

Uma Arquitetura de Serviço para Avaliação de Contextos em Redes de TV Digital

Luiz Eduardo C. Leite^{1,2}, Ozifrankly Lima², Guido L. de Souza Filho²,
Silvio R. de L. Meira¹, Patrícia C. A. R. Tedesco¹

¹Centro de Informática
Universidade Federal de Pernambuco (UFPE)
Recife – PE

²Laboratório de Aplicações de Vídeo Digital – Departamento de Informática
Universidade Federal da Paraíba (UFPB)
João Pessoa – PB

{leduardo, ozzy, guido}@lavid.ufpb.br, {silvio, pcart}@cin.ufpe.br

Abstract. *Aiming to increase the audience of TV programs, Digital TV content producers usually want to produce content which is most suitable for their users. So, they frequently need to know the context their users are in. This work presents the architecture and the implementation of a context evaluation service. The evaluation service is distributed and it is provided by the digital TV receivers for the content producers in the form of Web Services. The variables which constitute the context are defined in a domain ontology. The context evaluation is performed by an inference engine which uses formulations on the fuzzy logics.*

Resumo. *Hoje em dia é cada vez mais comum a geração e transmissão de conteúdo personalizado para os usuários das redes de TV Digital. Para que essa geração seja possível, faz-se necessário que os produtores de conteúdo conheçam o contexto em que seus usuários se encontram. Este trabalho apresenta a arquitetura e a implementação de um serviço distribuído para avaliação do contexto dos usuários das redes de TV Digital. Esse serviço é provido pelos receptores de TV Digital para os produtores de conteúdo na forma de Web Services. As variáveis que constituem o contexto são definidas na forma de uma ontologia de domínio. A avaliação do contexto é feita através de um motor de inferência baseado em lógica fuzzy.*

1. Introdução

Hodiernamente observa-se que diversos países em todo o mundo estão paulatinamente substituindo os já obsoletos sistemas de transmissão de televisão analógica por outros digitais. Devido aos grandes benefícios oriundos dessa tecnologia, espera-se que uma parcela significativa dos países contemporâneos implemente algum tipo de rede digital para transmissão de sinais de TV. Além da melhoria na qualidade da imagem e do som, dentre os principais benefícios proporcionados pela TV Digital deve ser citada a possibilidade de veiculação de programas interativos e personalizados.

Um programa de TV Digital Interativa – TVDI é aquele com o qual um telespectador pode interagir através da própria TV. Exemplos desses tipos de programa são: TV bank, Internet TV, TV games, dentre outros. Para que esses programas interativos sejam factíveis, faz-se necessário que os aparelhos de recepção de TV Digital sejam dotados de poder de processamento computacional que possa ser utilizado para executar pequenos aplicativos denominados Xlets – semelhantes a pequenos programas de computador. Esses Xlets são enviados pela emissora de TV para os receptores de TV Digital, juntamente com os sinais de áudio, vídeo e dados. No caso dos dados, eles podem ser imagens, sons, vídeos extras, textos ou qualquer outro tipo de informação que possa ser utilizada pelos Xlets. Um programa de TV interativa pode então ser considerado como uma composição de vários artefatos computacionais, enviada para os receptores de TV digital.

Ao gerar um conteúdo televisivo, uma produtora de TV comumente deseja que o mesmo seja do interesse e adequado à maioria dos seus telespectadores. Assim sendo, antes de materializar cada programa, a produtora precisa conhecer qual é o seu público e quais são suas principais características. De posse dessa informação, a produtora pode avaliar o grau de adequação do conteúdo produzido para um determinado público alvo. Em um ambiente de TV Digital essa avaliação pode ser ainda mais complexa. Considerando-se que existem diversos tipos de receptores de TV Digital – alguns mais sofisticados com processadores rápidos, grande quantidade de memória, canal de retorno de dados, etc. e outros mais simples, capazes apenas de exibir áudio e vídeo – determinados programas de TVDI serão mais adequados para alguns tipos de receptores do que para outros. Além disso, como os receptores de TV Digital podem ser dispositivos móveis, eles podem estar inseridos em diversos ambientes, recebendo sinais de TV através de redes de transmissão com diferentes níveis de QoS. Esse fato pode afetar de sobremaneira o funcionamento de alguns programas de TVDI, fazendo com que eles sejam mais adequados para alguns ambientes do que para outros.

Os telespectadores que estão correntemente fazendo uso de um receptor de TV, suas características, bem como as do aparelho e do ambiente no qual ele está inserido, no escopo deste trabalho, serão conjuntamente chamados de contexto dos telespectadores. Dessa forma, uma produtora TV deveria ponderar todas as variáveis que constituem o contexto dos seus telespectadores para avaliar o quão adequado é um determinado programa de TV para os mesmos. Como os programas de TV Digital podem ser considerados composições de vários artefatos computacionais, idealmente uma produtora de TV deveria ser capaz de avaliar também a adequação ou não de cada um desses artefatos para o contexto do seu público.

Esse trabalho apresenta a arquitetura de um serviço distribuído para avaliação da adequação de artefatos computacionais ao contexto de telespectadores em redes de TV digital. Na arquitetura proposta, uma ontologia de domínio é utilizada para definir quais são as variáveis que constituem o contexto dos telespectadores nas redes de TV digital. Através de um Web Service provido por cada receptor digital, produtores de conteúdo podem enviar uma descrição de como um conteúdo televisivo deve ser avaliado no próprio receptor, quanto a sua adequação ou não para os usuários correntes do receptor. Essa abordagem de implementar o serviço nos próprios receptores diminui a demanda por processamento nas emissoras, utilizando o processamento ocioso nos receptores, e evita o envio de informações privadas dos telespectadores para as emissoras. Nos

receptores, um motor de inferência baseado em lógica fuzzy é utilizado para determinar o grau de adequação do conteúdo televisivo descrito para os seus telespectadores.

O restante deste trabalho está organizado da seguinte maneira. As seções 2, 3 e 4 apresentam respectivamente embasamentos teóricos sobre TV digital interativa, contexto e lógica fuzzy. A seção 5 apresenta a arquitetura proposta para o serviço de avaliação de contexto em redes de TV digital, a seção 6 traz alguns aspectos relacionados à implementação e as provas de conceito realizadas; a seção 7 discute alguns trabalhos relacionados e a seção 8 traz as conclusões deste trabalho.

2. TV Digital Interativa

Televisão Digital pode ser considerada um sistema de transmissão de conteúdo multimídia que se caracteriza pela forma com que os seus usuários recebem a informação. Em outros sistemas computacionais, como os baseados na WEB por exemplo, os usuários podem selecionar e visualizar a informação na qual estão interessados, navegando pelas partes do conteúdo que forem do seu interesse, na ordem em que melhor lhes convier. Já em um sistema de TV digital, a seqüência de apresentação das informações é predefinida por produtores de conteúdo, antes do programa ser exibido para os usuários. Além disso, uma outra característica importante é que as informações são transmitidas e apresentadas simultaneamente a todos os usuários do sistema através de uma rede de transmissão digital. Determinar então quais são as preferências e intenções dos telespectadores que assistem TV em um dado momento é de fundamental importância para que a produção de um programa possa agradar à maioria dos telespectadores.

A comunicação em uma rede de TV normalmente é do tipo *simplex*, onde as cabeças de rede enviam o sinal de TV para retransmissoras e essas o reenviam para receptores de TV através de um canal de difusão – canal de *broadcast*, utilizando um protocolo de transporte apropriado – usualmente o MPEG-2 transporte. Entretanto, as redes de TV digital possibilitam que dados sejam enviados dos receptores para as emissoras de TV através de canais de retorno apropriados. Um canal de retorno pode ser de vários tipos, como por exemplo PTSN, GSM, Internet e transmissão intra-banda.

Nas redes de TV digital, além dos já tradicionais televisores, outros dispositivos podem ser utilizados para recepção e exibição do sinal de TV, tais como computadores, PDAs, aparelhos de telefone celular, etc. No escopo deste trabalho, estes dispositivos, incluindo os próprios televisores digitais, serão genericamente chamados de Dispositivo de Recepção de TV Digital – DRTVD. Além de exibir áudio e vídeo, estes dispositivos devem ser capazes de executar aplicativos, semelhantes a pequenos programas de computador – os Xlets, transmitidos pelas emissoras através do canal de *broadcast* juntamente com o áudio e vídeo.

Considerando-se que existem diversos tipos de DRTVDs e que para um mesmo tipo de aparelho cada fabricante pode produzir diversos perfis com funcionalidades e custos diferentes, além de utilizar arquiteturas de hardware proprietárias, as características de cada DRTVD podem variar muito de aparelho para aparelho. É inevitável que tais características exerçam alguma influência no funcionamento das aplicações. Por exemplo, DRTVDs com processadores mais velozes e maior quantidade de memória podem executar determinadas aplicações de forma mais apropriada do que

aqueles com menor poder de processamento; a presença ou não de um canal de retorno pode viabilizar ou inviabilizar a apresentação de determinados programas interativos; etc.

Cada tipo de DRTVD deve receber o sinal de TV digital através de protocolos apropriados. Além disso, alguns tipos de DRTVD são móveis. Quando esses dispositivos estão em ambientes ruidosos ou mesmo quando eles estão em movimento, a qualidade do serviço (QoS) provida pela rede de transmissão e pelo canal de retorno podem variar ao longo do tempo. Essa variação no nível do QoS provido pela rede de TV digital pode inviabilizar a execução de determinadas aplicações que precisam receber e enviar dados em tempo real.

No escopo deste trabalho, as características específicas do dispositivo de recepção de TV digital (CPU, memória, dispositivo de armazenamento, resolução da tela, etc.), aliadas às características do ambiente no qual o dispositivo está inserido (protocolo de transmissão, QoS de rede, etc.) e às preferências dos telespectadores que correntemente utilizam o aparelho de TV são denominadas de contexto de telespectadores de TV Digital.

3. Contexto

Hoje em dia encontram-se na literatura várias definições para contexto. De acordo com [Dey, 2000], "...contexto é qualquer informação que caracteriza a situação de uma entidade, onde uma entidade é uma pessoa, lugar ou objeto, considerados relevantes para a interação entre um usuário e uma aplicação. No caso da computação móvel, por exemplo, o contexto é tipicamente: a localização, a identidade e o estado das pessoas, grupos e objetos físicos e computacionais". Já [Brézillon, 1999] define contexto como sendo "...um conjunto de condições relevantes e influências que tornam uma situação única e compreensível". Um ponto comum que se pode extrair da maioria das definições de contexto é que as informações que caracterizam um contexto devem ser aquelas que são relevantes para o funcionamento de uma dada aplicação.

3.1. Representação de Contexto

Além da definição semântica das variáveis que definem o contexto de uma aplicação, é necessário que se tenha também uma definição sintática de como estas devem ser representadas computacionalmente, de forma a possibilitar que as aplicações possam acessar, manipular e intercambiar a informação contextual. Diversas técnicas vêm sendo utilizadas na literatura para representação de contexto, tais como par chave-valor, linguagem de marcação, orientação a objetos, mapas de tópicos e ontologias. Nesse trabalho optou-se por utilizar uma abordagem baseada em ontologias para representação do contexto, devido à sua característica de representação de conceitos através de classes e relacionamentos, o que facilita a inferência de informações não diretamente monitoradas e a representação de sistemas orientados a componentes.

Uma ontologia é uma especificação explícita de uma conceitualização compartilhada em que objetos, conceitos, entidades e relacionamentos do mundo real são definidos em uma determinada área de interesse ou domínio do conhecimento [Gruber, 1993]. Um dos grandes interesses na construção e uso de ontologias é tornar o conhecimento do mundo real processável por máquinas. Uma representação explícita de

uma ontologia consiste de um modelo estruturado de vocábulos (classes e propriedades) e sua semântica associada (relacionamentos entre diferentes classes e propriedades), bem como funções, axiomas e instâncias.

A OWL (Web Ontology Language) [Bechhofer, 2004] é um dos padrões definidos para a Web Semântica e permite a definição e representação de ontologias de domínio e compartilhamento de vocabulários. Através de OWL é possível que os diversos conceitos que constituem um determinado domínio sejam representados em uma abordagem orientada a objetos, através de classes e propriedades. Relacionamentos entre diferentes classes e propriedades podem ser definidos com vistas a associar semântica a estes elementos.

Para compor uma base de conhecimento em OWL, arquivos de ontologia têm associados a si arquivos de instâncias. Arquivos de ontologia contêm descrições dos conceitos do domínio, enquanto os arquivos de instâncias contêm os fatos acerca desses conceitos. Por exemplo, em um arquivo de ontologia pode-se definir o conceito de “Canal de Broadcast” que está associado a um outro conceito chamado “QoS” que tem uma propriedade chamada “Taxa de Perda de Pacotes”. Em um arquivo de instância associado ao arquivo de ontologia em questão pode-se ter definido que “existe um canal de broadcast pertencente à emissora de TV Tabajara”, esse canal possui uma “QoS” associada que possui, em um dado momento, “uma taxa de perda de pacotes” de 1%.

A partir das informações contidas em uma base de conhecimento um mecanismo de inferência pode ser utilizado para inferir novas informações. Por exemplo, dada a informação de que “a taxa de perda de pacotes de um canal de comunicação” é de 1%, o motor de inferência pode inferir que tal canal não é apropriado para aplicações conversacionais, por exemplo. Um tipo de mecanismo de inferência é o baseado em lógica fuzzy.

4. Conjuntos e Lógica Fuzzy

A teoria dos conjuntos nebulosos ou *fuzzy* é o modelo mais tradicional para o tratamento da informação imprecisa e vaga [Klir 1995]. Este modelo tem por objetivo permitir graduações na pertinência de um elemento a um dado conjunto. Dado um universo de discurso Ω , um conjunto nebuloso de A em Ω é definido por uma função de pertinência $\mu_A : \Omega \rightarrow [0, 1]$, que associa a cada elemento x de A o grau $\mu_A(x)$, com o qual x pertence a A. $\mu_A(x)$ indica o grau de compatibilidade entre x e o conceito expresso por A.

- $\mu_A(x) = 1$, x é completamente compatível com A
- $\mu_A(x) = 0$, x é completamente incompatível com A
- $0 < \mu_A(x) < 1$, x é parcialmente compatível com A, com grau $\mu_A(x)$.

Por exemplo, podemos considerar que o universo de discurso Ω representa a taxa de perda de pacotes em uma rede de transmissão de TV Digital. Nesse caso, como em termos percentuais essa taxa pode variar de 0% a 100%, $\Omega : [0, 100]$. Para esse universo de discurso, podem ser aleatoriamente definidos os conjuntos nebulosos BAIXA, MÉDIA e ALTA, que estão respectivamente associados semanticamente a uma baixa, média e alta taxa de perda de pacotes. As funções de pertinência desses conjuntos

μ_{Alta} , $\mu_{Média}$ e μ_{Baixa} estão representadas na Figura 1. Quando a taxa de perda de pacotes x for igual 5%, por exemplo, essa variável será completamente compatível com o conjunto BAIXA e completamente incompatível com os demais conjuntos. Entretanto, quando a taxa de perda x for igual a 20%, x será parcialmente compatível com o conjunto BAIXA, com grau 0,5 e com o conjunto MÉDIA, com grau 0,5. Nesse caso, pode-se dizer que a taxa de perda de pacotes é “um pouco baixa” e “um pouco média” simultaneamente. Esse processo de determinar o grau de compatibilidade de uma variável $x \in \Omega$ com um ou mais conjuntos nebulosos em Ω é denominado de processo de fuzzificação de x .

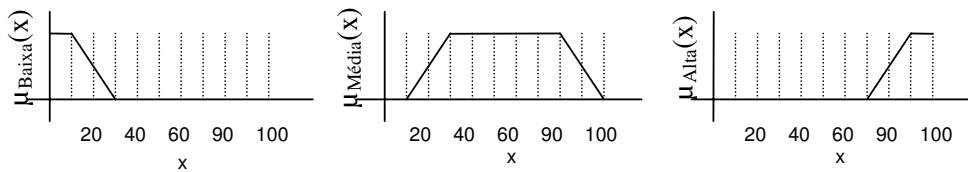


Figura 1 - Funções de pertinência dos conjuntos nebulosos BAIXA, MÉDIA e ALTA, associados a taxas de perdas de pacotes.

Na teoria de conjuntos nebulosos, as operações de complemento, intersecção e união de conjuntos são implementadas através de famílias de operadores matemáticos que preservam algumas propriedades características dessas operações. O complemento de um conjunto nebuloso A , denotado por A^C , é dado por $\mu_{A^C}(x) = n(\mu_A(x))$, onde $n : [0, 1] \rightarrow [0, 1]$, é um operador de negação que satisfaz algumas propriedades matemáticas específicas [Klir 1995]. A operação de complemento mais utilizada é:

$$n(a) = 1 - a.$$

As operações de intersecção e união entre conjuntos nebulosos são implementadas respectivamente pelas famílias de operadores denominadas t-normas $T : [0, 1]^2 \rightarrow [0, 1]$ e t-conorma $\perp : [0, 1]^2 \rightarrow [0, 1]$. Algumas das principais t-normas apresentadas na literatura são: $\min(a, b)$, $a \cdot b$ e $\max(a+b-1, 0)$ e suas t-conormas associadas: $\max(a, b)$, $a + b - a \cdot b$ e $\min(a + b, 1)$. Outras t-normas e t-conormas podem ser encontradas [Klir 1995].

4.1 Lógica Fuzzy

A chamada lógica nebulosa ou lógica fuzzy é a uma maneira de se implementar o raciocínio aproximado no contexto da teoria dos conjuntos nebulosos. Na realidade, a lógica nebulosa nada mais é do que a própria teoria dos conjuntos nebulosos sob um ponto de vista lógico. Nesse contexto, a regra do *modus ponens* da lógica clássica é substituída pela regra do *modus ponens generalizado* [Klir 1995]:

| | |
|-------------------|---|
| $A \rightarrow B$ | Se “perda de pacotes é baixa” então “artefato é adequado” |
| A' | “Perda de pacotes é muito baixa” |
| B' | “Artefato é muito adequado” |

Ao contrário da lógica clássica, não se pode falar realmente em valor-verdade no contexto da lógica fuzzy. No entanto, semelhantemente ao que ocorre com as operações

de complemento, intersecção e união na teoria dos conjuntos nebulosos, as operações lógicas de conjunção, disjunção, complemento e implicação podem ser implementadas através de famílias de operadores matemáticos.

A operação de complemento lógico pode ser implementada pela mesma família de operadores que implementam o complemento de um conjunto fuzzy – $n : [0, 1] \rightarrow [0, 1]$. As operações lógicas de conjunção e de disjunção podem ser implementadas pelos mesmos operadores matemáticos que implementam respectivamente a intersecção e a união de conjuntos nebulosos: $T : [0, 1]^2 \rightarrow [0, 1]$ e $\perp : [0, 1]^2 \rightarrow [0, 1]$.

Já a operação lógica de implicação serve para modelar regras de inferência do tipo “Se <premissa> então <conclusão>”. Essa operação é denotada por $a \in A \rightarrow b \in B$, cujo resultado pode ser obtido através do operador matemático $I(\mu_A(a), \mu_B(b))$, sendo $I : [0, 1]^2 \rightarrow [0, 1]$. Alguns dos principais operadores matemáticos que implementam a implicação são: $\max(1 - a, b)$; $\min(1 - a + b, 1)$; $1 - a + ab$; $\max(1 - a, \min(a, b))$; b^a ; $\min(a, b)$; $a \cdot b$, dentre outros [Klir 1995].

Um mecanismo de inferência baseado em lógica fuzzy deve ter como entrada e como saída um conjunto de variáveis pertencentes aos universos de discursos considerados. Por exemplo, pode-se ter um sistema de inferência fuzzy simples, que tenha como entrada uma única variável x que indica a taxa de perda de pacotes em uma rede de TV digital, $x : [0, 100]$. E como saída uma única variável y que indique numericamente o quão apropriado é o protocolo de transmissão utilizado, $y : [0, 100]$.

Todo sistema de inferência fuzzy deve possuir uma **base de conhecimento**, na qual ficam armazenadas as regras de inferência do tipo: $x_1 \in X_1 \wedge \dots \wedge x_n \in X_n \rightarrow y \in Y$, sendo $X_{1..n}$ e Y conjuntos fuzzy, x_i variáveis de entrada e y uma variável de saída. Após a **fuzzificação** das variáveis de entrada, um **motor de inferência fuzzy** é responsável por, utilizando as regras da base de conhecimento e fazendo uso dos operadores matemáticos que implementam as operações da lógica fuzzy, inferir quais são as variáveis de saída. Por exemplo, dependendo do valor de x e das regras armazenadas na base de conhecimento, o sistema pode inferir que a variável de saída $y \in \text{ADEQUADO}$ com grau de compatibilidade de 0,9, sendo ADEQUADO um conjunto nebuloso. O passo seguinte é então determinar o valor de y dentro do seu universo de discurso Ω_Y . Esse processo é conhecido como **desfuzzificação** e pode ser feito através de operadores matemáticos, tais como *Método do Centro de Área*, *Primeiro Máximo*, *Média dos Máximos*, *Centro da Soma* e *Média dos Máximos Ponderados* [Klir 1995].

Como pôde ser observado nos parágrafos anteriores, um sistema de inferência fuzzy pode ser personalizado através da definição de operadores matemáticos que implementam cada uma das operações da lógica fuzzy, além da operação de desfuzzificação. A adequação ou não de cada operador matemático deve ser estudada caso a caso, de acordo com o domínio de aplicação do sistema.

5. Arquitetura Proposta

A Figura 2 apresenta os principais elementos que constituem a arquitetura proposta para o Sistema de Avaliação de Contextos em Redes de TV Digital. Para fazer uso do processamento subutilizado dos Dispositivos de Recepção de TV Digital – DRTVD, a maioria dos elementos propostos na arquitetura está localizada e deve ser executada no

próprio DRTVD. Assim sendo, pode-se dizer que os DRTVDs provêm para os produtores de conteúdo de TV um serviço de avaliação da adequação de artefatos para o contexto dos seus usuários correntes. Esse serviço é oferecido por uma interface padronizada, a *Interface do Serviço de Avaliação*, através da qual os produtores de conteúdo podem enviar uma descrição de como um determinado artefato deve ser avaliado no DRTVD – a *Descrição de Artefato*, e receber de volta uma indicação numérica do grau de adequação do referido artefato para o contexto dos usuários do DRTVD – a *Avaliação de Artefato*.

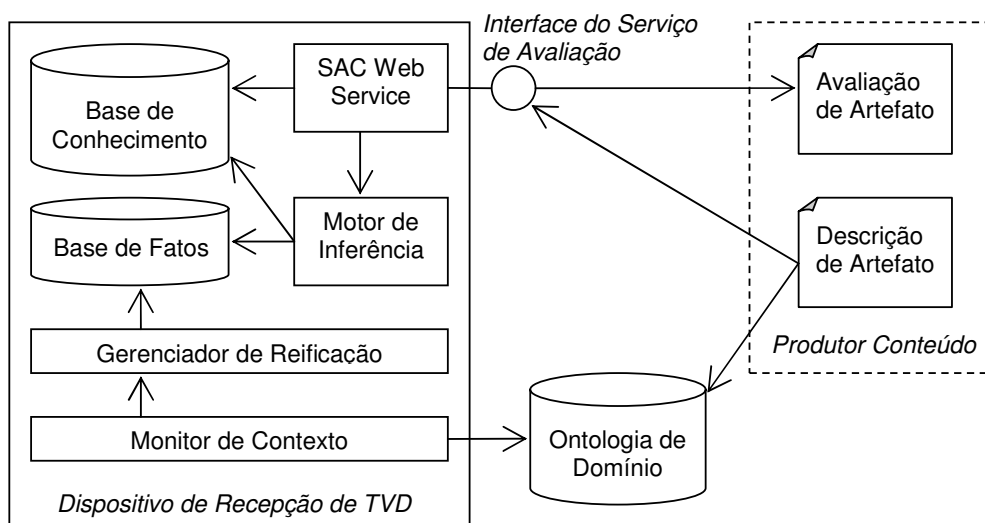


Figura 2 – Arquitetura do serviço de avaliação de contexto

A *Descrição de Artefato* é na realidade um conjunto de regras do tipo: *Se “condição ocorre” então “o artefato é adequado ou não”*. A princípio, poder-se-ia pensar em utilizar uma descrição baseada em lógica de primeira ordem para representar esse conjunto de regras. No entanto, espera-se que o serviço retorne um valor numérico indicando o grau de adequação do artefato para os seus usuários. Além disso, as condições colocadas como hipóteses das regras podem acontecer com maior ou menor intensidade. Por exemplo, numa regra do tipo *Se “sinal de recepção tem ruído” então “artefato é adequado”*, o sinal de recepção pode apresentar muito, pouco ou um nível moderado de ruído. Assim sendo, a conclusão da regra pode ser que o artefato é muito, moderadamente ou pouco adequado. Esse peso associado às hipóteses e às conclusões das regras faz com que descrições baseadas em lógica fuzzy se tornem bastante adequadas para representar o conjunto de regras da *Descrição de Artefato*, sendo essa a abordagem utilizada nesse trabalho.

No contexto das redes de TV Digital, pode ser extremamente caro para um produtor de conteúdo de TV, em termos de utilização de processamento e de banda passante de rede, endereçar cada DRTVD individualmente para solicitar que determinado artefato seja avaliado. Essa solicitação pode ser feita muito mais naturalmente, a todos os DRTVDs simultaneamente através do canal de broadcast. A resposta do DRTVD para o produtor, no entanto, deve dar-se através do seu canal de retorno. Como todos os sistemas de TV digital existentes atualmente especificam o IP

como protocolo de rede para os canais de interatividade, é interessante que a solicitação de avaliação de artefato também seja enviada da emissora para os DRTVDs usando-se o protocolo IP. Isso pode ser feito encapsulando os datagramas IP dentro do protocolo de transporte do sinal de TV, através de um mecanismo denominado *Multi Protocol Encapsulation*. Utilizando-se desse expediente, a comunicação bidirecional entre o produtor de conteúdo e os DRTVDs pode dar-se utilizando a pilha de protocolos TCP/IP.

Considerando-se os tipos de mensagens trocadas entre os produtores de conteúdo e os DRTVDs (basicamente mensagens textuais) e os protocolos de comunicação utilizados (TCP/IP), a tecnologia de *Web Services* se mostra bastante adequada para provisão do serviço de avaliação de contextos em redes de TV digital. Assim sendo, a arquitetura proposta apresenta, do lado do DRTVD, um elemento denominado *SAC Web Service – Serviço de Avaliação de Contexto Web Service* – que é responsável por promover a comunicação entre o DRTVD e o produtor de conteúdo, provendo para esse último uma interface de serviço bem definida.

Sempre que o *SAC Web Service* recebe uma solicitação de avaliação de artefato, tendo como parâmetro uma *Descrição de Artefato*, as regras fuzzy contidas na *Descrição de Artefato* são armazenadas em uma *Base de Conhecimento* presente no DRTVD. Em seguida um *Motor de Inferência* fuzzy deve ser acionado para consultar uma lista de fatos armazenados em uma *Base de Fatos*. Esses fatos representam o contexto de execução do DRTVD. Ou seja, quais os usuários que correntemente estão utilizando o DRTVD, quais suas preferências e intenções, quais as características do DRTVD (processador, memória, canal de retorno, dispositivos de armazenamento e exibição, etc.) e do ambiente no qual ele está inserido (QoS de rede, se o dispositivo está em movimento, etc). De posse dessas informações contextuais, o *Motor de Inferência* pode utilizar as regras armazenadas na *Base de Conhecimento* para determinar o quão adequado é um artefato para o contexto de execução dos telespectadores do DRTVD.

As informações que constituem os contextos dos telespectadores de um DRTVD devem ser continuamente coletadas por um elemento presente no próprio DRTVD, denominado *Monitor de Contexto*. Essas informações devem ser representadas de tal forma que possam ser manipuladas pelo *Motor de Inferência*. Para tanto, o DRTVD conta com um *Gerenciador de Reificação*, que é responsável por receber as informações coletadas pelo *Monitor de Contexto* e gerar uma representação apropriada para as mesmas na forma de fatos a serem armazenados na *Base de Fatos*.

Os fatos presentes na *Base de Fatos* são na realidade variáveis instanciadas de acordo com o contexto de execução do DRTVD. Por exemplo, um fato armazenado na *Base de Fatos* pode ser que a “taxa de perda de pacotes no canal de broadcast” é de 1%. Para que essa informação seja coletada apropriadamente pelo *Monitor de Contexto* e para que os produtores de conteúdo possam gerar *Descrições de Artefatos* que utilizem tal informação, esta deve ser semanticamente definida para que possa ser utilizada de forma uniforme por todos os atores envolvidos. No escopo desse trabalho essa descrição é feita através de uma ontologia presente em uma *Ontologia de Domínio*.

6. Implementação e Resultados de Experimentos

Para validar a arquitetura proposta na seção anterior, todos os seus elementos foram implementados e utilizados em provas de conceito. O primeiro elemento da arquitetura a ser desenvolvido foi uma *Ontologia de Domínio* para TV Digital. É nessa ontologia onde são definidos os principais vocábulos utilizados no âmbito da TV Digital, que podem ser utilizados na instânciação da *Especificação do Artefato*, da *Base de Conhecimento* e da *Base de Fatos*. Uma pequena parte dessa ontologia é ilustrada na Figura 3. Nessa figura, o conceito *User* interage com (interactWith) o conceito *Platform*, que por sua vez dividido em outros conceitos, dentre eles: *AnalogicTelevision* e *DigitalTelevision*. No escopo dos sistemas de TV digital, o conceito de *AnalogicTelevision* deve estar conectado a um *SetTopBox*, isso porque uma TV analógica não é compatível com o sinal digital. *DigitalTelevision* e *SetTopBox* são conceitos similares por possuírem um grande número de propriedades em comum.

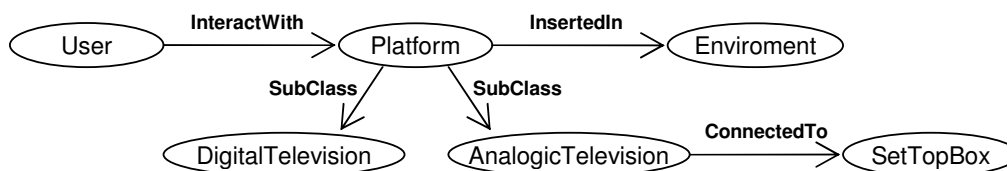


Figura 3 - Alguns conceitos presentes na ontologia de domínio para TV digital

A ontologia de domínio para TV Digital foi especificada na linguagem OWL utilizando-se a ferramenta Protegé¹. A ontologia conta com 62 classes, 40 propriedades e 10 relacionamentos. As classes da ontologia são classificadas em três grupos principais: aquelas relacionadas a conceitos relativos aos *usuários* (telespectadores), à *plataforma* e ao *ambiente*. Os conceitos relacionados aos usuários são utilizados para que se possa caracterizar quem são os telespectadores que correntemente estão utilizando o DRTVD e quais são as suas preferências. Os conceitos relacionados à plataforma servem para que se possa identificar qual o estado corrente do DRTVD (utilização de CPU, memória, dispositivo de armazenamento, resolução do dispositivo de exibição, tipo de canal de retorno, canal sintonizado, dentre outras). Já os conceitos relacionados ao ambiente servem para caracterizar o local no qual o DRTVD está inserido, por exemplo, determinando se o mesmo encontra-se em movimento ou parado, quais os protocolos de comunicação utilizados, qual o nível de QoS da rede, etc. Os conceitos presentes na ontologia foram identificados através de consulta a 30 pesquisadores de TV Digital que trabalharam no desenvolvimento do Sistema Brasileiro de TV Digital. Uma descrição detalhada dessa ontologia está fora do escopo deste trabalho.

Uma vez definidas através da ontologia de domínio, as variáveis que compõem o contexto dos telespectadores podem ser monitoradas nos DRTVDs pelo *Monitor de Contexto* e representadas na *Base de Fatos* pelo *Gerenciador de Reificação*. Neste trabalho a *Base de Fatos* foi implementada na forma de um arquivo de instância OWL. Essa foi uma escolha natural, tendo em vista que a ontologia foi especificada em OWL.

¹ Essa ontologia está disponível no endereço eletrônico <http://www.lavid.ufpb.br/ontotvd/ontoTVD.zip>

O *Monitor de Contexto* foi implementado em C++, visando aumentar a eficiência do processo de monitoramento, possibilitando a realização de chamadas de sistema para acessar informações sobre a plataforma, ambiente e usuários. Uma vez coletadas, essas informações são repassadas ao *Gerenciador de Reificação* para que possam ser armazenadas na *Base de Fatos*. O *Gerenciador de Reificação* foi implementado através do *framework* de programação denominado Jena. Esse *framework* provê, dentre outras ferramentas, uma API que permite a realização de consultas e modificações em modelos em OWL. Utilizando o Jena é possível verificar quais são os conceitos presentes na ontologia e instanciá-los, associando valores às suas propriedades. Por exemplo, uma informação da *Base de Fatos* pode ser que a variável “usuario.preferencias.generoMusical” possui um valor referente ao conceito “rock”.

Após povoar a *Base de Fatos* com informações contextuais, novas informações podem ser obtidas a partir de um sistema de inferência. Para implementar o serviço de avaliação de contexto, foi utilizado um sistema de inferência fuzzy presente no ambiente de programação matemática *Matlab*. Esse sistema é disponibilizado no ambiente de programação através de um componente denominado *Fuzzy Logic Controller*, que encapsula o *Motor de Inferência* e a *Base de Conhecimento*, previstos na arquitetura do serviço de avaliação de contexto. Esse componente apresenta métodos que possibilitam a especificação das regras que constituem a *Base de Conhecimento* e os parâmetros que determinam o funcionamento do *Motor de Inferência*. As regras da *Base de Conhecimento* devem ser do tipo $x_1 = X_1 \wedge \dots \wedge x_n = X_n \rightarrow y = Y$ ou $x_1 = X_1 \vee \dots \vee x_n = X_n \rightarrow y = Y$, onde $x_{1..n}$ São variáveis de entrada (variáveis do contexto), y é uma variável de saída (informação inferida pelo sistema), $X_{1..n}$ e Y são conjuntos fuzzy. Já os parâmetros que determinam o funcionamento do *Motor de Inferência* são os operadores matemáticos que implementam as operações em lógica fuzzy: And, Or e Implicação, bem como as operações de Agregação e Desfuzificação. As regras da *Base de Conhecimento* e os parâmetros que determinam o funcionamento do *Motor de Inferência* podem ser representados em um arquivo *FIS – Fuzzy Inference System*, cuja sintaxe é definida pelo ambiente *matlab*, que é passado como parâmetro para o componente *Fuzzy Logic Controller*.

Toda variável acessada pelo *Motor de Inferência* deve possuir um valor numérico. Isso se deve ao fato de os universos de discurso Ω de todas as variáveis de entrada consideradas pelo *Motor de Inferência* do *Fuzzy Logic Controller* serem numéricos. Assim sendo, antes de serem armazenadas na *Base de Fatos*, os valores de cada variável de contexto precisam ser mapeados em valores numéricos. Por exemplo, pode-se convencionar que a variável “usuario.preferencias.generoMusical” terá valor 7, quando o usuário gostar de “rock”. Nesse caso, as variáveis de contexto armazenadas na *Base de Fatos* poderão ser acessadas através da chamada *double getVarValue (char * varName)* constante na interface da *Base de Fatos*.

Para que um produtor de conteúdo possa avaliar o grau de adequação de um artefato para um determinado contexto, ele precisa fornecer uma *Descrição do Artefato* que especifique como o mesmo deve ser avaliado. Mais precisamente, a *Descrição do Artefato* deve conter uma lista de regras em lógica fuzzy para serem colocadas na *Base de Conhecimento*, um conjunto de parâmetros que especifiquem como o *Motor de Inferência* fuzzy deve funcionar e uma lista das variáveis do contexto utilizadas pelas regras fuzzy. Na implementação atual do serviço de avaliação de contexto, as duas

primeiras informações (regras fuzzy e parâmetros do *Motor de Inferência*) devem ser fornecidas em um arquivo FIS, obedecendo a sua sintaxe especificada pelo ambiente *matlab*. Esse arquivo é passado como parâmetro juntamente com a lista de variáveis de contexto utilizadas em uma chamada Web Service que implementa o serviço de avaliação de contexto. Essa chamada deve retornar o grau de adequação do artefato para o contexto em questão. A *Interface do Serviço de Avaliação* apresenta então o seguinte método para avaliação de artefato:

```
double eval (char *fisFile, char ** variables, int numVariables)
```

Onde, *fisFile* representa o conteúdo do arquivo no formato FIS, *variables* representa o nome das variáveis contextuais utilizadas como entrada no sistema de inferência fuzzy, e *numVariables*, o número de variáveis armazenadas na estrutura *variables*.

6.1 Prova de Conceito

Objetivando verificar a viabilidade de utilização do serviço de avaliação de contexto em um ambiente real de TV digital, o serviço foi instalado e testado em um *set-top box* Kalaheo, com sistema operacional Linux e *middleware* Ginga. Os testes foram realizados na rede de TV digital do Laboratório de Aplicações de TV Digital – LAVID da Universidade Federal da Paraíba, recebendo o sinal de TV digital através de um canal de *broadcast* de 6 MHz e canal de retorno baseado em modem PTSN. Em uma prova de conceito simples foi elaborada uma *Descrição de Artefato* contendo regras fuzzy que utilizavam duas variáveis de contexto: *dispositivo.resolução* e *vídeo.resolução*, referentes respectivamente à resolução do dispositivo de exibição e do sinal de vídeo recebido, representada em termos do número de linhas. Nesse caso, cada variável pode assumir um valor numérico entre 0 e 1920. Após serem fuzificadas as variáveis podem pertencer a três conjuntos fuzzy: LDTV, SDTV e HDTV, que representavam vídeos de resolução baixa, padrão ou alta. A variável de saída pode assumir valores de 0 a 100, de tal forma que quanto maior o valor da variável de saída, mais adequado é o artefato para o contexto de execução. Antes da desfuzificação a variável de saída pode pertencer aos conjuntos fuzzy: ADEQUADO, REGULAR ou INADEQUADO. Assim sendo, as seguintes regras fuzzy foram especificadas no experimento realizado:

```
Se dispositivo.resolução = LDTV e vídeo.resolução = LDTV então artefato é ADEQUADO
Se dispositivo.resolução = LDTV e vídeo.resolução = SDTV então artefato é REGULAR
Se dispositivo.resolução = LDTV e vídeo.resolução = HDTV então artefato é INADEQUADO
Se dispositivo.resolução = SDTV e vídeo.resolução = LDTV então artefato é REGULAR
Se dispositivo.resolução = SDTV e vídeo.resolução = SDTV então artefato é ADEQUADO
Se dispositivo.resolução = SDTV e vídeo.resolução = HDTV então artefato é REGULAR
Se dispositivo.resolução = HDTV e vídeo.resolução = LDTV então artefato é INADEQUADO
Se dispositivo.resolução = HDTV e vídeo.resolução = SDTV então artefato é REGULAR
Se dispositivo.resolução = HDTV e vídeo.resolução = HDTV então artefato é ADEQUADO
```

Os testes foram executados em *set-top box* Kalaheo, utilizando diversos valores para as variáveis *dispositivo.resolução* e *vídeo.resolução*. Nos resultados obtidos observou-se que quando o DRTVD e o vídeo recebido possuíam a mesma resolução, o grau de adequação do artefato retornado pelo serviço de avaliação de contexto era máximo. Quanto maior a diferença entre as resoluções do DRTVD e do vídeo recebido, menor foi o grau de adequação do artefato. Como esse era o objetivo da *Descrição do Artefato*, considerou-se que o experimento foi bem sucedido. Outros experimentos envolvendo um número maior de variáveis foram realizados. Em todos os casos foi

possível expressar através das regras fuzzy, o contexto para o qual um dado artefato é adequado e verificar, através do serviço de avaliação de contexto o quão adequado é o artefato para determinados contextos dos usuários. Uma descrição mais detalhada desses experimentos e uma análise mais detalhada do poder de expressividade da lógica fuzzy para descrição de contextos adequados para artefatos computacionais são deixados para trabalhos futuros.

7. Trabalhos Relacionados

Alguns trabalhos disponíveis na literatura especificam mecanismos que possibilitam indicar a adequação de composições de artefatos, como componentes de software, por exemplo, para determinados contextos, considerando-se aspectos não-funcionais.

O Aster [Issarny 1998] é um ambiente para interconexão de componentes de software que utiliza o modelo de programação baseado em configuração para a construção de middlewares orientados a componentes. Os componentes do middleware devem ser configurados de forma a atender aos requisitos das aplicações que serão executadas nele. No ambiente Aster, os requisitos não-funcionais das aplicações e os supridos por cada componente do middleware são especificados através de formulações em lógica de primeira ordem. Dessa forma, um provador lógico pode ser utilizado para encontrar uma configuração de middleware que atenda a todos os requisitos das aplicações. No entanto, por utilizar formulações em lógica de primeira ordem, o ambiente Aster é apenas capaz de dizer se uma dada configuração de componentes é adequada ou para uma aplicação. Ele não é capaz de indicar se uma dada configuração é mais ou menos adequada com base na intensidade dos requisitos não-funcionais.

Turkay [Turkay 2004] apresenta um esquema baseado em engenharia de sistemas orientado a modelos para endereçar o problema de reconfiguração de middlewares orientados a componentes. Esse trabalho apresenta uma linguagem denominada *Options Configuration Modeling Language (OCML)* que pode ser utilizada para descrever a compatibilidade entre os componentes e as dependências entre os valores assumidos pelos parâmetros de configuração do middleware. A linguagem define regras de dependência que restringem a disponibilidade das combinações de opções de configuração do middleware, descritas através de operadores lógicos como “E”, “OU”, “NÃO” e “SE SOMENTE SE”. Apesar de a *OCML* conseguir restringir as possíveis configurações do middleware com base em seus componentes, ela é apenas capaz de indicar se uma determinada configuração é válida ou não, não sendo capaz de informar se uma configuração é melhor ou pior do que outra.

8. Conclusões

Este trabalho apresentou a arquitetura e a implementação de um serviço para avaliação de contexto em redes de TV digital, que pode ser utilizado por provedores de conteúdo televisivo para avaliar o quão apropriado é um determinado conteúdo para o contexto em que seus telespectadores encontram-se. As principais características do serviço de avaliação de contexto são:

- Definição formal das informações que caracterizam o contexto de um telespectador através de uma ontologia de domínio para TV digital;

- Implementação distribuída do serviço nos próprios receptores de TV digital, utilizando a capacidade de processamento ociosa destes dispositivos;
- Provisão do serviço na forma de Web Services em redes de TV digital, onde a solicitação do serviço pode ser enviada para todos os receptores simultaneamente através do canal de *broadcast* e o resultado retornado através dos canais de retorno de cada dispositivo;
- Utilização de um mecanismo de inferência baseado em lógica fuzzy para determinação do grau de adequação de um conteúdo televisivo para o contexto dos seus telespectadores.

O serviço de avaliação de contexto em redes de TV digital foi instalando e testado em um receptor de TV digital Kalaheo. Os testes foram realizados na rede de TV digital do Laboratório de Aplicações de Vídeo Digital da Universidade Federal da Paraíba. Nesses testes foram utilizadas diversas especificações em lógica fuzzy, indicando como determinados conteúdos deveriam ser avaliados. Todos os testes apresentaram resultados satisfatórios.

Apesar de ter sido possível nos testes realizados representar os contextos para os quais determinados artefatos são ou não adequados, uma análise mais profunda do poder de expressividade da lógica fuzzy para representação de contextos é deixada como trabalho futuro.

Referências Bibliográficas

- Bechhofer, S. et al. (2004) "Web Ontology Language (OWL) Reference", W3C Recommendation.
- Brézillon, P. (1999) "Context in Problem Solving: A survey", in: The Knowledge Engineering Review, v. 14, n. 1, pp. 1-34.
- Dey, A. and Abowd, G. (2000) "Towards a Better Understanding of Context and Context-Awareness" in: Proceedings on the CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness.
- Gruber, T. (1993) "A Translation Approach to Portable Ontologies", in: Knowledge Acquisition, v. 5, n. 2, pp. 199-220.
- Issarny V., Bidan C. and Saridakis T. (1998) "Achieving Middleware Customization in a Configuration-based Development Environment: Experience with the ASTER Prototype", in Proceedings of the International Conference on Configurable Distributed Systems, 207-214.
- Klir, G. and Yuan, B. (1995) "Fuzzy Sets and Fuzzy Logic: Theory and Applications", Prentice Hall, New Jersey, 574p.
- Turkay, E., Gokhale, A. and Natarajan, B. (2004) "Addressing the middleware configuration challenges using model-based techniques", in ACM Southeast Regional Conference, 166-170.