

7

Trabalhos Relacionados

Durante o desenvolvimento deste trabalho, diversas referências foram analisadas nas áreas de computação ubíqua, trabalho colaborativo e tecnologias multimídia. Na área de computação ubíqua estávamos particularmente interessados na configuração, comunicação, sincronia e adaptabilidade das unidades de processamento para a reprodução das apresentações multimídia.

O projeto Gaia [25, 26], desenvolvido pela Universidade de Illinois em Urbana-Champaign, é um meta-sistema operacional baseado no paradigma de computação ubíqua. Seu objetivo é oferecer serviços similares aos de um sistema operacional para suportar aplicações distribuídas em diferentes e múltiplos dispositivos e que interagem para a execução de uma determinada atividade. *Gaia* oferece facilidades como busca de dispositivos através de suas características, reserva de recursos computacionais, comunicação entre computadores, detecção de erros e sistema de arquivos. Em nosso trabalho, buscamos utilizar os resultados conceituais de Gaia, porém optamos por implementações simplificadas da maioria dos serviços oferecidos. Como exemplo, temos o *Disparador* (Seção 4.1.5), que é capaz de iniciar e finalizar processos remotamente, o *Sistema de Arquivos Distribuído* (Seção 4.1.4), que permite a troca de dados entre diferentes dispositivos computacionais, e o *Serviço de Nomes* (parte do padrão CORBA), que possibilita encontrar os diferentes grupos de aplicações em execução.

Um projeto similar ao *Gaia* é o *Aura* [27]. *Aura* também oferece serviços que auxiliam a execução de aplicações que seguem o modelo de computação ubíqua. Alguns subprojetos de *Aura* são: *Odyssey*, que faz monitoramento de recursos e automaticamente adapta aplicações de acordo; *Coda*, que provê acesso a arquivos em um ambiente com desconexão; *Spectra*, um mecanismo de execução remota; e finalmente, *Prism*, responsável pela captura e manipulação da intenção do usuário, provendo suporte à pró-atividade e adaptação. *ActivePresentation* oferece um suporte simples à desconexão pois permite, por exemplo, excluir e incluir um *Nó de Apresentação* a um grupo em tempo de execução. Além disso, um sistema mais

elaborado de detecção de desconexão poderia ser implementado através de um mecanismo de *heartbeat*. Nota-se que este e outros serviços (e.g. captura de intenção do usuário e adaptação) também podem existir dentro da infra-estrutura *ActivePresentation*.

Uma terceira infra-estrutura para a construção de sistemas de computação ubíqua é o *Event Heap* [36, 51, 52], que através de um sistema de comunicação por tuplas permite o desacoplamento do transmissor e receptor das mensagens.

Além desses sistemas de computação ubíqua, estávamos interessados em analisar as funcionalidades oferecidas por sistemas de trabalho colaborativo síncrono à distância, que de fato se assemelham a uma apresentação multimídia distribuída. Tais sistemas também precisam tratar de questões como configuração remota das unidades de processamento, replicação de conteúdo, sincronia e adaptabilidade da apresentação.

Distinguimos as soluções para trabalho colaborativo síncrono em dois grupos: as naturalmente distribuídas e as que duplicam a execução de aplicações *single-user* [32]. Como exemplos dos sistemas naturalmente distribuídos temos os aplicativos para envio de mensagens instantâneas na Internet e a World Wide Web.

A Microsoft Research, por exemplo, possui um projeto de pesquisa de aplicações naturalmente distribuídas para trabalho colaborativo chamado *ConferenceXP* [37]. Este sistema oferece, além de videoconferência grupo-a-grupo, um conjunto de ferramentas como o *real-time inking*, um quadro virtual para desenho em grupo, e o *PowerPoint Presentation*, para o acompanhamento de uma apresentação distribuída através de imagens geradas por ele.

Já os sistemas de reprodução da execução de aplicações *single-user* podem ser divididos em duas categorias: os sistemas centralizados, nos quais as aplicações são executadas em uma máquina e transmitem sua saída gráfica para múltiplos dispositivos, e os sistemas de replicação, em que aplicações *single-user* executadas localmente são sincronizadas entre diferentes máquinas. Alguns sistemas comerciais utilizam a arquitetura centralizada, como é o caso do NetMeeting e do SunForum. Essa solução, apesar de ser mais facilmente implementada, oferece uma interação com o usuário lenta e, em caso de desconexão com o servidor, invalida imediatamente o cliente. Além disso apresenta problemas de escalabilidade.

O projeto *Access Grid* [15, 53] apresenta um sistema para interação grupo-a-grupo envolvendo videoconferência e uso de aplicações replicadas como visualizadores de imagem, vídeo, PowerPoint® e navegadores Web.

Este projeto apresenta um forte enfoque na qualidade da videoconferência e, até onde sabemos, não apresenta uma infra-estrutura elaborada para a execução das aplicações em grupo.

O *ABC Framework*, descrito por J. Bardram [43], enfoca a construção de um framework para a utilização de aplicações replicadas em trabalho colaborativo. Similarmente ao presente estudo, o *ABC Framework* se inspira em trabalhos que descrevem ambientes de computação ubíqua [26, 27]. Além disso, ele pesquisa o uso de aplicações heterogêneas durante a realização de uma mesma atividade. Nos diferenciamos dessa abordagem por acreditar que o controle de grupos de aplicações homogêneas possui um maior apelo prático devido à sua relativa simplicidade.

Em uma linha diferente e enfocando apresentações para platéias em uma mesma localidade, o trabalho de Stephen Pope [38] descreve um sistema de controle distribuído de aplicações que manipulam imagem e som com garantias de alto desempenho. Como diferença principal, esse trabalho possui uma maior ênfase no desempenho dos componentes envolvidos e uma menor ênfase na integração entre eles.

Outros estudos recentes em sistemas multimídia também possuem uma forte relação com este trabalho. Por exemplo, o projeto *Hourglass* [59] (ou ampulheta) descreve um sistema de composição de conteúdo multimídia em ambientes de computação ubíqua através de um modelo onde o conteúdo é retirado de várias origens, processado dependendo de seleções ou interesse do usuário, e então distribuído para diferentes dispositivos (por isso o nome ampulheta). Um trabalho similar, porém com um enfoque maior na distribuição de serviços, é o *SpiderNet* [60], que trata com maior detalhe escalabilidade, flexibilidade e gerência de qualidade de serviço. Na infra-estrutura *ActivePresentation*, qualidade de serviço e compressão de dados, por exemplo, ajudariam a assegurar o início rápido de uma apresentação baseando-se em informações sobre o desempenho computacional e a velocidade de comunicação entre computadores.

Apesar dos diversos estudos na área de multimídia, no escopo deste trabalho estávamos particularmente interessados em aproveitar resultados obtidos na sincronização das múltiplas unidades de processamento para a reprodução de aplicações em sincronia. A construção do módulo *Gerente de Grupo* (Seção 4.1.2) é resultado deste estudo baseado no trabalho de Diwakar Gupta [31].

Para a coordenação da execução de uma apresentação, observamos algumas soluções comerciais que combinavam o uso de diferentes mídias. Algumas delas, como o *Online Presentations* [20], o *Presentation Stu-*

dio [47], o Accordent's PresenterOne [54] e o StreamAuthor [48], permitem a sincronização de áudio e vídeo com o PowerPoint®. Alguns desses sistemas oferecem mecanismos de autoria que através de um documento permitem que uma apresentação seja armazenada e reproduzida. Todavia, essas soluções diferem deste trabalho pois buscávamos soluções abertas para o armazenamento da apresentação e buscávamos um formato capaz de facilmente integrar novos tipos de mídia ao documento. Além disso, esses sistemas foram desenvolvidos para execução em apenas um dispositivo computacional, não para ambientes de computação ubíqua. Desta forma, analisamos os formatos SMIL e NCL, como foi descrito respectivamente nas seções 5.2 e 5.3.