

SIMULADOR DE EMPRESAS LÍDER: protótipo de um sistema baseado em agentes



Orientando: Marcos Virgilio da Costa

Orientador: Mauricio Capobianco Lopes



Roteiro

- Introdução
- Objetivos
- O Simulador de Empresas Líder
- Agentes & Sociedades de Agentes
- Desenvolvimento do protótipo
- Conclusões e sugestões
- Bibliografia básica



Introdução

- O simulador de empresas Líder, é um sistema direcionado para treinamento em gestão de recursos humanos.
- Aplicado em disciplinas de graduação e pós-graduação de diversas universidades de Santa Catarina .
- Software (sistema computacional).



Introdução

- O Líder idealiza a representação do modelo comportamental humano mais próximo da realidade.
- As Sociedades de Agentes são um exemplo claro da aplicação deste modelo que tem como foco o comportamento inteligente em um contexto social.



Introdução

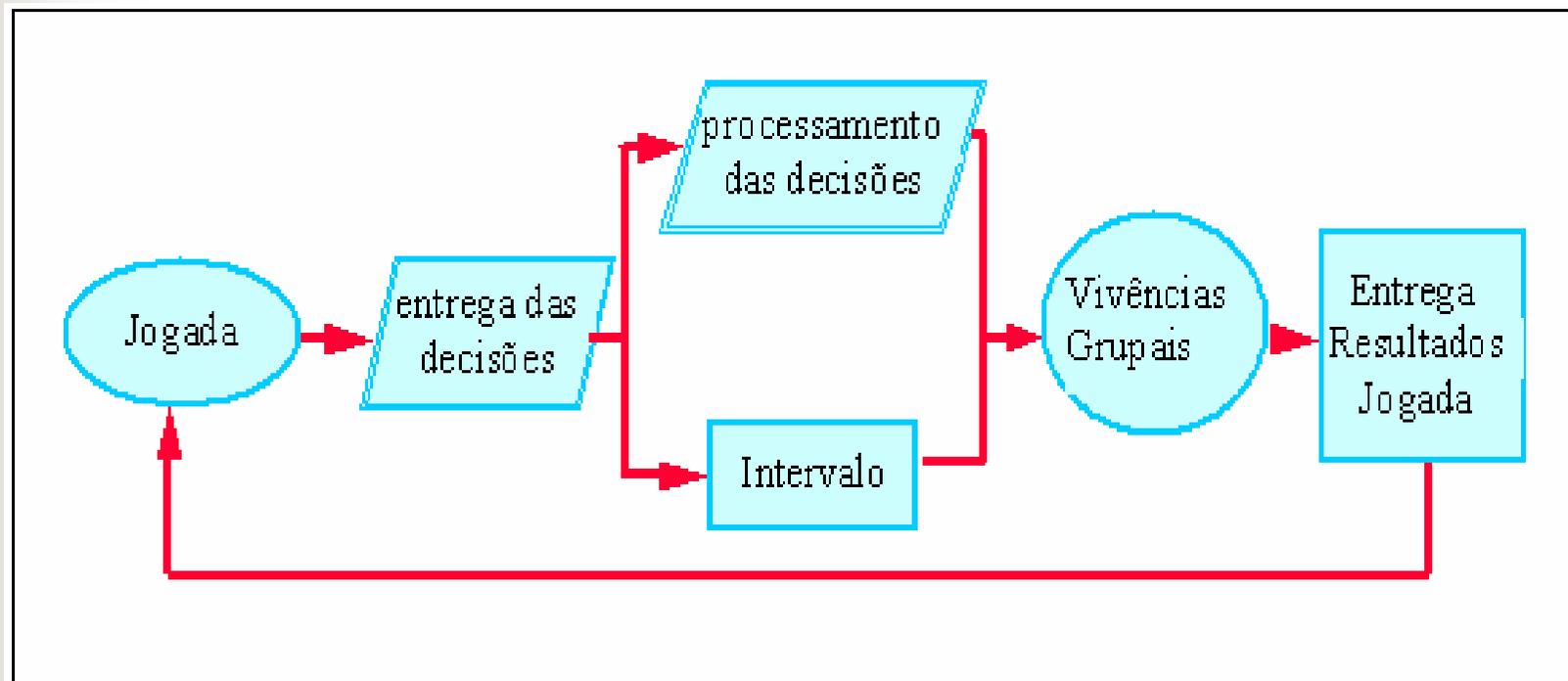
- Este trabalho visa representar, através de Agentes Inteligentes, o contexto social interativo do simulador de empresas Líder de uma forma mais próxima dos modelos humanos.
- Especificação em UML e implementação em Object Pascal (Delphi).



Objetivos

- Rever a modelagem do simulador para uma eventual adequação do mesmo ao contexto de agentes, sem divergir do modelo matemático original do Líder.
- Propor uma nova forma de comunicação entre participantes e agentes e entre os próprios agentes.

Simulador de Empresas Líder





Simulador de Empresas Líder

- Sistema atual implementado em Delphi, linguagem Object Pascal e Banco de Dados Paradox.
- Scripts SQL.
- Os estados dos colaboradores e decisões ficam armazenados no banco de dados.
- Constante atualização do banco de dados após cada processamento.



Simulador de Empresas Líder

- Deficiência quanto à interatividade com os usuários do sistema.
- Relação muito direta entre decisão e processamento.



Agentes

- “Uma entidade real ou virtual que emerge num ambiente onde pode agir intencionalmente, ser hábil a perceber e representar o ambiente, e a comunicar-se com outros agentes, além de ser autônomo”
Jomi Fred Hübner.



Agentes

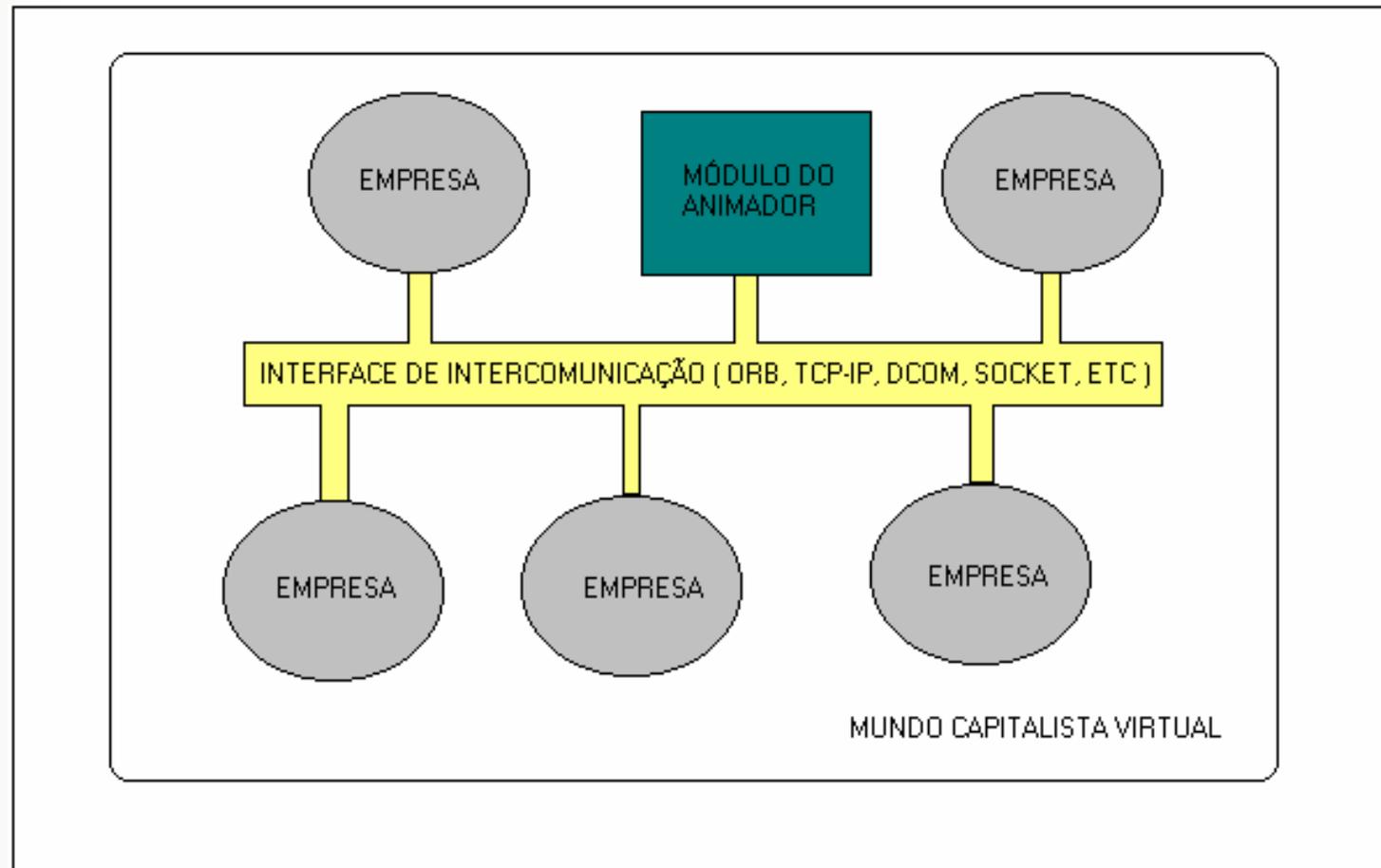
- "Agentes devem agir autonomamente para cumprir um conjunto de objetivos" Pattie Maes.
- "Um sistema de computador baseado em hardware ou software que desfruta as propriedades de: autonomia, capacidade social, reatividade e pro-atividade".
Wooldridge e Jennings



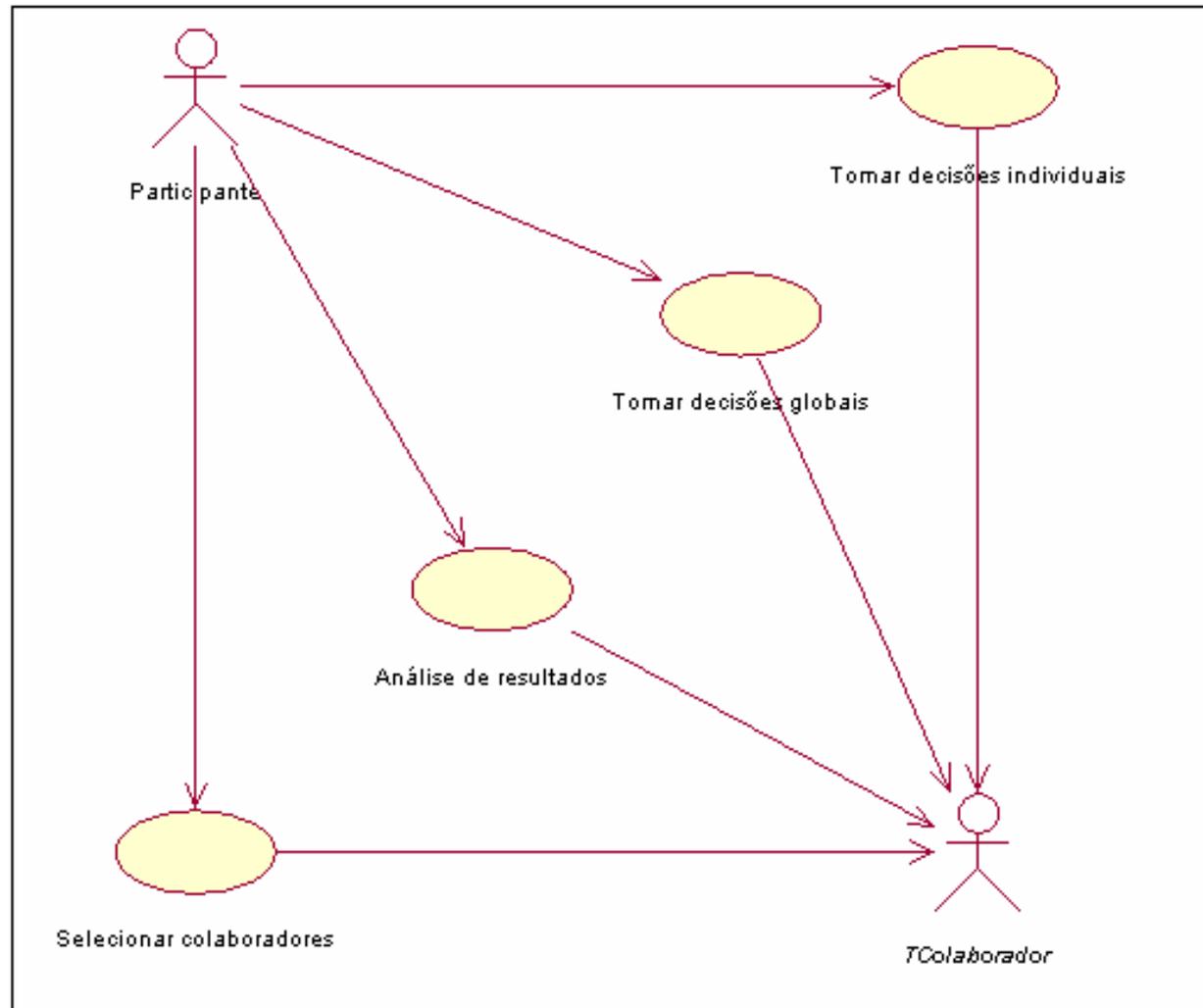
Sociedades de Agentes

- Uma sociedade de agentes pode ser definida, numa perspectiva muito abrangente, como uma tripla $\langle \mathbf{A}, \mathbf{M}, \mathbf{L} \rangle$ com um conjunto de agentes (\mathbf{A}) que compartilham um conjunto de meios (\mathbf{M}) e um conjunto de linguagens (\mathbf{L}) que permitem comunicação entre eles.

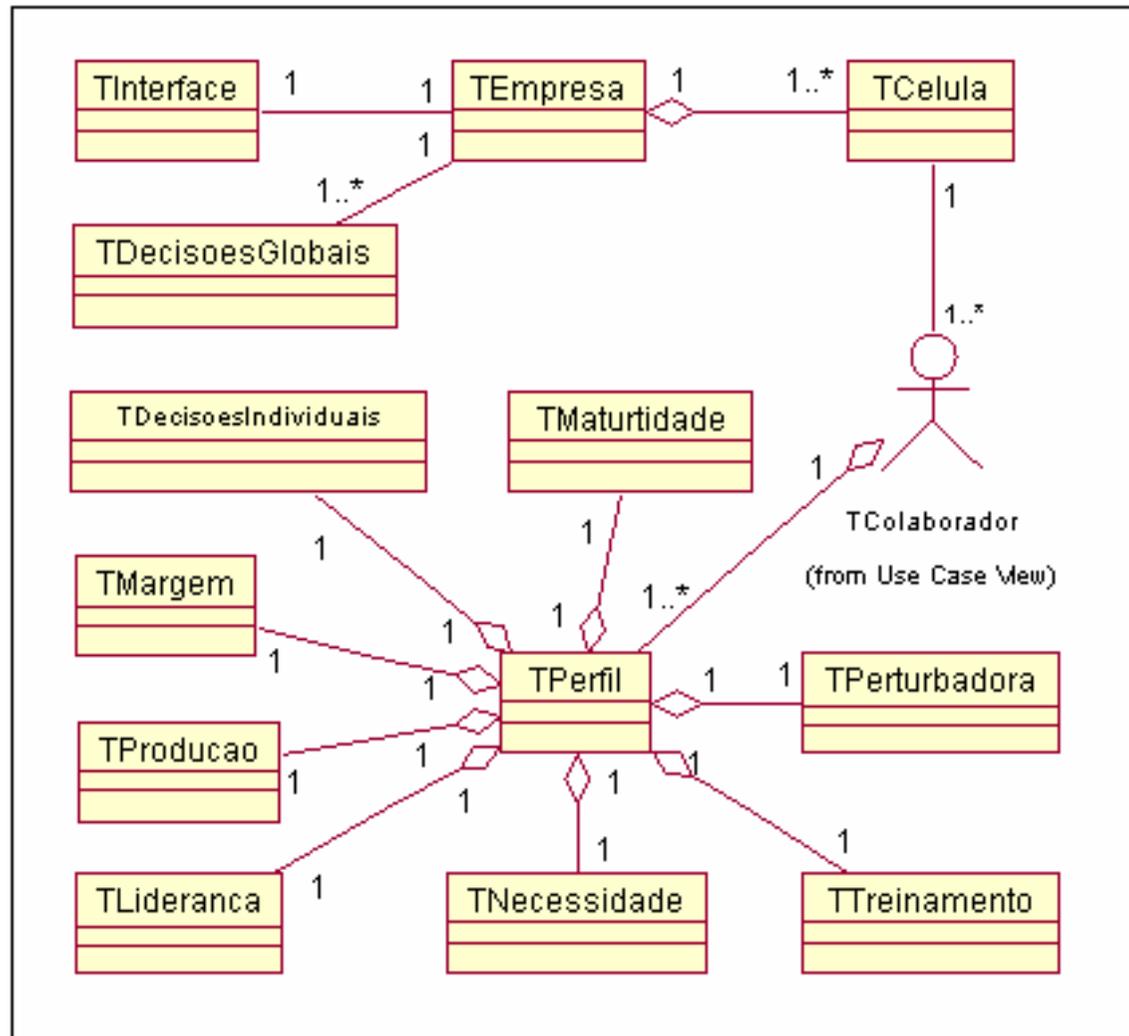
Desenvolvimento do protótipo



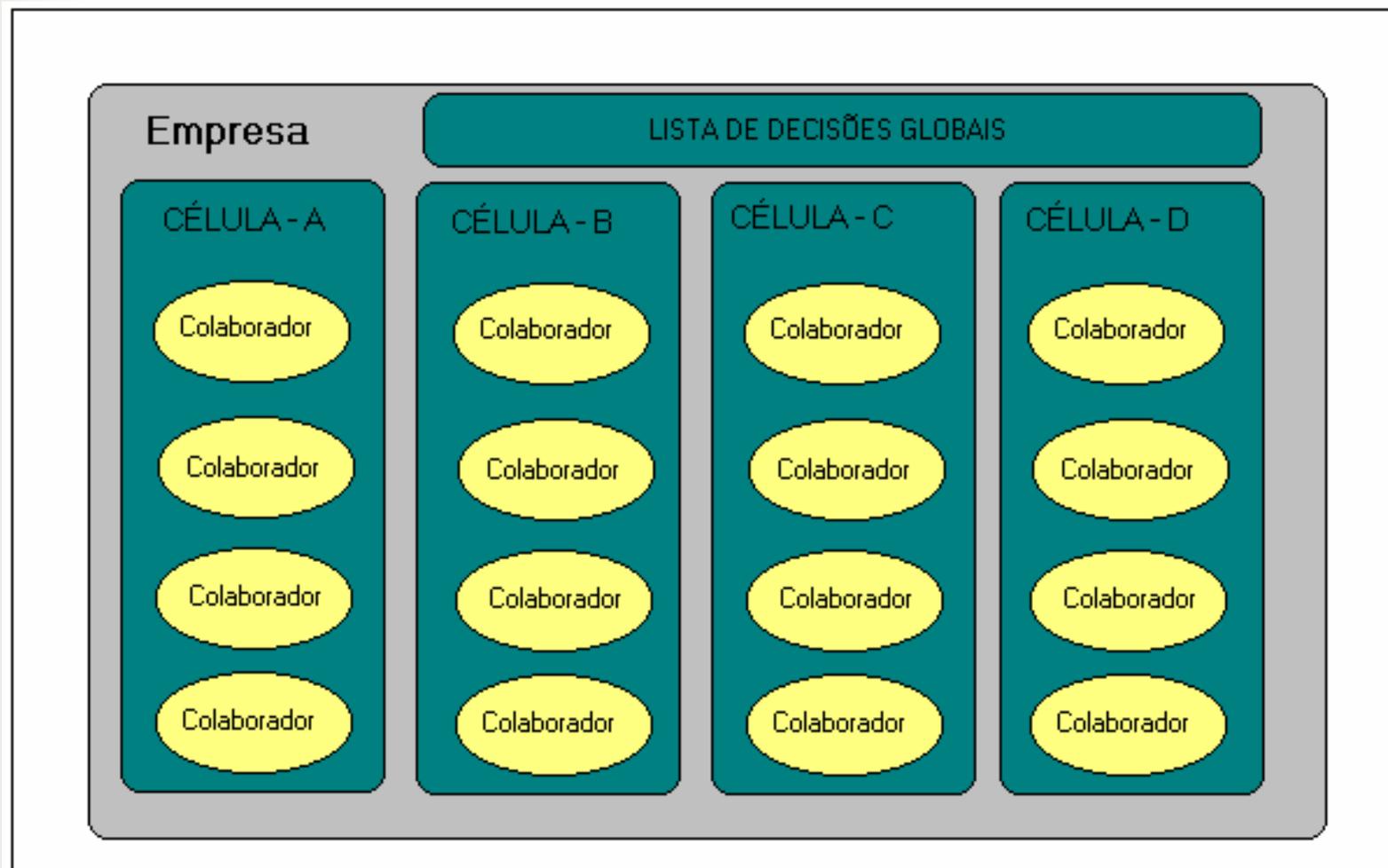
Desenvolvimento do protótipo



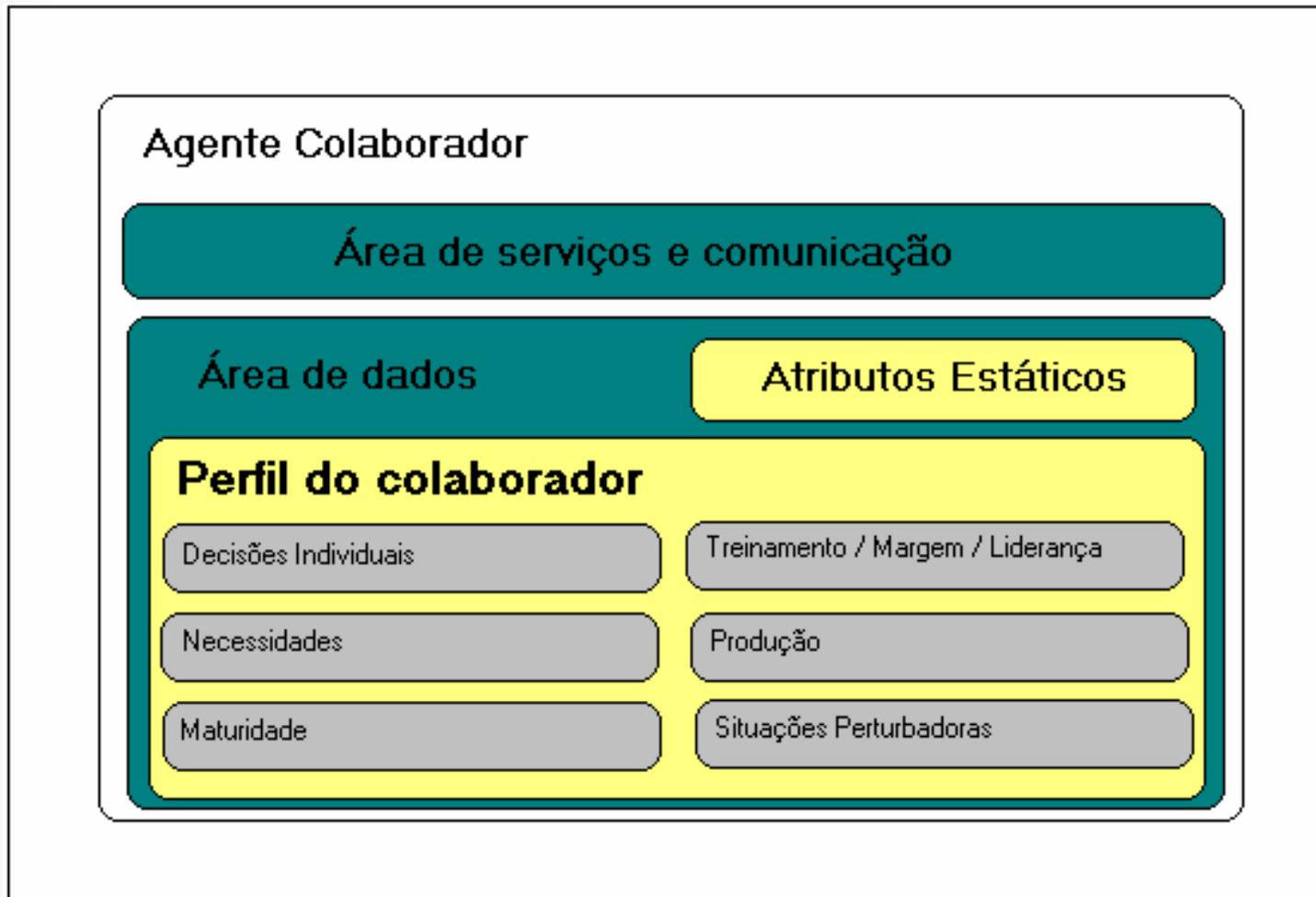
Desenvolvimento do protótipo



Desenvolvimento do protótipo



Desenvolvimento do protótipo





Desenvolvimento do protótipo

- 2 Camadas (interface e aplicação).
- Aplicação totalmente encapsulada dentro da classe empresa, facilitando a implementação de um módulo multi-empresas.
- Comunicação através de troca de mensagens.



Desenvolvimento do protótipo

- `Consulta.Sql.Clear;`
- `Consulta.Sql.Add (' Select Avg (Producao_Real) as Producao from Funcionarios where ' +`
- `' Codigo_Empresa = ' + IntToStr(Self.CODIGO_EMPRESA) +`
- `' and Período = ' + IntToStr(Self.PERÍODO - 1) +`
- `' and Cargo <> ''Operário'');`
- `Consulta.Open;`
- `Self.Producao_Media_Outros := Consulta.FieldByName('Producao').AsFloat;`

Desenvolvimento do protótipo

```
■ SomaProducao := 0;  
■     Cont := 0;  
■     //  
■     Celulas := F_Interface.Empresa.GetListaCelulas;  
■     For ContCel := 0 to Celulas.Count - 1 do  
■     begin  
■         Colaboradores := TCellula(Celulas.items[ContCel]).GetListaColaboradores;  
■         For ContCol := 0 to Colaboradores.Count - 1 do  
■         begin  
■             perfil_periodo_atual :=  
TPerfil(TColaborador(Colaboradores.Items[ContCol]).perfis.items[TColaborador(Colaboradores.It  
ems[ContCol]).getperiodo_atual]);  
■             if perfil_periodo_atual.GetCARGO[1] <> 'O' then  
■             begin  
■                 inc(Cont);  
■                 SomaProducao := SomaProducao + perfil_periodo_atual.Producao_real;  
■             end;  
■         end;  
■     end;  
■     Self.Media := SomaProducao / Cont;  
■     Result := Self.Media;
```



Desenvolvimento do protótipo

- Autonomia - Threads.
- Capacidade social - Comunicação com outros agentes.
- Reatividade - Início do processamento após tomada de decisões.
- Pro-atividade - Auto-processamento.



Conclusões

- Nenhuma alteração nos valores dos cálculos presentes no modelo original do Líder.
- Consistência do processamento como nas versões anteriores.
- Nova forma de comunicação entre os colaboradores tornou o sistema mais ágil.



Conclusões

- A utilização de agentes permitiu um modelo para o sistema mais próximo do real.
- Nova interface baseada neste modelo, permitiu uma maior interatividade entre os participantes e a evolução dos colaboradores.



Conclusões

- Rational Rose 2000, na modelagem do sistema, se mostrou eficiente e bastante completa no que diz respeito ao suporte a metodologia adotada (UML).
- Delphi apresentou estabilidade na implementação de threads, tornando sua utilização para este tipo de aplicação, satisfatória



Limitações

- Não estão presentes todos os relatórios disponíveis para a análise dos resultados do participante.
- Não está implementada a persistência dos agentes.



Sugestões

- Estudo de uma forma para a implementação da persistência dos agentes, no qual a utilização de bancos de dados orientados a objetos, ou tecnologia compatível com armazenamento de objetos em disco, ou ainda um mapeamento de como armazenar objetos em estruturas relacionais.



Sugestões

- Intercomunicação entre várias instâncias do sistema através de redes de computadores, simulando uma empresa e filiais, ou concorrentes.
- Criação de um agente para auxiliar o animador no acompanhamento do desempenho dos participantes.



Sugestões

- Relatórios dirigidos aos participantes que incluam históricos decisórios, e evolutivos dos colaboradores, aproveitando que neste novo modelo implementado, tanto os perfis, quanto as decisões individuais ficam armazenados dentro do agente como seu estado interno, funcionando como um histórico da simulação.



Bibliografia básica

- [FUR1998] FURLAN, José Davi. **Modelagem de objetos através da UML – The unified modeling language.** São Paulo: Makron Books, 1998.
- [JEN1998] JENNINGS, Nicholas R., WOOLDRIDGE, Michael. **Agent technology foundations, applications and markets.** New York: Springer, 1998.



Bibliografia básica

- [NIV1998] NIVEIROS, Sofia Inês.
Estudo e aperfeiçoamento do modelo das maturidades dos funcionários no jogo de empresas líder. Florianópolis, 1998.
Dissertação. Programa de Pós-Graduação em Engenharia de Produção. Universidade Federal de Santa Catarina.