



DI - DEPARTAMENTO DE INFORMÁTICA

Controle e interação em salas inteligentes

*Rafael Ramos de Oliveira¹,
Renato Fontoura de Gusmão Cerqueira².*



¹ *Aluno de Graduação do curso de Engenharia de Computação da PUC-Rio.*

² *Eng. De Computação, Professor do Departamento de Informática da PUC-Rio.*

SUMÁRIO

1. INTRODUÇÃO.....	3
2. METODOLOGIA.....	3
3. TRABALHOS FUTUROS.....	6
4. CONCLUSÕES.....	8
REFERÊNCIAS.....	9

1. INTRODUÇÃO

Avanços computacionais evidentes, como maior poder de processamento e armazenamento, miniaturização, interconectividade, maior capacidade de baterias, redes sem fio e tecnologias de software, vêm tornando o computador cada vez mais integrado à vida cotidiana, e um modelo a que chamamos Computação Ubíqua vem sendo vislumbrado há alguns anos.

Cada vez mais freqüentemente as salas de trabalho estão sendo dotadas de muitos e variados dispositivos computacionais. Planejadas para reuniões, trabalhos em grupo e conferências, essas salas podem executar apresentações multimídia em múltiplos dispositivos compostas por vídeo, som, imagens estáticas, páginas Web ou até mesmo aplicações convencionais que não estejam diretamente ligadas a conteúdo multimídia. Entretanto, essas salas ainda sofrem da ausência de integração dos seus recursos, o que dificulta a sua utilização e acaba por desperdiçar o potencial que seu uso em conjunto traria.

Este trabalho de iniciação científica está inserido no contexto do projeto Active Classroom, desenvolvido no Departamento de Informática da PUC-Rio, em colaboração com grupos de pesquisa de outras instituições, tais como Microsoft Research, Universidade de Illinois em Urbana-Champaign e IME-USP. O projeto Active Classroom tem como objetivo prover uma infra-estrutura de software que facilite o desenvolvimento de aplicações que tirem proveito efetivo dos diversos recursos computacionais e de interação disponíveis nesses novos ambientes de trabalho. Este projeto focará inicialmente na construção de salas de aula inteligentes, podendo futuramente ser ampliado para outros tipos de ambientes de trabalho em grupo.

Visto tal contexto, este trabalho concentrou-se no estudo do controle e nas formas de interação com dispositivos eletrônicos visando o uso em salas de aula. Ao final deste trabalho estará completo um produto que disponibiliza uma API para que programadores tenham o controle da infra-estrutura de tal sala.

O estudo foi realizado no laboratório de mesmo nome do projeto, Active Classroom, montado graças à parceria do Departamento de Informática e a Microsoft Research. O laboratório possui tecnologia de ponta e toda infra-estrutura necessária para o desenvolvimento do projeto.

2. METODOLOGIA

O trabalho pode ser dividido primeiramente em quatro tópicos, todos explicados na evolução desse relatório. Cada um dos quatro tópicos lida com uma variedade de tecnologias, o estudo de cada tecnologia pode ser chamado então de etapa.

Cada etapa do trabalho pode ser dividida em dois momentos: um primeiro momento, e também o mais extenso, consistia na pesquisa da tecnologia em questão. Um segundo momento seria a implementação de protótipos usando a tecnologia que foi aprendida.

As tecnologias usadas no trabalho serão explicadas na evolução desse relatório de forma mais precisa, mostrando também suas importâncias e onde foram usadas. Cabe observar que este trabalho de iniciação científica teve um caráter maior de aprendizado do que de produção.

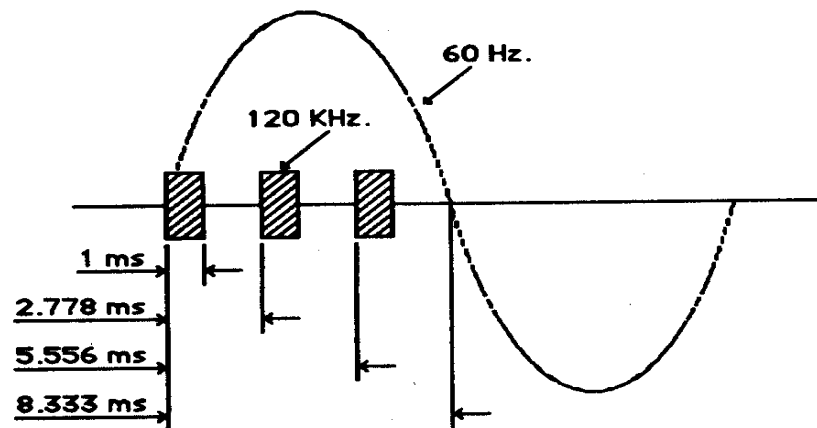
Segue abaixo os tópicos estudados e uma explicação para as tecnologias envolvidas:

- *Construção de controladores de dispositivos*

Como uma parte significativa do trabalho consiste no controle de dispositivos, houve o estudo dos aparelhos eletrônicos pertencentes a sala. Tendo então o conhecimento dos

recursos de tais aparelhos e de suas interfaces foi possível implementar drivers para controle por software de suas funções.

Para o controle da rede elétrica e sensores de presença, foram utilizados dispositivos que se comunicam através da rede elétrica e/ou por rádio-freqüência obedecendo o protocolo X10. No caso da rede elétrica, este protocolo utiliza pulsos de 120Khz nas raízes do sinal elétrico para representar o '1' binário, e a ausência deste pulso para indicar o '0' binário. Na figura abaixo uma possível comunicação é exemplificada. O sinal de 60Hz obviamente é descartado por um filtro passa-altas para que tenhamos somente a informação da comunicação na momento da interpretação do sinal.



Os retângulos indicam presença de pulsos de 120KHz. Nesse período temos a string '10' pois somente o primeiro pulso foi considerado (os outros não coincidem com raízes do sinal de 60Hz) e a segunda raiz não apresenta um pulso.

Os dispositivos foram adquiridos através de empresas como Active Home e Home Seer. Para a comunicação com um computador são utilizados tipos especiais de dispositivos, chamados aqui de dispositivos-interface. Os modelos dos dispositivos-interface utilizados foram o CM11A, que permite conexão com a porta serial, e o CM15A, que permite conexão com a porta USB. Para o controle por software de tais dispositivos-interfaces, as empresas disponibilizaram objetos COM[1].

Com posse dessa tecnologia, foi possível então instalarmos na sala tais dispositivos eletrônicos. Através dos já mencionados dispositivos-interface, foi construído um protótipo capaz de controlar os dispositivos instalados nas saídas da rede elétrica. Este protótipo foi desenvolvido utilizando a linguagem Lua, com apoio da biblioteca LuaCOM, para a manipulação do objeto disposto pelas interfaces.

Com isso é possível ligar ou desligar as luzes da sala por controle de software. Como a sala só possui lâmpadas frias, o controle da diferença de potencial (para dimerização) não pode ser usado, porém também foi implementado.

A tecnologia X10 apresenta porém problemas de comunicação em caso de linhas de transmissão ruidosas já que erroneamente pulsos de 120Khz podem acontecer. Além disso, alguns aparelhos eletrônicos retiram os pulsos desejados da rede elétrica, impedindo que dois outros aparelhos se comuniquem. Também há a impossibilidade de comunicação para aparelhos em fases distintas. Utilizando rádio freqüência na camada física os dispositivos não

apresentaram problemas significativos, mas devido aos problemas relativos à rede elétrica, um passo futuro para o trabalho seria o estudo de uma outra tecnologia de automação para uso real da sala.

Para o cenário de uma sala de aula, viu-se também como interessante o uso de câmeras para acompanhar as aulas ou gravar eventos e reuniões. Foi estudado então o controle de câmeras PTZ[2] da empresa D-Link modelo DCS6610G.

Tais câmeras possuem um servidor Web capaz de receber solicitações pelo protocolo HTTP[3]. Inicialmente o trabalho não possuía nenhuma forma de controle desta câmera, visto que a empresa só disponibilizava programas de código fechado para o controle. Utilizando então a ferramenta Wireshark para a captura de pacotes da rede foi compreendido a forma de comunicação, e utilizando a linguagem Lua mais uma vez, agora com o auxílio da biblioteca LuaSocket, um outro protótipo foi criado para sala, este capaz de controlar o movimento e capturar imagens destas câmeras.

Ao final do ano de trabalho, a empresa disponibilizou o SDK do dispositivo, também na forma de um objeto COM. O trabalho foi modificado para usar este objeto já que o método anterior era menos preciso.

Ainda há outros controladores de dispositivos a serem implementados (projetores, receivers, etc.), apesar de alguns ainda terem sido estudados, o caráter acadêmico já não possuía um valor significativo. Portanto, o trabalho passou a focar o estudo de serviços para auxiliar o uso dos controladores de dispositivos.

- *Serviços de auxílio*

Visando um sistema maior, foi estudado alguns mecanismos que oferecessem robustez, segurança e organização para auxiliar esse aumento na escalabilidade.

Primeiramente há a necessidade de uniformizar a forma de referência aos dispositivos. Dadas as várias tecnologias diferentes, os dispositivos apresentam endereços não só obviamente diferentes, mais de naturezas diferentes. As câmeras por exemplo são determinadas por endereços do protocolo TCP/IP, já os dispositivos X10 seguem os endereços do protocolo X10. Um modelo de endereçamento que relacione os endereços reais a um conjunto de endereços mais uniforme se mostrou necessário.

Levando em consideração também a segurança, um sistema de controle de acesso se tornou necessário. Os controladores de dispositivo são controlados em última instância por usuários, seja o leigo com acesso direto através de uma interface, ou o programador com seu software. Em qualquer caso, saber qual usuário está utilizando os recursos é importante para que possamos limita-lo, e dessa forma determinar quem usa e quais dispositivos usa. Ainda podemos ter um controle de ações para saber o que o usuário fez, para isso um sistema de controle de eventos foi criado conjuntamente.

O controle da seqüência de eventos se mostrou importante não só para o mencionado acima, mas também para podermos ter um registro dos acontecimentos da sala. Caso algum dispositivo apresente problema, podemos verificar nesse log de eventos qual a última vez que ele funcionou, qual foram suas últimas ações e de que forma ele interagiu com outros dispositivos. Isso facilitaria a análise de problema.

Os serviços mencionados acima formam implementados, porém a nível de aprendizado. Em outros projetos coordenados pelo orientador deste trabalho há implementações mais completas e genéricas o suficiente para a fusão com o produto dessa iniciação científica. Logo, no produto final não serão usadas as implementações feitas.

Serviços de interesse continuam a ser analisados e protótipos são implementados a medida que se vê necessário testá-los.

- *Interface com o usuário*

Além de se ter em mente a API a ser exportada, seria interessante o sistema já ter em si alguma forma de interação com o usuário comum para que consigamos um uso mais interno deste. Dado o forte cenário da Internet que temos hoje, houve a criação de uma interface Web para a comunicação visual com o usuário.

Com isso, o estudo necessitou de uma tecnologia que nos possibilitasse o pré-processamento de uma página Web utilizando uma linguagem específica para que assim tenhamos a possibilidade de comandar os controladores de dispositivos por uma interface mais simples e clara para o usuário leigo.

A tecnologia CGI (Common Gateway Interface) nos permite fazer isso, mais especificamente, foi estudado e usado no trabalho o CGI Lua, já que o trabalho possui em sua maior parte códigos implementados na linguagem Lua.

Outras formas de contato com o usuário como sintetização de voz e reconhecimento de movimentos estão sendo estudadas, mas devido a dificuldade de suas soluções resultados só seriam apresentados com um espaço de tempo maior.

- *Distribuição dos componentes do sistema*

Até agora todo o sistema encontra-se concentrado em um único servidor. Mas com isso estamos obrigando a um só computador estar conectado a todos os dispositivos fisicamente. Isso não é interessante, já que além de ser uma concentração grande de processamento, um só computador possuiria limitações físicas de conexão. Os serviços auxiliares também deveriam estar sendo executados na mesma máquina, aumentando ainda mais a carga de processamento.

Utilizando então a arquitetura SCS[4] podemos mudar esse quadro. Os controladores de dispositivos não mais estarão numa só máquina, possibilitando uma quantidade de conexões físicas a dispositivos limitadas agora não mais pela quantidade de portas de um computador, mas sim pela quantidade de portas de um conjunto de computadores.

Cada um dos serviços podem estar em máquinas diferentes também. Em função do interesse nos recursos cada serviço pode estar em uma máquina que tenha uma ênfase em tal. Por exemplo, o serviço de logs podem ser instanciado em máquinas com capacidade de armazenamento maior.

Dessa forma, alguns protótipos criados das seções anteriores, foram modificados para terem características de componentes de software definidas pela arquitetura SCS.

[1] *Component Object Model*: tecnologia de componentes de software da família de sistemas operacionais Windows®

[2] *Pan-Tilt-Zoom*: significa possibilidade de movimentos para os lados, para cima, para baixo, aumentar e diminuir o controle de zoom

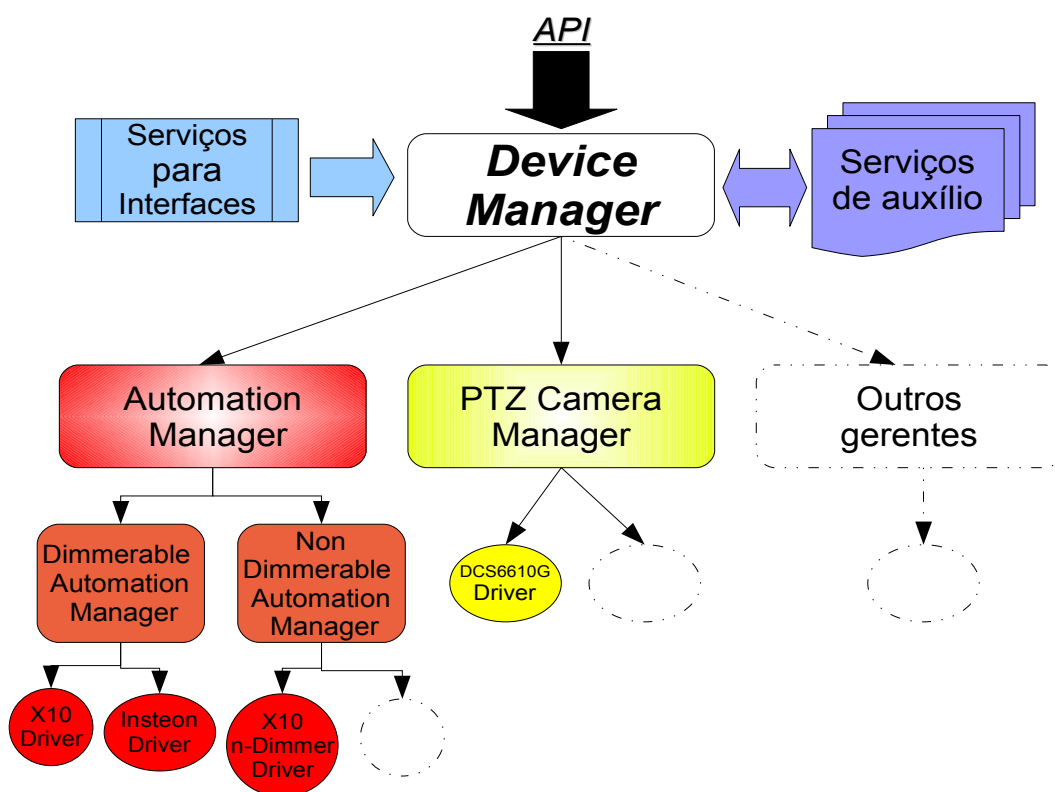
[3] *Hypertext Transfer Protocol*: protocolo da camada de aplicação do modelo de protocolos TCP/IP (Transmission Control Protocol/Internet Protocol)

[4] *Software Component System*: sistema de componentes de software para a arquitetura CORBA (Common Object Request Broker Architecture) cujo objetivo é prover uma infra-estrutura leve e simples para distribuir, instanciar e executar aplicações baseadas em componentes de software

3. TRABALHOS FUTUROS

O presente trabalho apresentou um caráter de aprendizado muito maior do que de produção, visto a quantidade de tecnologias que deveriam ser dominadas. Esse primeiro ano possibilitou assim uma melhor visão de como a idéia inicial deste trabalho poderia ser implementada e que cenários esse possibilitaria.

A estrutura atual do trabalho será explicada a seguir. Considere a figura abaixo:



Cada elemento mostrado acima desempenha também o papel de uma componente de software definido pela arquitetura SCS, o que possibilita de antemão a modularidade do sistema e de instalações de componentes em tempo de execução.

No nível mais abaixo da figura estão os controladores de dispositivos. Todos os controladores de dispositivos devem ser implementados seguindo critérios de seu gerente. Por exemplo, um programador que deseje implementar um controlador de dispositivos de automação com ajuste de intensidade deve seguir uma interface definida pelo gerente de controladores de dispositivos de automação com ajuste de intensidade, e este por sua vez define que todo dispositivo deste tipo deve ter funções de ajuste de intensidade, além das funções definidas pelo seu próprio gerente, no caso o gerente de gerentes de dispositivos de automação. Este último havia já definido que um dispositivo de automação deve poder ligar e desligar.

A árvore é formada então seguindo uma hierarquia de funções, onde quanto mais baixo na árvore está se observa que mais funções os gerentes obrigam os dispositivos a terem, até que cheguem a uma folha, onde não há mais um gerente, e sim o próprio controlador do dispositivo.

Na raiz da árvore segue o gerente mais genérico, o 'Device Manager'. Além de definir o básico para o que podemos chamar de um dispositivo, esse também disponibiliza a API a ser usada pelo programador. Para seu funcionamento, o uso de serviços de auxílio (alguns já comentados no presente trabalho) são utilizados.

É importante observar que os serviços de auxílio não são algo presente somente no 'Device Manager'. Ao implementar algum gerente intermediário, o programador pode ver a necessidade de criar uma infra-estrutura de apoio também. Por exemplo, um gerente de

câmeras PTZ pode ver como necessário algum serviço que o auxilie a guardar em disco fotos e vídeos gravados de uma câmera qualquer. A idéia seria também disponibilizar esses serviços aos componentes filhos para que os serviços de auxílio sigam coerentemente a hierarquia de funções.

Os serviços para interfaces inteligentes são utilizados também pelo 'Device Manager' para que a interação do software que utilize a API seja a mais abrangente possível. Poderia-se ter como opção por exemplo que todos os serviços de log de eventos comuniquem ao usuário por sintetização de voz o que ocorre. Os serviços de interface podem ser enxergados como os demais serviços auxiliares, ou seja, podem ser criados em algum ponto mais abaixo da árvore. Um serviço para interface que está sendo estudado para o nível de câmeras é a execução de uma chamada a partir do reconhecimento de movimentos capturados por uma câmera. O destaque para esses serviços está sendo dado somente devido a sua importância, mas a nível de implementação elas não diferem muito dos serviços auxiliares.

Este trabalho então continuará na tentativa de implementar a estrutura apresentada usando o conjunto básico de tecnologias já aprendidas e eventuais outras que venham a aparecer.

4. CONCLUSÕES

A abordagem da estrutura final mostrada é importante principalmente no que diz respeito a facilidade de um controle físico da sala por artefatos de software, aumentando os cenários de automação possíveis. Seu caráter distribuído e extensível facilitaria a implementação de novos controladores de dispositivos e de suas instalações sem prejudicar o sistema em execução.

Apesar da dificuldade no desenvolvimento de implementações, pesquisas na área de computação ubíqua se mostram promissoras já que a quantidade de cenários possíveis aumenta a medida que novas tecnologias vão sendo dominadas.

Estudos finais em tecnologias necessárias estão em progresso e implementações iniciais do *AdminTool* sendo finalizadas.

REFERÊNCIAS

- IERUSALIMSKY, R. **Programming in Lua**. 2.ed. Rio de Janeiro: Lua.org, 2006. 308p.
- ROMÁN, M., HESS, C., CERQUEIRA, R. F. G., RANGANATHAN, A., CAMPBELL, R. H., Nahrstedt, K. Middleware Infrastructure for Active Spaces. **Pervasive Computing**, p. 74-83, out./dez. 2002.
- AUGUSTO, C. E. L., FONSECA, E. , MARQUES, L. , ROENICK, H., CERQUEIRA, R. F. G. , CORRÉA, S. L, **SCS - Sistema de Componentes de Software**
- CGI, <http://hoochoo.ncsa.uiuc.edu/cgi/>
- CGILua, <http://www.keplerproject.org/cgilua/>
- LuaSocket, <http://www.cs.princeton.edu/~diego/professional/luasocket/>
- LuaCOM, <http://www.tecgraf.puc-rio.br/~rcerq/luacom/>
- CORBA, <http://www.cs.wustl.edu/~schmidt/corba.html>
- SCS, <http://www.tecgraf.puc-rio.br/~scorrea/scs/>
- X10, <http://www.engr.udayton.edu/faculty/jloomis/ece445/topics/x-10/x10.html>