

MODELAGEM UML E DESENVOLVIMENTO DE UM SIMULADOR PARA CONTROLE DIFUSO DE GRUPO DE ELEVADORES

Alvaro A. Patiño-Forero, patino@unb.br¹
Vivian Milen Orejuela Ruiz, vorejuela@uceva.edu.co²
Guilherme Caribé de Carvalho, gccarval@unb.br¹
Carlos H. Llanos, llanos@unb.br¹

¹Universidade de Brasília, Departamento de Engenharia Mecânica, Campus Universitário Darcy Ribeiro, FT-ENM, Brasília, D.F., 70910-900, Brasil.

²Unidad Central del Valle, Facultad de Ingenieria de Sistemas, Carrera 27 A No. 48-144 Kilómetro 1 Salida Sur Tuluá-Colombia.

Resumo: Este trabalho apresenta o desenvolvimento da modelagem UML (Unified Modeling Language) e a implementação de um simulador de grupo de elevadores, incluindo técnicas de controle difuso aplicadas ao problema de controle de despacho dos carros componentes do grupo nas condições normalmente encontradas em prédios comerciais. O objetivo da modelagem UML foi explicitar as interrelações entre as diversas partes do modelo do simulador, de modo a tornar sua implementação em software mais facilitada do ponto de vista de organização. Em um edifício comercial, ao longo do dia, ocorrem mudanças no tipo de tráfego a ser atendido pelo grupo de elevadores. Neste trabalho, os diferentes tipos de tráfego foram identificados e classificados em três grupos: Up Peak (pico de demanda de chegada trabalhadores); Business Time (horário comercial); e Down Peak (pico de demanda de saída trabalhadores). Considerando esses grupos de demanda, foi desenvolvido um gerador de tráfego baseado em distribuições probabilísticas. O simulador é apresentado por meio de diagramas das partes mais importantes da modelagem. Os diagramas descrevem como o simulador foi desenvolvido, mostram a interação entre as diferentes entidades do sistema, tais como os usuários e o Controlador Lógico Programável (CLP). A comunicação de dados entre o aplicativo e o CLP, que foi substituído por um emulador, foi implementada com base no protocolo de comunicação OPC (OLE for Process Control) que permite uma comunicação bidirecional de dados, além de ser um padrão para troca de informações de controle no meio industrial. Desta forma consegue uma maior aproximação com o sistema real. O simulador resultante mostrou-se uma ferramenta importante para o desenvolvimento e implementação de novas estratégias de controle de grupos de elevadores, além de fornecer uma boa plataforma para futuros estudos de otimização nesse tipo de sistema.

Palavras-chave: UML, sistemas de controle, sistema de elevadores, lógica difusa, CLP.

1. INTRODUÇÃO

A Linguagem de Modelagem Unificada (UML) é uma linguagem gráfica padrão para a elaboração da estrutura de projetos complexos de software. A UML possibilita a visualização de todos os passos de desenvolvimento de um problema de modelagem na forma de diagramas padronizados, envolvendo elementos estruturais (classes, objetos, relacionamentos, etc.), de comportamento (*casos de uso*, iteração, máquinas de estado), de agrupamento (pacote, modelo, subsistema, etc.), de relacionamento (composição, agregação, generalização). A grande importância da modelagem é que permite avaliar a aderência e a qualidade da arquitetura, por parte do desenvolvedor, de modo a possibilitar-lhe detectar e corrigir falhas antes que estas possam comprometer o sucesso do projeto. A modelagem visual permite que os detalhes do processo sejam expostos ou escondidos conforme a necessidades, auxiliando o desenvolvimento de projetos complexos e extensos. Além disso, a UML ajuda a manter a consistência entre a especificação e a implementação por meio do desenvolvimento iterativo e do planejamento de testes em cada iteração.

Na área de sistemas de elevadores, por exemplo, quando os requerimentos no planejamento de um prédio são muito complexos, com vários elevadores a serem controlados de modo a minimizar o tempo de espera dos usuários e o consumo de energia, o desenvolvimento de algoritmos de controle envolveria o uso exaustivo do sistema real. Isso implicaria, em um primeiro momento, em problemas de atendimento da demanda de tráfego, uma vez que o desenvolvimento do controlador exigiria o conhecimento dos padrões de tráfego atuantes no sistema, além da necessidade de realizar muitos testes até se atingir um algoritmo otimizado. Diferentemente, o uso de simuladores de tráfego, baseados em padrões de tráfego já identificados e publicados na literatura (Peter. et al, 1998), aceleraria o desenvolvimento dos algoritmos de controle.

É amplamente conhecido que a simulação por computador é uma técnica particularmente adequada em situações em que o processo de estudo é difícil, perigoso e, adicionalmente, de alto custo. Um sistema elevadores cai dentro desta categoria (Barney et al, 2003). Isso tanto é verdade que empresas líderes do setor têm crescentemente desenvolvido e utilizado ferramentas de simulação para a realização de novos projetos de instalação e controle de elevadores (Cortéz, et al, 2006).

Nesse contexto, este trabalho apresenta a modelagem de um simulador para um sistema de controle difuso de grupo de elevadores (FEGCS) usando a metodologia de processo unificado de desenvolvimento de software proposta pela Linguagem Unificada de Modelagem (UML). O desenvolvimento da modelagem proposta foi feito de modo a se possibilitar sua realização em uma arquitetura física de um sistema de elevadores moderno, usando instrumentação industrial baseada em uma rede DeviceNet® e sensores com interfaces apropriadas para este tipo de rede (Patiño A. *et al.*, 2009; Xinyun Y. *et al.*, 2009). O uso da UML neste trabalho possibilitou a descrição do sistema em um grau de abstração bem perto da realidade.

Este trabalho está organizado em seções, cujos conteúdos são explicitados a seguir. A seção 2 descreve brevemente o sistema de controle de grupo de elevadores e os sistemas modernos utilizados para este quesito. Adicionalmente, a mesma seção apresenta como se faz a geração dos padrões de tráfego do sistema e uma descrição do algoritmo de controle utilizado. A seção 3 apresenta um estudo comparativo do estado da arte em relação à técnica proposta neste trabalho. A seção 4 descreve a modelagem UML do simulador e o desenvolvimento do sistema de simulação proposto. Finalmente, a seção cinco mostra as conclusões do trabalho.

2. COMPONENTES DO SISTEMA

2.1. Sistema de Elevadores

O Sistema de Controle de Grupo de Elevadores (EGCS) é um sistema que consegue dirigir sistematicamente três ou mais elevadores com o objetivo de se otimizar alguma função de custo. No caso em estudo, buscou-se diminuir o tempo de espera dos passageiros e o consumo de energia dos elevadores.

O EGCS é considerado, computacionalmente, como um problema NP-hard (Nikovsky e Brand, 2003) e envolve processos dinâmicos estocásticos, considerando incertezas sobre os instantes de tempo em que ocorrem novas chamadas de pavimento e sobre o número de passageiros esperando nos andares onde se originaram as chamadas (Beielstein *et al.*, 2003). São, conseqüentemente, sistemas de dinâmica complicada que incluem não linearidades e problemas de controle multi-objetivo (Markon, et al, 2006).

Atualmente, existem muitas técnicas avançadas para controle de grupo de elevadores que podem ser classificadas no campo da inteligência artificial (IA). Por exemplo, no caso de grupos de elevadores que usam o sistema tradicional de botões de chamada para acima e para baixo, várias técnicas têm sido publicadas na literatura especializada, dentre as quais citam-se: controle ótimo (Closs. et al, 1970), lógica difusa (Ho and Robertson, 1994), programação dinâmica (Chan and So, 1996), algoritmos genéticos (Miravete. et al, 1999.), sistemas especialistas (Qun et al., 2001) e redes neurais (Barney and Imrak, 2001).

Sistemas de elevadores recentes têm uma estrutura diferente daquelas normalmente utilizadas nos sistemas tradicionais. Uma dessas diferenças é a possibilidade de se conhecer o andar de destino do passageiro antes que ele entre no carro, como nos chamados “Sistemas de Controle de Destino” (SCD) (Markon, 2008). O desempenho de sistema de elevadores que usam o sistema SCD é muito melhor porque a estratégia de controle pode ser desenvolvida usando informação prévia sobre o andar de destino, assim como o número de passageiros esperando pelo carro no pavimento, supondo que todos os passageiros tenham informado seus respectivos destinos. Neste caso, o SCD fornece uma boa aproximação da demanda para o sistema de controle agrupar os passageiros que irão seguir para um mesmo andar, diminuindo o tempo de espera e o número de paradas enquanto chega ao destino (Sorsa. et al, 2005).

2.2. Arquitetura do Sistema de Elevadores

A Figura 1 apresenta um sistema moderno de controle de grupo de elevadores usando o sistema de controle de destino (SCD), o qual usa uma interface homem-máquina (IHM) (por exemplo um PanelView®) em cada andar e um controlador lógico programável (CLP) que implementa um sistema de controle local (LCS) para cada elevador (Patiño A., 2009). O LCS recebe como entradas os comandos enviados pelo FEGCS e controla a velocidade para cada elevador, realizando também outras tarefas tais como: verificação da presença de obstáculos antes de fechar as portas, inspeção do peso da cabina do elevador, controle de abertura e fechamento das portas, etc. O LCS é conectado ao FEGCS por meio do protocolo de comunicação OPC (*Object Linking and Embedding for Process Control*) (OPC Foundation, 1996). O OPC permite o intercâmbio de informação em tempo real entre aplicações (*software*) e processos (*hardware*). O sistema de elevadores também possui uma rede industrial de campo (DeviceNet®), para conectar vários instrumentos industriais como: Drives AC (um para cada carro), sensores indutivos, módulos de entrada e saída digital e IHM.

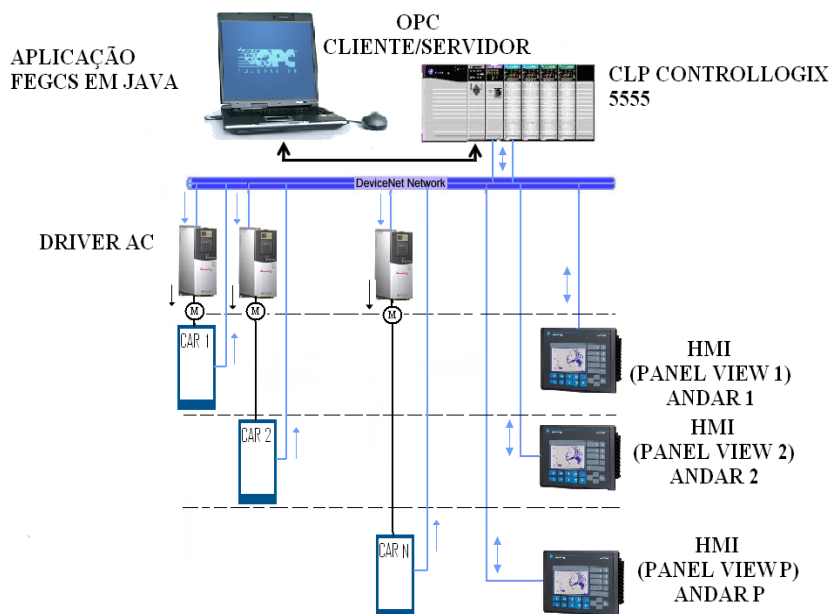


Figura 1 Arquitetura do Sistema de Elevadores.

2.3. Gerador de Tráfego de Chamadas.

O fluxo de tráfego de passageiros é fundamental no EGCS. Modelar este fluxo é muito importante para realizar o simulador de tráfego de elevadores. Apresentar a forma como chegam as pessoas ao sistema de elevadores é de extrema importância para o correto funcionamento do EGCS. No caso em estudo, o componente que faz essa tarefa é o gerador de tráfego de passageiros, o qual gera as chamadas de andar de origem e as chamadas de destino, de acordo com o tráfego presente. Este tipo de processo é estocástico. A taxa de chegada dos passageiros segue uma distribuição Poisson (Dos Santos, 1972) e sua distribuição de probabilidade é mostrada na equação (1) (Zhifeng et al, 2007):

$$P = \frac{(\lambda T)^n e^{-\lambda T}}{n!}, \quad n = 0, 1, 2, \dots \quad (1)$$

Na eq. (1), a probabilidade de chegada, P , é definida em um tempo T , com taxa de chegada de passageiros, λ e n é o número de eventos (chegadas de passageiros). O intervalo de tempo transcorrido entre as chegadas de passageiros segue uma distribuição exponencial (Hummet et al, 1978). A equação (2) mostra como calcular os tempos em que se esperam chegadas de passageiros com base em intervalos de tempo distribuídos segundo uma função de densidade de probabilidade exponencial.

$$t_0 = 0 \quad t_i = t_{i-1} - \frac{\ln(r)}{\lambda}, \lambda > 0 \quad (2)$$

Na eq. (2), t_0 é o tempo inicial, r é um número aleatório com distribuição uniforme entre 0 e 1 e λ é a taxa temporal de chegada de passageiros, calculada em termos de um percentual da população total, em um período de 5 minutos. O valor de λ é calculado conforme mostra a eq. (3) (Hummet et al, 1978).

$$\lambda = \frac{0.01(TI)}{300} \sum_i POP_i \quad (3)$$

Na eq. (3), TI é a intensidade de tráfego fornecida, POP_i é a população do andar i e as unidades de λ são em passageiros por segundo.

Depois de calcular o tempo de chegada do passageiro seguinte, a próxima tarefa é estabelecer o andar de origem e o andar de destino da chamada.

2.4. Algoritmo de Controle

Um sistema de controle de grupo de elevadores (EGCS) deve considerar muitos fatores que afetam o estado atual e o estado futuro do sistema. Dentre esses fatores, observa-se a facilidade de se monitorar a posição atual de cada

elevador e de se armazenarem em uma pilha as chamadas de pavimento e os respectivos destinos. Entretanto, é difícil conhecer informação precisa sobre o número de passageiros esperando nos pavimentos onde aconteceram as chamadas, tendo em vista que às vezes pessoas andam em grupo e somente um componente do grupo chama o elevador. Além disto, informações futuras sobre chamadas de pavimento e cabina são incertas (Chenais e Weinberger, 1992, Kim *et al.*, 1998). Geralmente, em edifícios comerciais, as mesmas características de tráfego se apresentam em horários similares e o mesmo número de vezes por dia. Portanto, o tempo é um fator de importância na classificação do tráfego, porém, a maior peça de informação é o número de chamadas de subida, de descida e de cabina (Siikonen, 1997, Kim *et al.*, 1998). A Figura 2 mostra um padrão de tráfego de subida e descida típico de um edifício comercial. O número de chamadas de subida representa o tráfego de subida (TS) e o número de chamadas de descida representa o tráfego de descida (TD). A partir destes gráficos, característicos em cada edifício, é possível determinar as estratégias que o controle de grupo deve efetuar.

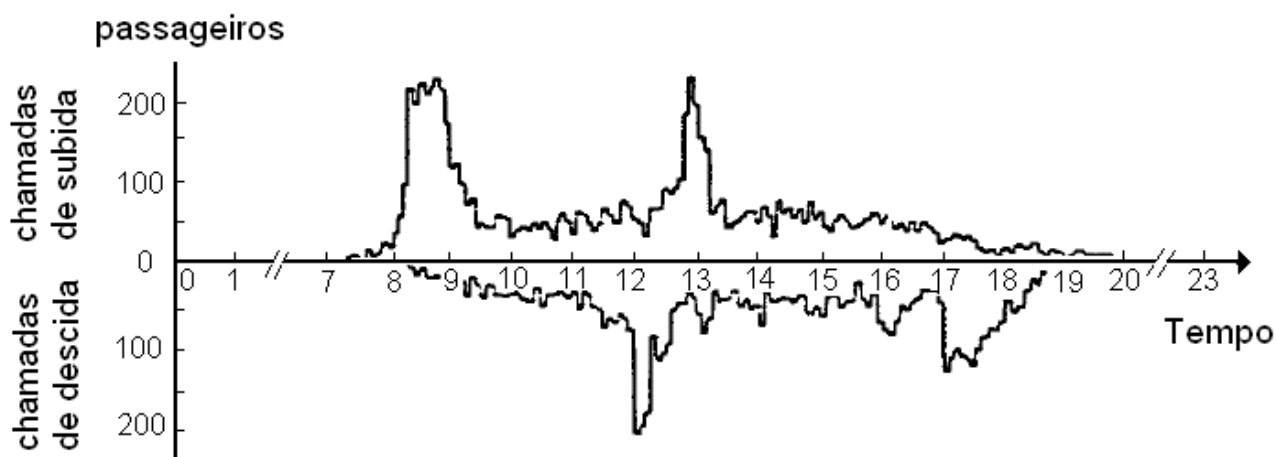


Figura 2: Tráfego de subida e descida em edifícios comerciais (Tsuji e Amano, 1989).

Algoritmos de controle convencionais “*Nearest Car*” são usados principalmente em prédios de escritório de sete andares devido à facilidade de implementação (Barney et al, 2003). Esses tipos de algoritmos normalmente seguem o princípio de proximidade, ou seja, o elevador que se encontra mais perto e que se movimenta no mesmo sentido atenderá à chamada de pavimento. Observe-se que, este tipo de algoritmo tem um único critério de desempenho: o tempo que o passageiro espera para ser atendido, que se procura diminuir.

O algoritmo de controle de grupo elevadores realizado neste trabalho, por outro lado, é baseado em lógica difusa e foi desenvolvido com o intuito de melhorar o desempenho de um algoritmo de controle convencional, que apresenta debilidades em relação aos controladores implementados nos sistemas mais modernos. O algoritmo convencional não utiliza outros tipos de critério de desempenho que ajudariam a melhorar a eficiência do sistema, tais como consumo de energia e tempo de viagem. Diferentemente, o uso da lógica difusa possibilita a inclusão de vários objetivos que serão avaliados simultaneamente como regras de decisão, de maneira a proporcionar um valor de prioridade a cada elevador buscando determinar aquele que se apresente como o mais conveniente para atender as chamadas de pavimento.

O sistema de controle desenvolvido foi projetado com os objetivos de diminuir o tempo de espera dos passageiros e o consumo de energia. De modo a possibilitar o teste de desempenho do controlador proposto nas diversas condições de tráfego normalmente encontradas em edifícios comerciais ao longo do dia, decidiu-se por classificá-las em três grupos: Up Peak (pico de demanda de chegada de trabalhadores); Business Time (horário comercial); e Down Peak (demanda de saída de trabalhadores). As funções de pertinência e as regras de inferência do controlador difuso foram desenvolvidas com base nessa classificação. Para se gerarem dados para a simulação, implementou-se um algoritmo para geração probabilística das chamadas de pavimento seguindo as equações (2) e (3). Na Figura 3 apresentam-se três funções de pertinência que são as entradas do sistema difuso utilizado. Essas são explicadas abaixo:

- 1) **Tempo_de_espera**: é o tempo que passageiro espera no andar até que o elevador chegue onde ocorreu a chamada. O tempo de espera é calculado pela equação 4:
- 2) **Disponibilidade atual de carga do elevador**: é o número atual de vagas disponíveis no elevador, no momento em que o elevador chega ao andar onde a chamada de pavimento ocorreu.
- 3) **Distância a ser percorrida pelo elevador**: é o número de andares que o elevador deve percorrer desde a posição atual até o andar onde a nova chamada de pavimento ocorreu.

$$\text{Tempo de Espera} = (\text{Numero de paradas}) * \text{Tempo parada} + (\text{Distância percorrida}) * \text{Tempo Vôo.} \quad (4)$$

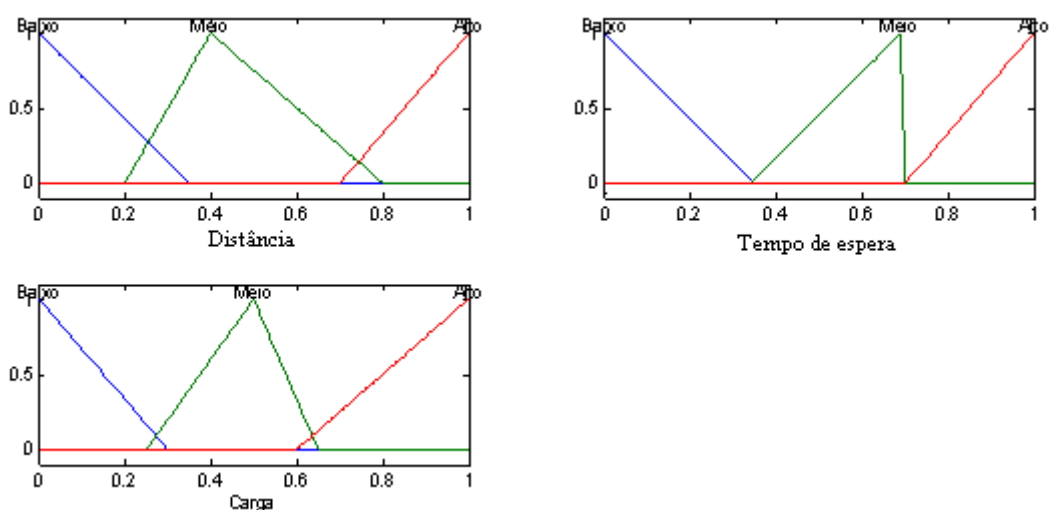


Figura 3: Funções de pertinência

Na Figura 4, apresentam-se as regras de inferência propostas neste trabalho, as quais tem o objetivo de avaliar a conveniência de cada elevador para atender às chamadas de pavimento realizadas em cada andar.

```

1. If (Distância is Alto) and (Tempo_de_espera is meio) then (Prioridade is baixa) (1)
2. If (Distância is Baixo) and (Tempo_de_espera is alto) then (Prioridade is meia) (1)
3. If (Distância is Alto) and (Tempo_de_espera is alto) then (Prioridade is baixa) (1)
4. If (Distância is Alto) and (Tempo_de_espera is baixo) then (Prioridade is meia) (1)
5. If (Distância is Baixo) and (Tempo_de_espera is baixo) then (Prioridade is alta) (1)
6. If (Distância is Baixo) and (Tempo_de_espera is meio) then (Prioridade is meia) (1)
7. If (Tempo_de_espera is baixo) and (Carga is alto) then (Prioridade is alta) (1)
8. If (Distância is Baixo) and (Carga is baixo) then (Prioridade is baixa) (1)
9. If (Tempo_de_espera is meio) and (Carga is baixo) then (Prioridade is baixa) (1)
10. If (Distância is Meio) and (Tempo_de_espera is meio) then (Prioridade is meia) (1)
11. If (Distância is Meio) and (Tempo_de_espera is alto) then (Prioridade is baixa) (1)

```

Figura 4: Base de regras

3. ESTADO DA TÉCNICA.

Atualmente, existem poucas referências na bibliografia que envolvam simulação de transporte vertical. Dentre essa referências podem-se citar: os trabalhos de Siikonen (1993 e 2001) e o trabalho de R.D. Peters (1998), fundador de Peters Research Ltd, que comercializa o simulador Elevate™. Além deste encontraram-se referências a outros simuladores comercialmente disponíveis no mercado: o simulador OtisPlan® da Otis Elevator Co e o sistema Building Traffic Simulator (BTS) da tradicional KONE corporation. Observe-se que o desenvolvimento de simuladores tende a ser feito por empresas privadas de transporte vertical. Estas empresas desenvolvem seus próprios simuladores, os quais por razões comerciais são protegidos e seus códigos não ficam abertos. Por conseguinte, o desenvolvimento deste tipo de ferramenta não tem um modelo único de elaboração. Cada empresa modela seu próprio simulador de tráfego para sua implantação final. A modelagem apresentada neste trabalho aspira deixar uma arquitetura de desenvolvimento aberta de um sistema moderno de elevadores através da linguagem de desenvolvimento de software UML. Deste modo, com a modelagem UML é possível que outros pesquisadores possam usar este tipo de modelo para suas pesquisas nesta área.

4. MODELAGEM UML DO SIMULADOR

Modelagem do simulador de controle difuso de tráfego de elevadores é apresentada a Figura 2 através dos diagramas de *casos de uso*¹. O diagrama de *casos de uso* propicia uma visão geral do sistema (Rumbaugh et al, 1999).

¹ Na engenharia de software, o termo “caso de uso” é um tipo de classificador representando uma unidade funcional coerente provida pelo sistema, subsistema, ou classe manifestada por seqüência de mensagens entre os sistemas e um ou mais atores.

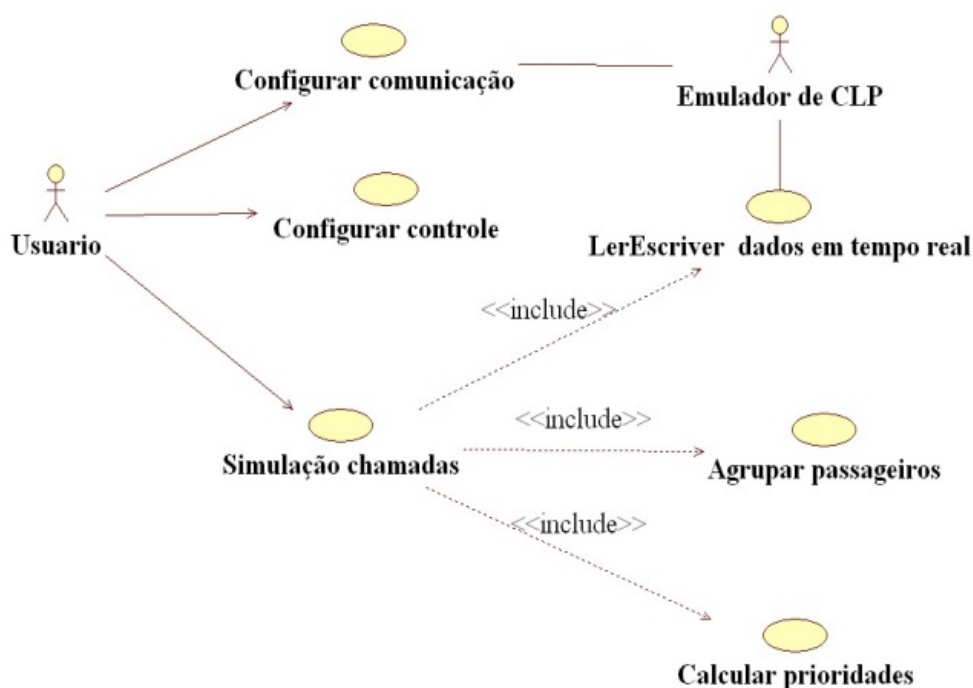


Figura 5. Casos de uso do sistema

A Figura 3 apresenta um ator usuário que representa a pessoa que inicia a configuração do sistema. “**Configurar comunicação**” é o primeiro *caso de uso* do sistema onde o usuário através do servidor OPC escolhe os parâmetros necessários para o funcionamento do sistema (ver Figura 4). O *caso de uso* “**Configurar Controle**” é onde se configura: (a) o número de andares, (b) número de elevadores e (c) capacidade do sistema. No *caso de uso* “**Simular Chamadas**” o usuário inicia o controle grupal de elevadores e a sua simulação. O *caso de uso* “**Ler-escrever dados em tempo real**” o sistema lê e escreve dados no emulador de CLP por meio do protocolo OPC. No *caso de uso* “**Agrupar Passageiros**”, o sistema agrupa os passageiros com um mesmo destino. No *caso de uso* “**Calcular Prioridade**” o sistema, por meio da lógica difusa, calcula qual elevador é o mais conveniente para atender a chamada de pavimento. A Figura 5 mostra a interface gráfica de simulação onde se apresentam os tempos de simulação, cálculos do controle difuso de grupo de elevadores, tempos de espera, tempo viagem, números de paradas de cada elevador.

Quando o usuário configura a comunicação OPC tem a opção de testar cada variável do simulador de CLP programada, para conferir seu correto funcionamento. Depois pode escolher as variáveis (endereços de memória de cada variável) a trabalhar dentro do simulador em uma lista para depois salvar estas por meio de uma janela em um arquivo texto com se mostra na figura 4.

A interface gráfica de simulação permite que o usuário escolha o arquivo de configuração, selecionando o botão “*configurar leitura*”. O arquivo de configuração é aquele gerado por meio da interface de configuração da comunicação e nele são armazenados os endereços de memória das variáveis do CLP. O botão “start” permite colocar os elevadores em uma posição inicial. Por outro lado o botão “*simulação chamadas*” permite escolher o arquivo gerado pelo sistema gerador de tráfego. O sistema funciona quando o tempo do relógio do computador é igual ao tempo gerado pelo gerador de tráfego. Além disso, o sistema apresenta os cálculos de prioridades proporcionados pelo sistema de controle difuso, como também a divisão de zonas de acordo com o tráfego presente, de tal modo que o sistema de controle de grupo escolha o elevador mais conveniente. Por outro lado, na interface gráfica de simulação colocam-se os números de andares, número de elevadores e população do prédio que faz parte da configuração do sistema.

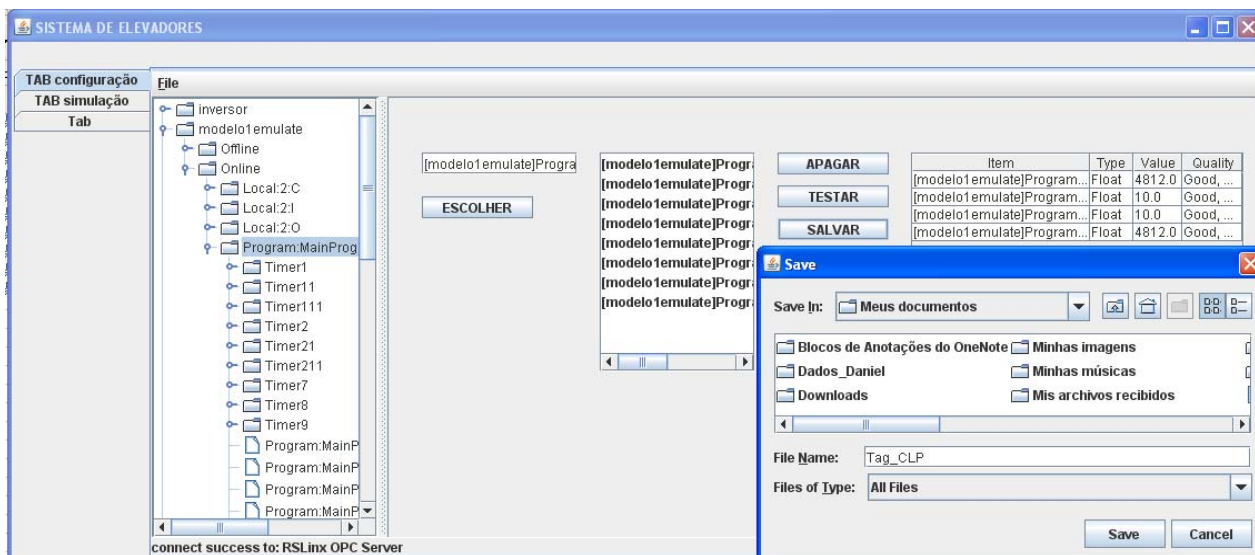


Figura 6 Interface gráfica de configuração da comunicação



Figura 7 Interface gráfica de simulação

Cada *caso de uso* tem um diagrama de seqüência por meio do qual se faz uma descrição detalhada dos processos necessários à execução de cada tarefa. Por exemplo, o *caso de uso* “**Simular Chamadas**” tem seu diagrama de seqüência conforme mostrado na Figura 6. O usuário inicia o sistema para iniciar a simulação. O software executa o objeto *interfacecontrole* para apresentar a interface gráfica para que o usuário possa iniciar a configuração do sistema. A primeira coisa que usuário tem que fazer é colocar o andar origem e andar destino de cada elevador (valores iniciais). Logo, o usuário deve selecionar o botão *Configuração leitura* onde por meio de uma janela se escolhe o arquivo onde estão os endereços das variáveis do emulador para trabalhar dentro do simulador. O usuário depois disso, deve selecionar o botão *start*. Desta maneira, o simulador por meio do objeto *OPCLeitura* envia os valores iniciais da posição atual dos elevadores para o servidor OPC, a mesmo tempo o servidor lê os dados que apresentam mudanças dentro do emulador, de forma concorrente, por meio do método *OnvalueChanged()* da biblioteca OPC. Quando os elevadores chegam às posições iniciais, o usuário seleciona o botão *Simulação chamadas*, o qual dispara a apresentação de uma janela, por meio da qual escolhe-se o arquivo onde se mostram os andares origem, andares destino e tempos (este arquivo é criado pelo gerador de tráfego de elevadores). Assim por meio do objeto *Leituradados* o sistema lê os dados de tempo da simulação e os compara com o relógio do computador, de modo que, a cada instante de tempo em que a comparação de tempo é verdadeira, executa-se o sistema de controle difuso para escolher o elevador mais conveniente para cada chamada.

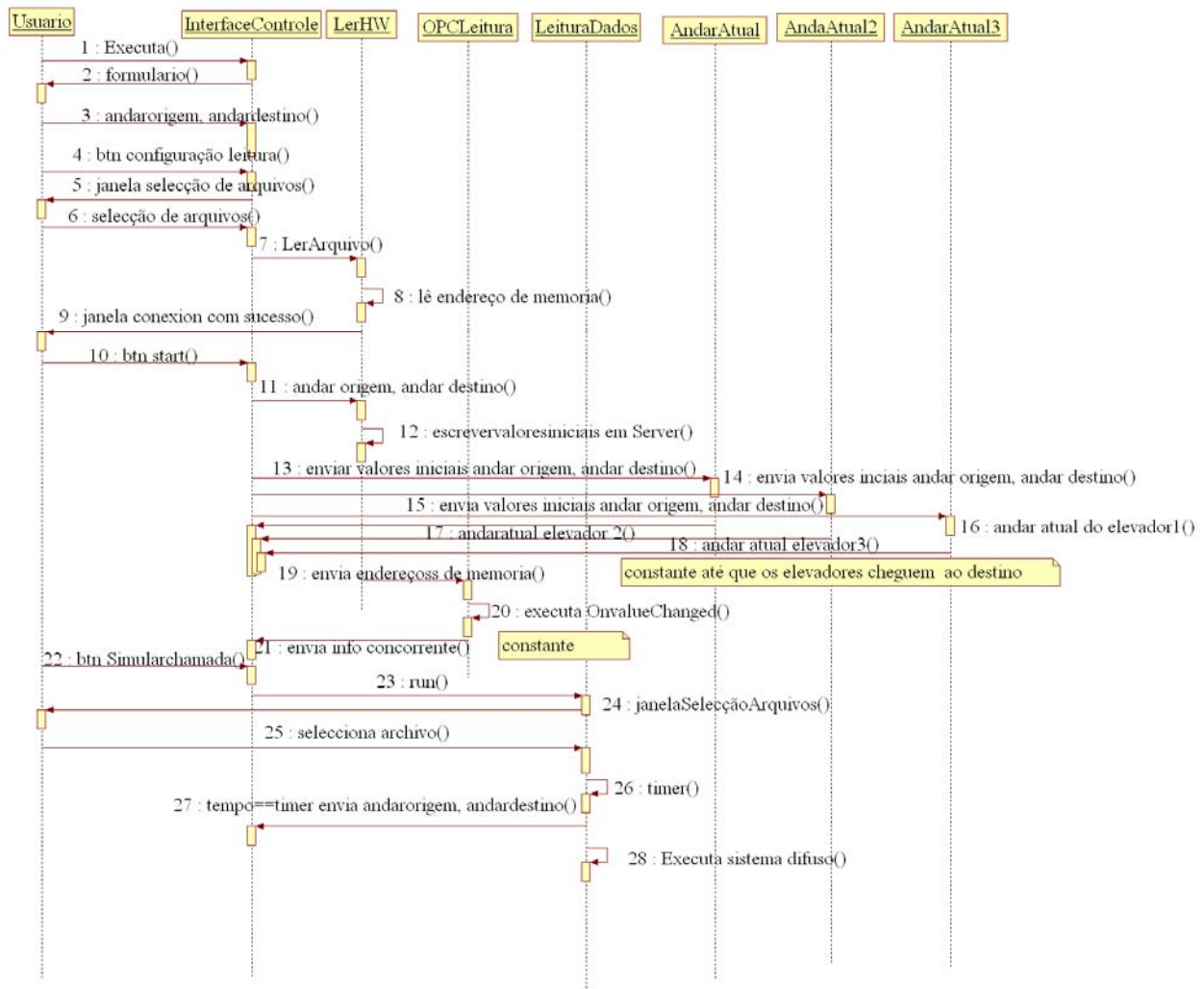


Figura 8 Diagrama de seqüência simular chamadas

Cada *caso de uso* também tem um diagrama de classes onde se apresenta a estrutura das relações entre estas. Por exemplo, para o *caso de uso* “**Configurar configuração**” (ver Figura 7), apresentam-se 6 classes. A classe *interfacecontrole* que é onde está a parte visual e onde se criam os objetos para interagir com as outras classes, como o caso da classe *ControladorComunicação*. Esta tem vários objetos criados por meio da biblioteca OPC para poder usar os métodos da biblioteca e desta maneira configurar a comunicação. A classe *controlador comunicação* executa a classe *Showconnectdialog*, a qual apresenta uma janela onde mostra os diferentes servidores OPC instalados dentro do computador. A classe *OpcConnection* realiza a conexão do sistema com o servidor escolhido. As classes *DinamicIOtemView* e *AbstractTableModel* apresentam os valores das variáveis selecionadas do emulador dentro de uma tabela, de modo que se atualizam os valores de forma dinâmica dentro da tabela a medida que os valores do emulador variam.

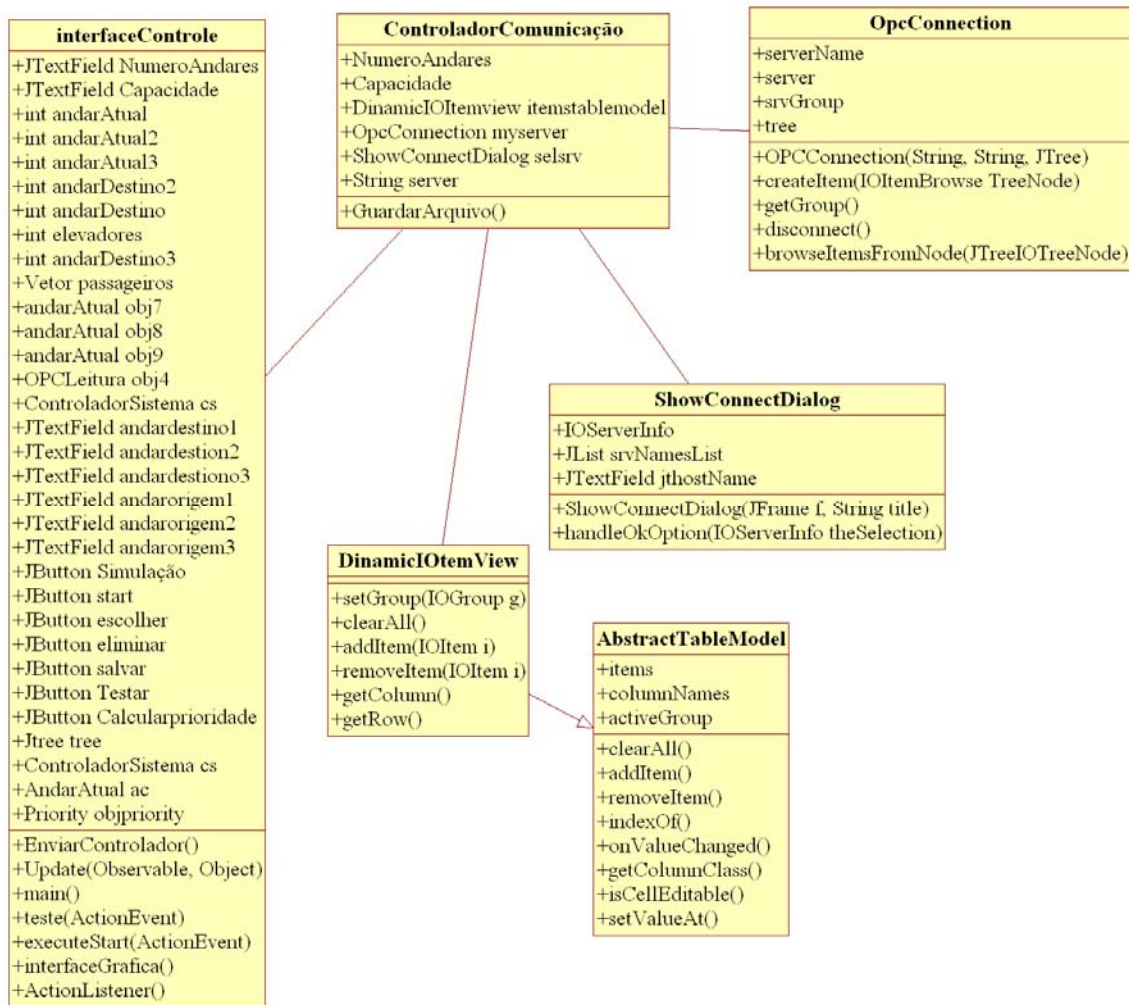


Figura 9 Diagrama de classes de configurar comunicação

O diagrama de componentes (ver figura 8) apresenta de forma geral o sistema como um todo. O cliente é o simulador para o controle de grupo desenvolvido em Java, o qual se comunica com o servidor por meio do protocolo OPC. O servidor OPC está em um *software* que funciona como uma interface de comunicação, cujo objetivo é possibilitar a comunicação entre diferentes aplicativos com instrumentos físicos. O servidor se encontra nos programas de comunicação de cada fabricante de CLP (para o desenvolvimento deste trabalho utilizou-se o software RSLinx® da Rockwell Automation). O Emulador de CLP é uma ferramenta de *software* que substitui o CLP, em que é possível programar como se estivesse utilizando um controlador real. No caso deste trabalho, utilizou-se o emulador RSLogix Emulate 5000®, da empresa Rockwell Automation.

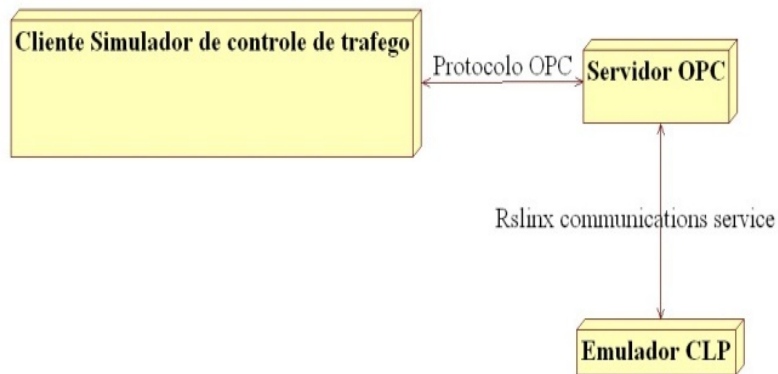


Figura 10 Diagrama de componentes

5. CONCLUSÕES

O simulador de controle de trafego proposto foi projetado para se ter um ambiente controlado que represente as características mais importantes do ambiente real de uma arquitetura moderna de controle de grupo de elevadores. A modelagem UML permite que a ferramenta possa facilmente ser adaptada a diferentes ambientes caso necessário, de modo a possibilitar estudos mais aprofundados nessa área. O trabalho demonstrou de forma didática a potencialidade de utilização da UML na especificação de sistemas de tempo real a partir da perspectiva da Orientação a Objetos. O simulador desenvolvido possibilita planejar um sistema de transporte vertical, determinando o número de carros a serem utilizados, a arquitetura física da instalação, os algoritmos de controle de grupo e de otimização, além de outras opções, de acordo com as características específicas de cada prédio. O sistema de controle de elevadores proposto explora o uso técnicas da Inteligência Artificial como a lógica difusa como uma alternativa para o controle de um grupo, objetivando identificar padrões de tráfego e adotar estratégias de despacho de elevadores usados nos sistemas mais modernos.

6. AGRADECIMENTOS

Os autores gostariam de agradecer à empresa Rockwell Automation do Brasil Ltda, cujo contínuo apoio, tanto material e financeiro, possibilitou a execução do projeto em que se insere o assunto escrito neste artigo.

7. REFERÊNCIAS

- Barney, G.C. and Imrak, E. (2001): applications of neural networks to lift traffic control, Elevator World, May. -11.4.4
- Barney G.C, (2003)“Elevator Traffic Handbook: Theory And Practice”, Spon Press, London and New York.
- Beielstein Thomas, Markon Sandor, Mike Preuss, MIC (2003) “A Parallel Approach to Elevator Optimization Based on Soft Computing” in The Fifth Metaheuristics International Conference.
- Closs, G.D, (1972): The computer control of passenger traffic in large lift system, Phd thesis, UMIST.-10.2.3.
- Chan, W.L. and So, A.T.P. (1996): Dynamic zoning for intelligent supervisory control, Int J. Elev. Eng., 1.-11.4.4
- Cortéz P., Delgado M., Ibañez N., Muñozuri J. (2006), “Herramienta de planificación y simulación de sistemas de transporte vertical”, X congreso de Ingeniería de organización, Valencia.
- Ho, M. and Robertson, B. (1994): Elevator group supervisory control using fuzzy logic, Canadian Conference on Elevator and Computer Engineering, 2.-11.4.4.
- Hummet George T., Moser. D., Powell A. 1978, “Real time simulation of elevators”, IEEE computer Society Press, Los Alamitos, CA, USA.
- Lin, S.L. and Van Ness J.E (1994). Parallel Solution of Sparse Algebraic Equations. IEEE Transactions on Power Systems, Vol.9, No. 2, pp. 743- 799.
- Markon Sandor, Aoki ichi Ken, Masami Nakagawa, and Takeshi Sudo, (2008) “Recent Trends Group Control System”, Conference System computers and communication.
- Markon Sandor, Kita Hajime, Hiroshi Kise, Beielstein, (2006) “ Control of Traffic Systems in Buildings”, Advances in Industrial Control, Springer, pages. 69-78.
- Miravete, A. (1999): Genetics and intense vertical traffic, Elevator World, July- 11.4.4
- Nikovsky, D., Brand, M. (2003). “*Decision-Theoretic Group Elevator Scheduling.*” In: AAAI Proc. 13th Int. Conference on Automated Planning and Scheduling, pp. 133-142, Trento, Italy.
- OPC Foundation, (1996) “Dedicated to interoperability in automation”, 21 August 2009 < <http://www.opcfoundation.org> >
- Patiño Forero Alvaro A., (2009). “Modeling of elevator group control system using programmable logic control and destination control system”. 20th International congress of Mechanical Engineering. November 15-20 Brazil.
- Peters, R.D. (1998). Simulation for control system design and traffic analysis, Elevator Technology 9, proceedings of ELEVCON`98.
- Qun, Z. Ding, S., Yu, C. and Xiaofeng, L. (2001): Elevator group control system modelling based on object orientated Petri Net, Elevator world, August.-11.4.4
- Rumbaugh J., G. Booch, I. Jacobson, (1999) "The Unified Modeling Language Reference Manual", Addison Wesley.
- Siikonen, M-L. (1993). Elevator traffic simulation. Simulation, Vol. 61, No 4, pp. 257-267.
- Sorsa J., Hakonen and M. L. Siikonen, (2005) “Elevators selection with Destination Control System”. In Elevators Technology.
- Tsuji, S., Amano, M. (1989). “*Application of the Expert System to the Elevator Group- Supervisory Control.*” In: IEEE Proceedings of the 5th Conference on Artificial Intelligence Applications, pp. 287-294, Miami, USA.
- Xinyun Yu, Xiaoyun Feng, Chenglin Xiong, (2009), “The Design and Implementation of Elevator Group Control System Research Platform” International Workshop on information Security and Application (IWISA 2009).
- Zhifeng Pan, Fei Luo, Yuge Xu. (2007): “Elevator Traffic flow model based on dynamic passenger distribution ” IEEE International Conference Automation Guangzhou, CHINA - May 30.

DIREITOS AUTORAIS

Os autores são os únicos responsáveis pelo conteúdo do material impresso incluído no seu trabalho.



VI CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA
VI NATIONAL CONGRESS OF MECHANICAL ENGINEERING
18 a 21 de agosto de 2010 – Campina Grande – Paraíba - Brasil
August 18 – 21, 2010 – Campina Grande – Paraíba – Brazil

UML MODELLING AND SOFTWARE IMPLEMENTATION OF A SIMULATOR FOR ELEVATOR GROUP SYSTEM USING FUZZY CONTROL

Alvaro A. Patiño-Forero, patino@unb.br¹

Vivian Milen Orejuela Ruiz, vorejuela@uceva.edu.co²

Guilherme Caribé de Carvalho, gccarval@unb.br¹

Carlos H. Llanos, llanos@unb.br¹

¹Universidade de Brasília, Departamento de Engenharia Mecânica, Campus Universitário Darcy Ribeiro, FT-ENM, Brasília, D.F., 70910-900, Brasil.

²Unidad Central del Valle, Facultad de Ingeniería de Sistemas, Carrera 27 A No. 48-144 Kilómetro 1 Salida Sur Tuluá-Colombia.

Abstract. *This work presents the UML (Unified Modeling Language) modeling and the implementation of an elevator group simulator. The simulator uses fuzzy control techniques for dealing with the problem of controlling the dispatching of elevator cars as a result of hall call conditions typically encountered in commercial buildings. The objective of the UML model was to show the relationship between the various parts necessary to build the simulator, such that its software implementation could be made easier, in an organizational point of view. In a commercial building, the pattern of elevator traffic changes according to the moment during the working hours. In this work, three different patterns were identified and classified: Up Peak traffic; Business Time traffic and Down Peak traffic. Considering these groups of demand, a traffic generator based on probability distribution was developed. The simulator is represented through diagrams the most important parts of the modeling. The diagrams describe how the simulator was developed; show the interaction between different system entities such as users and Programmable Logical Controller (PLC). The data communication between the application and PLC, which was replaced by an emulator, was implemented based on the communication protocol OPC (OLE is Process Control) which allows a bi-directionally communicate of data, besides being a standard for exchanging control information in the industrial. The resulting simulator proved to be an important tool for the development and implementation of new control strategies for groups of elevators, moreover to providing a good platform for future optimization studies in this type of system.*

Keywords: *UML, system control, system elevators, fuzzy logic, PLC.*