetivo deste trabalho tratar a correção ou redução de falhas, tão pouco prover um mecanismo de tolerância a falhas, o que pode ser encontrado no trabalho de (Gatti, 2006).

Um contrato define o contexto no qual um serviço deverá ser oferecido. Por meio de pré e pós condições para que um serviço seja executado, o contrato protege ambas partes que o integram: o cliente, assegurando que receberá o serviço requisitado; e o executor, delimitando o serviço a ser prestado. O mecanismo de gerenciamento de contratos proposto neste trabalho visa identificar agentes que firmam essas pré e pós condições que compõem um contrato, com o objetivo de facilitar a substituição ou correção destes agentes e reduzir as falhas no sistema.