

7 Referências

BELTRÁN, F.; ROGGENDORF, M.; **A simulation model for the dynamic allocation of network resources in a competitive wireless scenario**, Mobility Aware Technologies and Applications, Second International Workshop, MATA 2005, Montreal, Canada, October 17-19, 2005, Proceedings. Lecture Notes in Computer Science 3744, ISBN 3-540-29410-4. 54-64.

CHEN G.; KOTZ D. **A Survey on Context-Aware Mobile Computing Research**, Technical report TR2000-381, 2000.

CHEN L. J.; SUN T.; CHEN B.; RAJENDRAN V.; GERLA M. **A Smart Decision Model for Vertical Handoff**, Department of Computer Science, University of California at Los Angeles, Los Angeles, USA, 2004.

DEVLIC A. **Context-addressed communication dispatch**, Msc Thesis, Royal Institute of Technology (KTH), October 2006.

DEY A. K.; ABOWD G. D. **Towards a better understanding of context and contextawareness**, In Conference on Human Factors in Computing Systems CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness, April 2000.

FELICÍSSIMO, C. (2006b). **Providing Contextual Information in Open MAS**, 2007.

FELICÍSSIMO, C.; LUCENA, C.; BRIOT, J.-P.; AND CHOREN, R. (2006b). **An approach for contextual regulations in open MAS**, In Proceedings of AAMAS06 - Workshop on Agent-Oriented Information Systems (AOIS-06).

FERBER, J.; GUTKNECHT, O.; MICHAEL, F.; **From Agents to Organizations: an Organization View of Multi-Agent Systems**, In Proc. of AOSE-2003. Australia, 2003.

FONTOURA, M. **Enhancing Framework Design and Utilization** – ACM'99 Student Research Contest (Graduate), 1999.

FROEHLICH, G.; HOOVER, H. J.; LIU, L.; SORENSON, P. **Requirements for Hooks Tools** – University of Alberta, 1998.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison-Wesley, Reading, 1995.

GARCIA, A. F. **Objetos e Agentes: Uma Abordagem Orientada a Aspectos**. Tese (Doutorado em Informática) – Departamento de Informática, Pontifícia

Universidade Católica do Rio de Janeiro, 2004.

JENNINGS, N. **Agent-Oriented Software Engineering**. Proceedings of the 12th International Conference on Industrial and Engineering Applications of Artificial Intelligence, pp. 4-10, 1999.

JENNINGS, N.; **On Agent-Based Software Engineering. Artificial Intelligence**, 117 (2) p. 277-296. 2000.

JENNINGS, N.; WOOLDRIDGE, M. **Agent-Oriented Software Engineering**. Handbook of Agent Technology, J. Bradshaw (ed.). AAAI/MIT Press, 2000.

KHEDR, M; KARMOUCH, A. (1995) **ACAI: Agent-Based Context-aware Infrastructure for Spontaneous Applications**. Journal of Network & Computer Applications, Vol. 28, Issue 1, p. 19-44.

MATTSSON M. **Object-Oriented Frameworks: A Survey of Methodological Issues**, Licentiate Thesis, Department of Computer Science, Lund University, 1996.

MATTSSON, M.; BOSCH, J.; FAYAD, M. E. **Framework Integration Problems, Causes, Solutions** – Communications of the ACM, 1999.

MOKHTAR S.; FOURNIER D.; GEORGANTAS N.; ISSARNY V. **Context-Aware Service Composition in Pervasive Computing Environments**, 2nd International Workshop RISE 2005..

PASCOE J. **Context-Aware Software**, PhD thesis, Computing Laboratory, University of Kent at Canterbury, August 2001.

ROGGENDORF, M.; BELTRÁN, F.; GUTIERREZ, J. **Architecture and implementation of an agent-based simulation tool for market-based pricing in Next-Generation Wireless Networks**, Proceedings of TridentCOM 2006, 2nd International IEEE/Create-Net Conference on Testbeds, Barcelona, Spain, March 1-3, 2006.

SCHILIT B.; ADAMS N.; WANT R. **Context-aware computing applications**, In IEEE Workshop on Mobile Computing Systems and Applications, 1994.

SILVA A. J. S. **Introdução a redes sem fio**, Rede Nacional de Ensino e Pesquisa (RNP), Boletim bimestral sobre tecnologia de redes, volume 2, número 5, ISSN 1518-5974, 15 de maio de 1998.

SOUZA J. L.; TUDE E. **Tutorial de Telefonia Celular**, Revista eletrônica Teleco, www.teleco.com.br, 09 de Dezembro de 2002.

SUN J. Z.; RIEKKI J.; SAUVOLA J.; JURMU M. **Towards Connectivity Management Adaptability: Context Awareness in Policy Representation and End-to-end Evaluation Algorithm**, 3rd international conference on Mobile and ubiquitous multimedia. College Park, Maryland. ISBN:1-58113-981-0, 2004.

THOMAS, G. AND WILLIAMS, A.; **Roles in the Context of Multiagent Task**

Relationships, In Proc. of AAAI Fall Symposium “Roles, an Interdisciplinary Perspective: Ontologies, Programming Languages, and Multiagent Systems”, TR FS-05-08. ISBN 978-1-57735-254-9., 2005.

WEYNS, D.; PARUNAK, H.; MICHEL, F.; HOLVOET, T. AND FERBER, J.; **Environments for Multiagent Systems State-of-the-Art and Research Challenges**. In Proc. of E4MAS-2005, p.1-47. ISBN 3-540-24575-8, 2005.

Anexo A - Modelagem do Framework

Neste anexo será descrita a modelagem do framework, com todos os seus diagramas de caso de uso, seqüência e classes e pacotes utilizados. Uma descrição dos casos de uso e dos padrões de projetos utilizados.

O diagrama de caso de uso é um dos diagramas da linguagem de modelagem UML que visa representar objetos, logo não é o ideal para representar um sistema multi-agentes, no entanto o diagrama de caso de uso nos ajuda no entendimento do sistema.

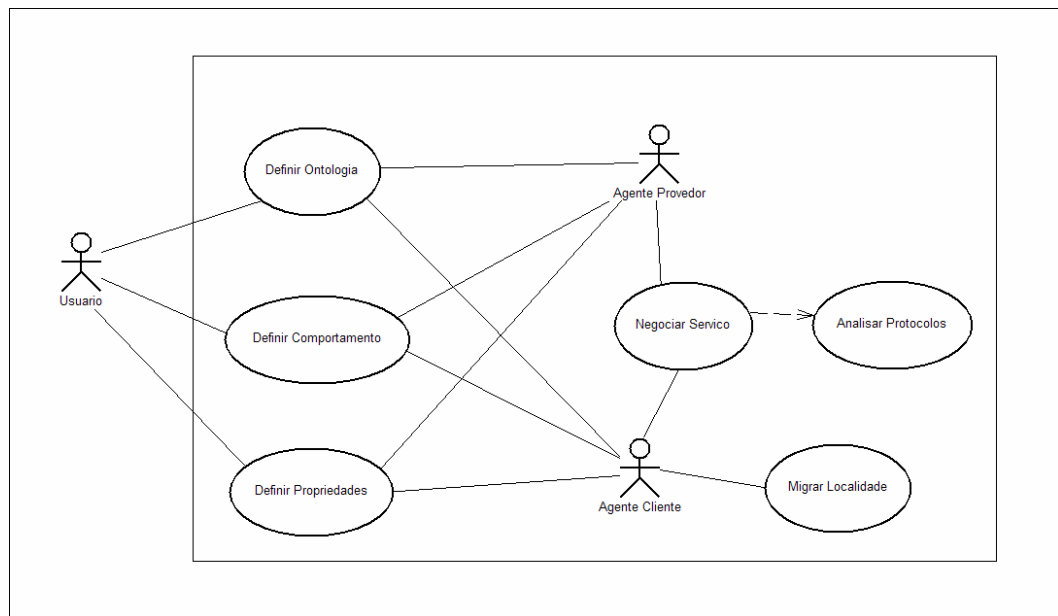


Figura 34. Diagrama de caso de uso.

Abaixo serão apresentadas as descrições de casos de uso:

Tabela 1. Diagrama de caso de uso CDU00.

Caso de Uso CDU00:	Definir Ontologia
Ator(es):	Usuário, Agente Provedor, Agente Cliente.
Descrição:	Antes da simulação o usuário define seu modelo da ontologia, durante a simulação, os agentes Provedor e cliente utilizam este modelo, que irão orientar suas ações.
Curso Típico dos Eventos	

Nº.	Ações
1	Usuário define as instâncias das classes de Ontologia.
2	Sistema lê os dados da Ontologia e as disponibiliza.
3	Agente Provedor carrega os Comportamentos definidos para ele.
4	Agente Cliente carrega os Comportamentos definidos para ele.
5	Sistema informa sucesso na carga.
---- Fim do Caso de Uso ----	
Curso Alternativo Nº. 1 – Passo 2 do Curso Típico: O sistema não consegue ler dados da Ontologia	
Nº	Ações
1	Sistema informa que não conseguiu ler dados da Ontologia
---- Fim do Caso de Uso ----	
Curso Alternativo Nº. 2 – Passo 3 do Curso Típico: Agente Provedor já carregou seu(s) Comportamento(s)	
Nº	Ações
1	Sistema informa que o(s) Comportamento(s) já foram carregados
---- Fim do Caso de Uso ----	
Curso Alternativo Nº. 3 – Passo 4 do Curso Típico: Agente Cliente já carregou seu(s) Comportamento(s)	
Nº	Ações
1	Sistema informa que o(s) Comportamento(s) já foram carregados
---- Fim do Caso de Uso ----	

Tabela 2. Diagrama de caso de uso CDU01.

Caso de Uso CDU01:	Definir Comportamento
Ator(es):	Usuário, Agente Provedor, Agente Cliente.

Descrição:	O usuário define os Comportamentos dos Agentes Provedor e Cliente que serão utilizados para tratar as informações de Contexto.
Curso Típico dos Eventos	
Nº.	Ações
1	Usuário define as classes que representaram o Comportamento dos Agentes.
2	O sistema identifica quais são as classes que deverão ser instanciadas para cada Agente após ler a Ontologia.
3	O sistema instancia as classes e associa estes Comportamentos ao Agente responsável.
4	O Agente Provedor utiliza os Comportamentos adicionados pelo sistema para tratar as informações de Contexto.
5	O Agente Cliente utiliza os Comportamentos adicionados pelo sistema para tratar as informações de Contexto.
6	Sistema informa sucesso na Carga dos Comportamentos.
---- Fim do Caso de Uso ----	
Curso Alternativo Nº. 1 – Passo 2 do Curso Típico: Não foi possível identificar as classes que serão instanciadas.	
Nº	Ação
1	Sistema informa problemas na leitura da ontologia.
---- Fim do Caso de Uso ----	
Curso Alternativo Nº. 2 – Passo 4 do Curso Típico: O Agente Provedor não possui um Comportamento esperado.	
Nº	Ação
1	Sistema informa que o Agente Provedor não possui um Comportamento para tratar uma informação de Contexto.
---- Fim do Caso de Uso ----	

Curso Alternativo N°. 3 – Passo 5 do Curso Típico: O Agente Cliente não possui um Comportamento esperado.	
Nº	Ação
1	Sistema informa que o Agente Cliente não possui um Comportamento para tratar uma informação de Contexto.
---- Fim do Caso de Uso ----	

Tabela 3. Diagrama de caso de uso CDU02.

Caso de Uso CDU02:	Definir Propriedades
Ator(es):	Usuário, Agente Provedor, Agente Cliente.
Descrição:	Usuário define as propriedades que serão usadas pelos Agentes Provedor e Cliente.
Curso Típico dos Eventos	
Nº.	Ações
1	Usuário define as propriedades que serão utilizadas durante a simulação.
2	Sistema lê as propriedades para o início da simulação.
3	Agente Provedor identifica as propriedades específicas para sua simulação.
4	Agente Cliente identifica as propriedades específicas para sua simulação.
---- Fim do Caso de Uso ----	
Curso Alternativo N°. 1 – Passo 2 do Curso Típico: Não foi possível ler as propriedades.	
Nº	Ação
1	Sistema informa que não conseguiu ler as propriedades.
---- Fim do Caso de Uso ----	
Curso Alternativo N°. 2 – Passo 3 do Curso Típico: Não foi possível identificar as propriedades do Agente Provedor.	
Nº	Ação

1	Agente Provedor informa que não identificou as suas propriedades para a simulação.
---- Fim do Caso de Uso ----	
Curso Alternativo N°. 3 – Passo 4 do Curso Típico: Não foi possível identificar as propriedades do Agente Cliente.	
Nº	Ação
1	Agente Cliente informa que não identificou as suas propriedades para a simulação.
---- Fim do Caso de Uso ----	

Tabela 4. Diagrama de caso de uso CDU03.

Caso de Uso CDU03:	Migrar Localidade
Ator(es):	Agente Cliente
Descrição:	O Agente Cliente migra de localidade para simular mobilidade.
Curso Típico dos Eventos	
Nº.	Ações
1	Sistema identifica que o Agente terá mobilidade.
2	Agente recebe o tempo de migração determinado em suas propriedades.
3	Após o tempo de migração esperado, o agente escolhe aleatoriamente uma localidade para migrar.
4	Agente migra de localidade.
---- Fim do Caso de Uso ----	
Curso Alternativo N°. 1 – Passo 1 do Curso Típico: O Agente Cliente não terá mobilidade.	
Nº	Ação
1	Sistema identifica que o Agente não terá mobilidade.
---- Fim do Caso de Uso ----	
Curso Alternativo N°. 2 – Passo 2 do Curso Típico: O Agente Cliente não recebe o tempo de migração.	
Nº	Ação

1	Agente Cliente não identifica em suas propriedades o tempo de migração.
---- Fim do Caso de Uso ----	

Tabela 5. Diagrama de caso de uso CDU04.

Caso de Uso CDU04:	Negociar Serviço
Ator(es):	Agente Provedor, Agente Cliente, DFAgent.
Descrição:	Agente Provedor negocia um serviço com o Agente Cliente.
Curso Típico dos Eventos	
Nº.	Ações
1	Agente Cliente escolhe um serviço.
2	Agente Cliente solicita ao serviço de páginas amarelas a lista de Agentes Provedoras que oferecem o serviço escolhido.
3	Agente Cliente solicita a todos os Agentes Provedores da lista recebida, uma proposta de utilização do serviço.
4	Agente Provedor recebe a solicitação de proposta por utilização de um serviço.
5	Agente provedor executa o CDU05.
6	Agente provedor envia proposta ao Agente Cliente.
7	Agente Cliente recebe propostas.
8	Agente Cliente executa CDU05.
9	Agente Cliente envia aceitação de proposta a um dos Agentes Provedores e envia rejeição de proposta aos demais.
10	Agente Provedor envia mensagem de sucesso na negociação.
---- Fim do Caso de Uso ----	
Curso Alternativo Nº. 1 – Passo 3 do Curso Típico: Não existem Provedoras para o serviço solicitado.	

Nº	Ação
1	Agente Cliente não solicita proposta para o serviço escolhido pois não existem Provedoras na localidade.
---- Fim do Caso de Uso ----	
Curso Alternativo Nº. 2 – Passo 4 do Curso Típico: Agente Provedor não oferece serviço solicitado.	
Nº	Ação
1	Agente Provedor não recebe solicitação de proposta pois não oferece o serviço requisitado.
---- Fim do Caso de Uso ----	

Tabela 6. Diagrama de caso de uso CDU05.

Caso de Uso CDU05:	Analisar Protocolos
Ator(es):	Agente Provedor, Agente Cliente.
Descrição:	Agentes analisam informações de Contexto e executam ações.
Curso Típico dos Eventos	
Nº.	Ações
1	Agente analisa os Protocolos, Diretivas e SubDiretivas de sua ontologia.
2	Agente analisa condições e limiares de Contexto em Comportamentos específicos.
3	Agente executa ações resultantes da análise das informações de Contexto.
---- Fim do Caso de Uso ----	
Curso Alternativo Nº. 1 – Passo 3 do Curso Típico: Agente não executa ação	
Nº	Ação
1	Agente não executa nenhuma ação, pois os limiares das informações de Contexto não foram alcançados.
---- Fim do Caso de Uso ----	

Diagramas de seqüência:

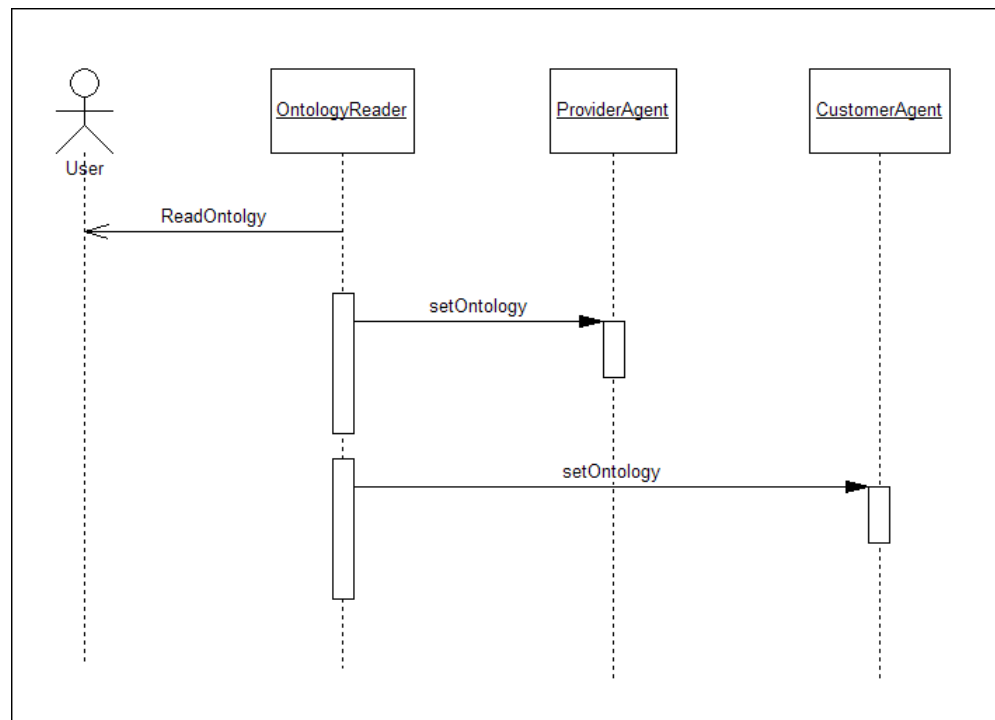


Figura 35. Definir Ontologia (Diagrama de Seqüência CDU00).

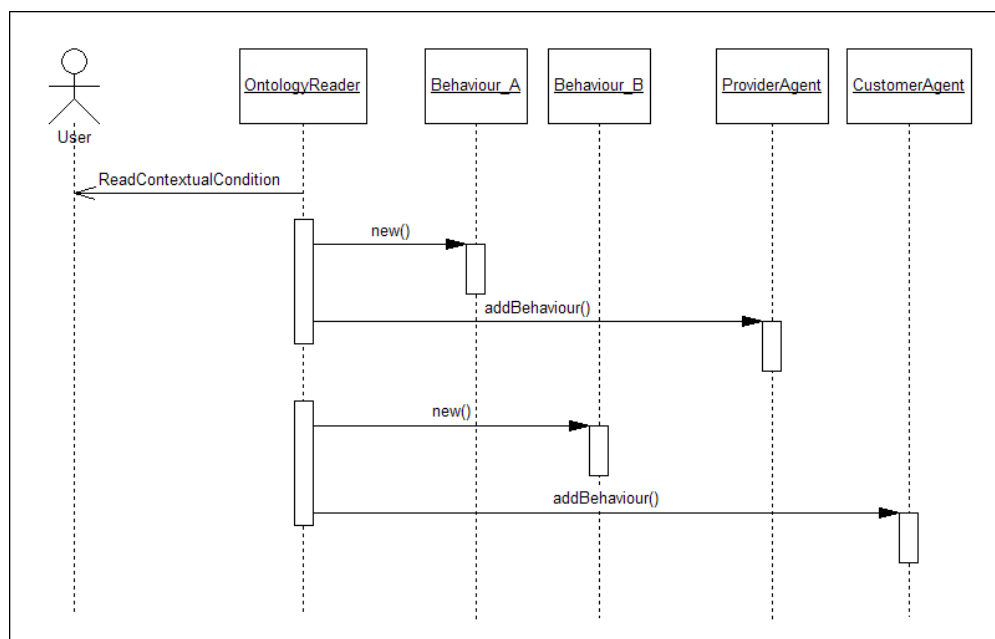


Figura 36. Definir Comportamento (Diagrama de Seqüência CDU01).

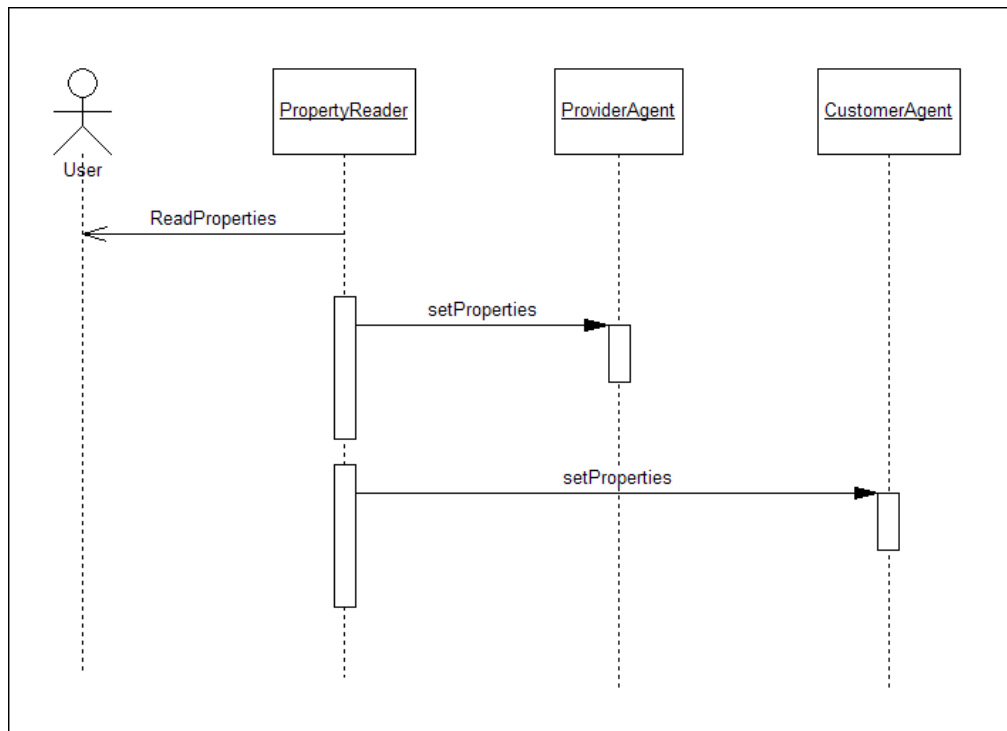


Figura 37. Definir Propriedades (Diagrama de Seqüência CDU02).

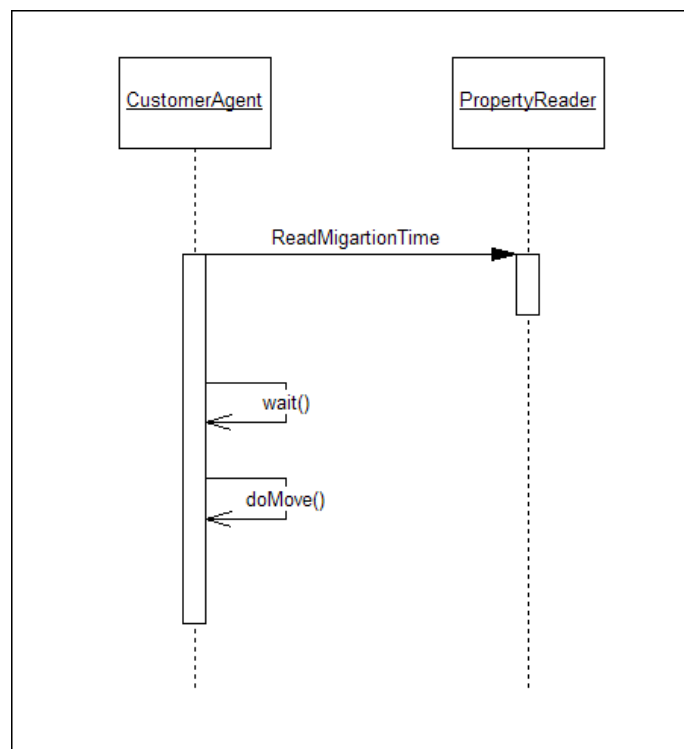


Figura 38. Migrar Localidade (Diagrama de Seqüência CDU03).

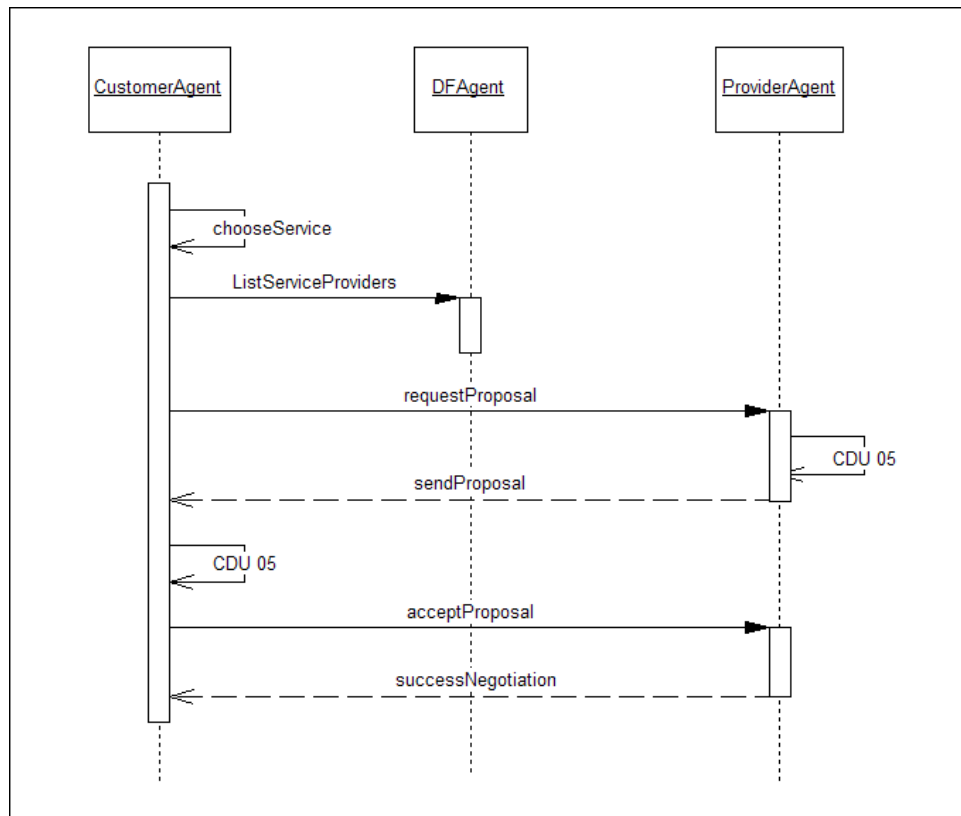


Figura 39. Negociar Serviço (Diagrama de Sequência CDU04).

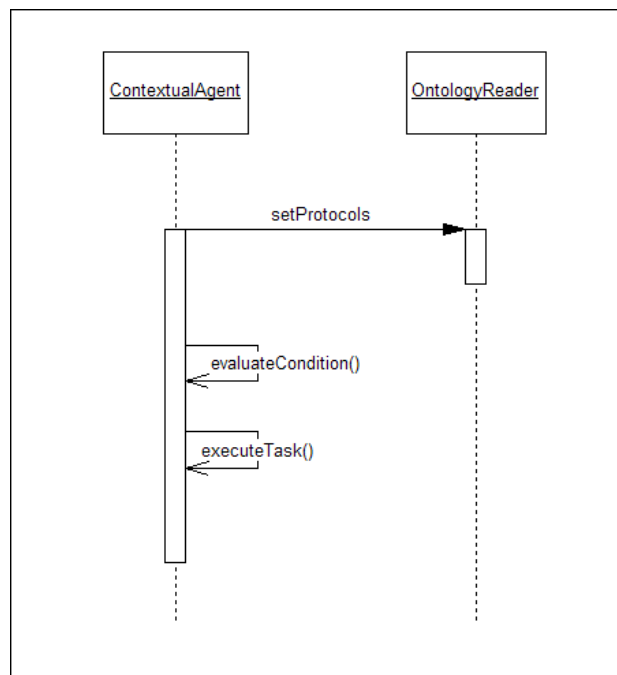


Figura 40. Analisar Protocolos (Diagrama de Sequência CDU05).

O diagrama de Pacotes é composto por quatro pacotes: behaviour (onde são modelados os comportamentos dos agentes), application (onde se encontram as Classes responsáveis instanciação e controle da aplicação), agent (onde estão modelados os agentes da simulação) e ontology (onde estão modeladas as Classes da Ontologia). Estes pacotes representam o

modelo do framework, em cada pacote estão modeladas as classes que permitem a dinâmica da simulação.

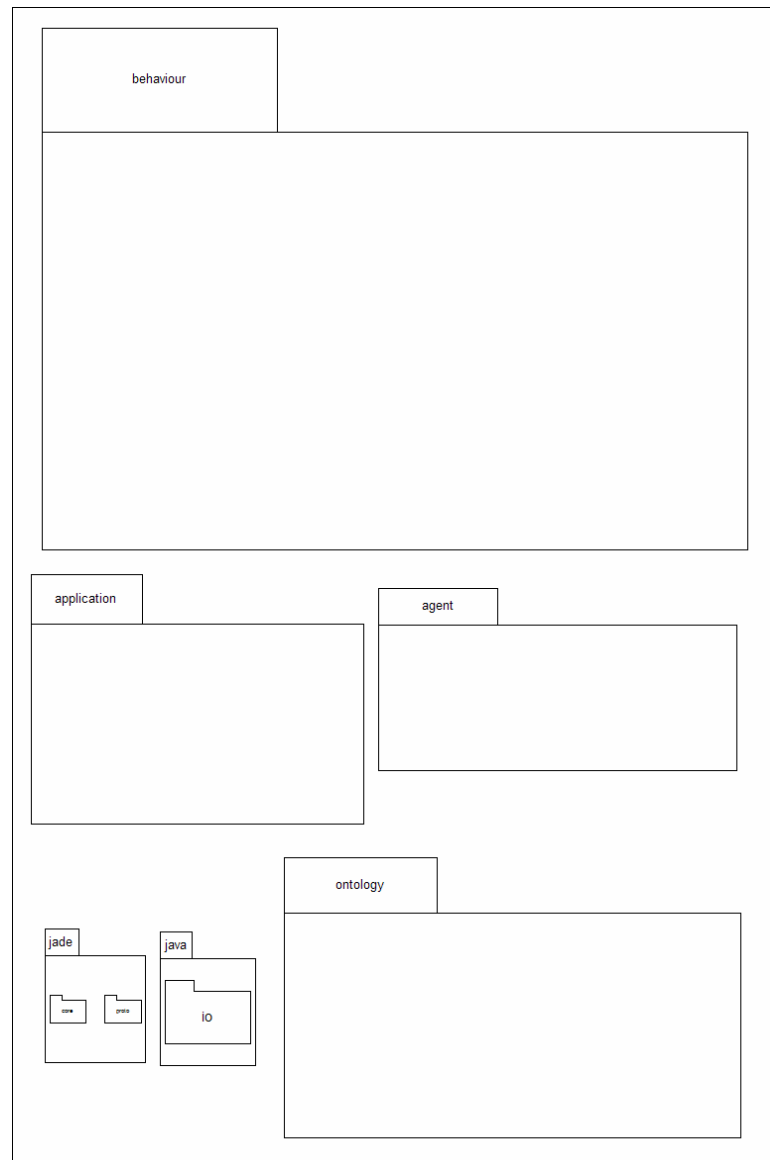


Figura 41. Diagrama de Pacotes.

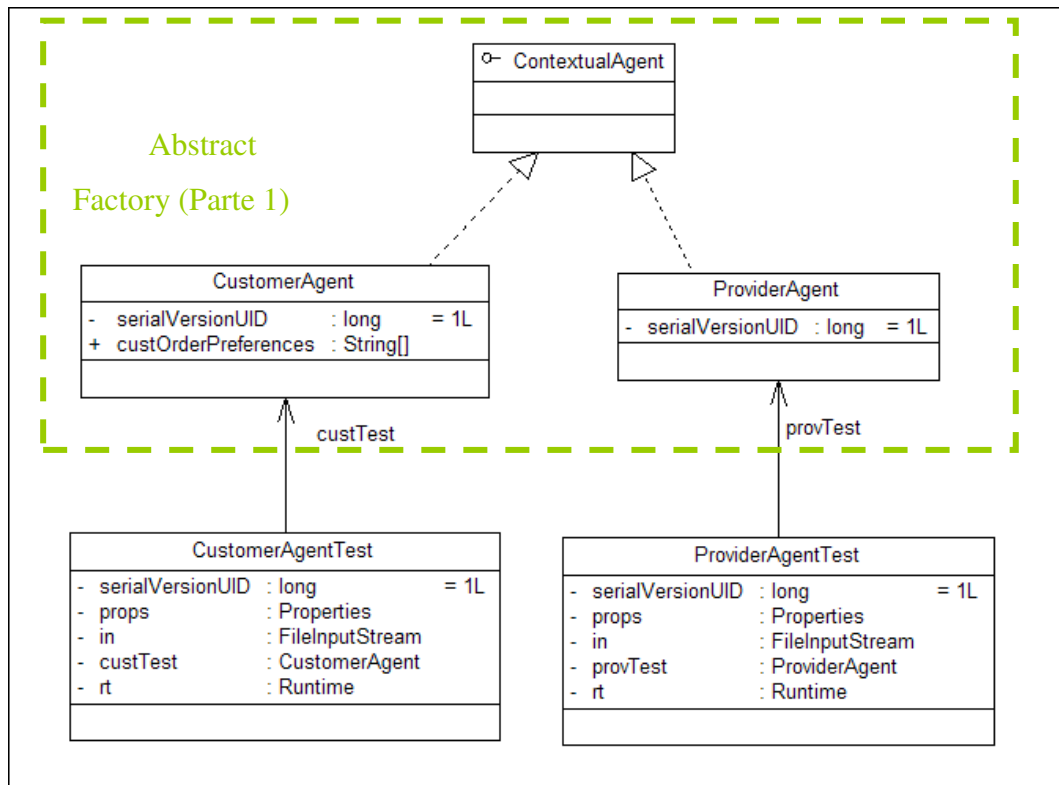


Figura 42. Diagrama de Classes (Pacote Agent).

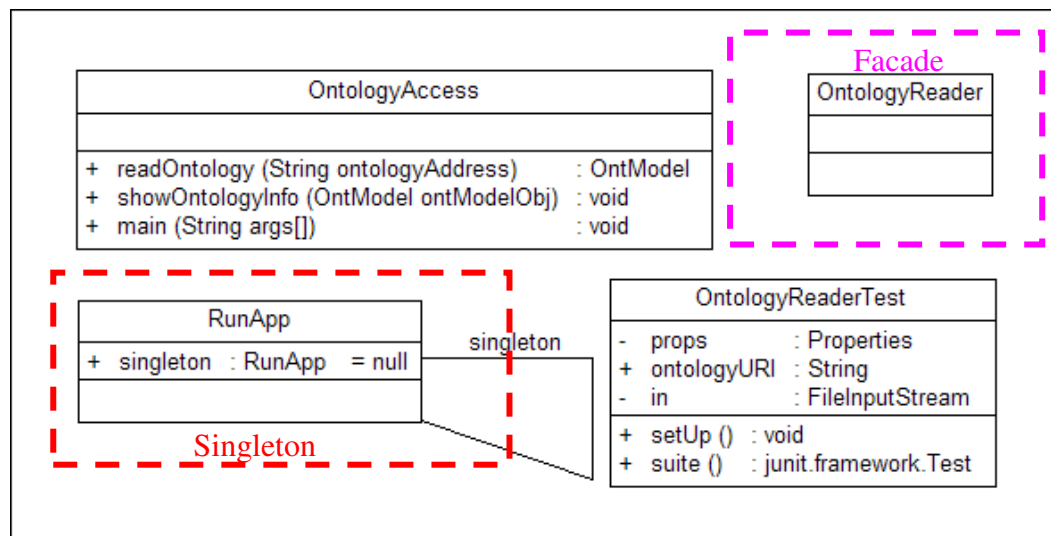


Figura 43. Diagrama de Classes (Pacote Application).

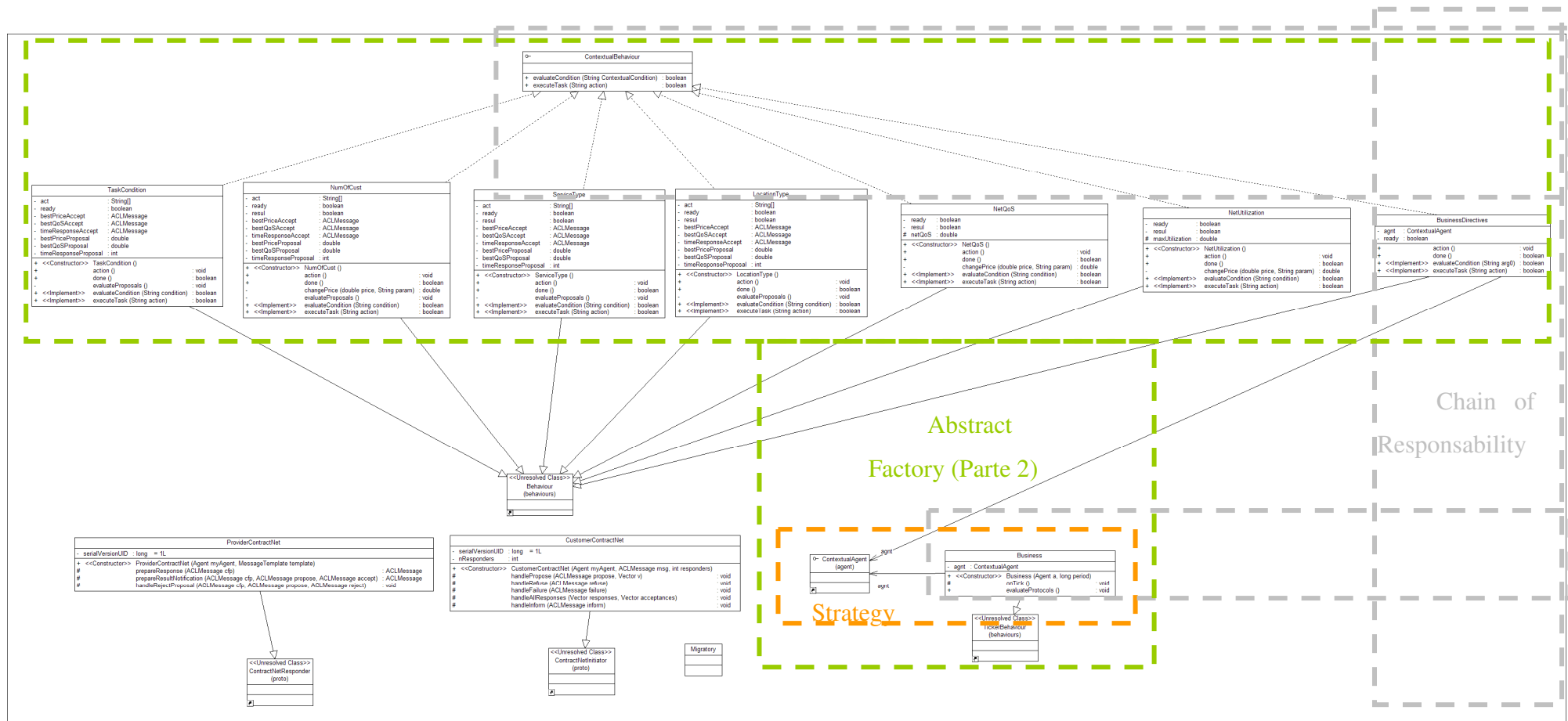


Figura 44. Diagrama de Classes (Pacote Behaviour do ContextualNGWN).

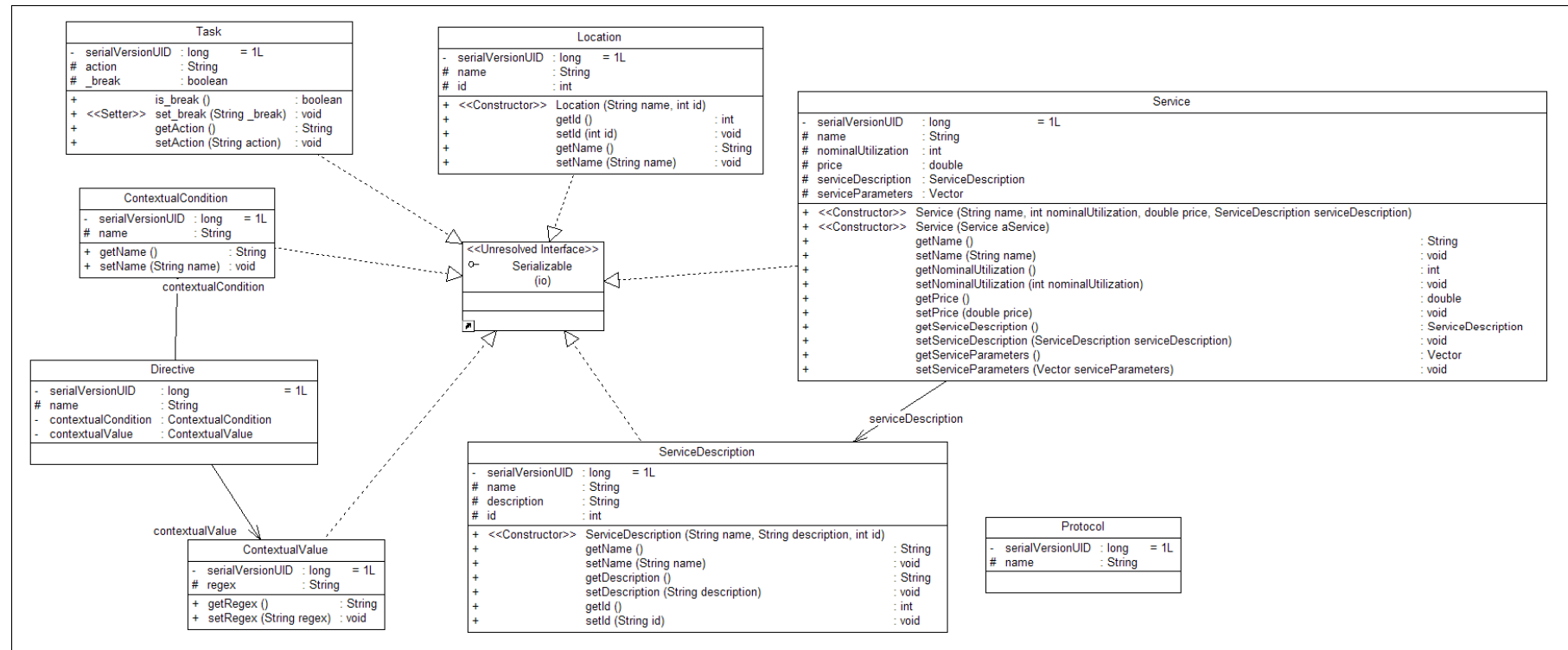


Figura 45. Diagrama de Classes (Pacote Ontology).

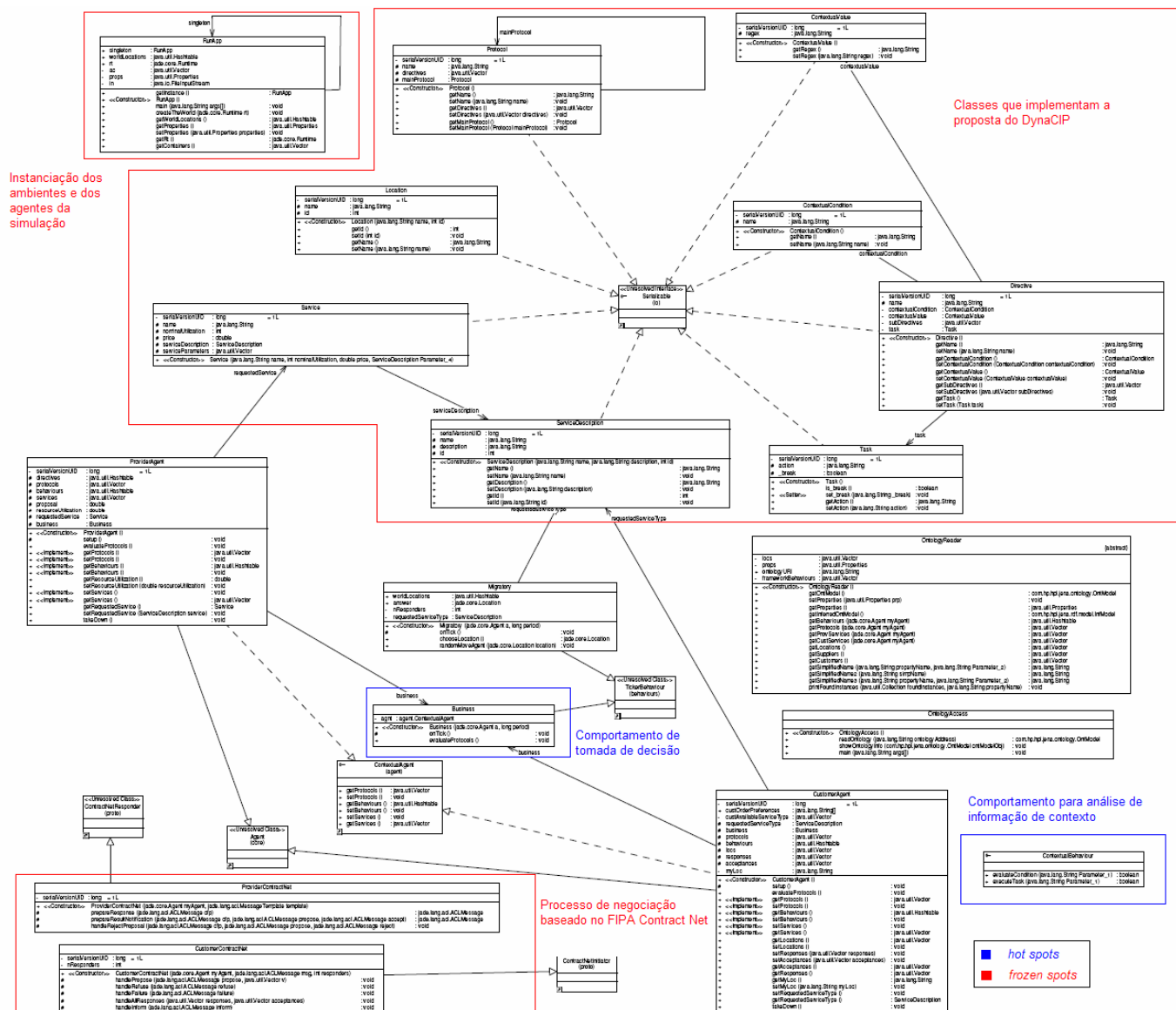


Figura 46. Hot spots e Frozen spots do diagrama de classes.

O diagrama de classe está baseado em cinco padrões de projeto: Facade, Singleton, Strategy, Chain of Responsibility, Abstract Factory. Para o desenvolvimento destes padrões foram levadas em consideração as abordagens contidas em (Gamma et al., 1995):

Facade: A intenção é fornecer uma interface unificada para um conjunto de interfaces em um subsistema e, dessa forma, estruturar um sistema em subsistemas para reduzir a complexidade. A classe OntologyReader implementa este padrão conduzindo todas as outras a classe de acesso a Ontologia.

Singleton: Para garantir que uma classe tenha somente uma instância e fornecer um ponto global de acesso para a mesma. A Classe de inicialização da simulação implementa este padrão.

Strategy: Define uma família de algoritmos, encapsula as mesmas de modo que o algoritmo varie independentemente dos clientes que o utiliza. Este padrão de projeto tem uma grande utilidade para o agente contextual já que este agente possui diferentes funções na simulação (estratégias) Pensando numa expansão deste projeto o padrão Strategy foi utilizado para todos os outros agentes, já que este pode ter estratégias diferentes em outras situações de simulação.

Chain of Responsibility: Evita o acoplamento do remetente de uma solicitação ao seu receptor, ao dar a mais de um objeto a oportunidade de tratar a solicitação. Neste sistema, o Padrão chain of responsibility é implementado através de diferentes possibilidades de implementação do algoritmo que analisa os protocolos da ontologia. O comportamento Business do framework delega a responsabilidade de avaliar os protocolos ao comportamento BusinessDirective a ser implementado pela aplicação. A implementação do padrão é feita inteiramente por meio de Behaviours do JADE.

Abstract Factory: Fornece uma interface para a criação de uma família de objetos relacionados ou dependentes sem especificar suas classes concretas. Este padrão de projeto foi utilizado para a criação dos agentes JADE onde os agentes Provedor e Cliente e criação dos Comportamentos utilizados na simulação.

Anexo B - O *framework* em outros domínios de aplicação

As características do *framework* não se restringem ao domínio de redes sem fio de nova geração. Problemas que envolvem provedoras/fornecedoras e clientes/usuários na negociação de um produto/serviço, independente do domínio da aplicação, talvez possam utilizar o *framework* como uma ferramenta para simulação. Negociação de serviços de empresas aéreas e negociação de serviços de empresas seguradoras são exemplos de possíveis aplicações que poderiam utilizar o *framework* em suas simulações.

Em negociação com empresas aéreas o tipo de serviço/produto poderia ser, por exemplo, os assentos nas aeronaves/passagens aéreas, o transporte de cargas e fretamento. O contexto avaliado pelas empresas aéreas poderia ser, por exemplo, horário do voo, milhas do cliente, ocupação da aeronave e número de voos entre origem-destino. O contexto avaliado pelos clientes poderia ser, por exemplo, condições de pagamento, tipo de aeronave, número de escalas e tempo de voo.

Em negociação com empresas seguradoras o tipo de serviço/produto poderia ser, por exemplo, seguro de vida, seguro de imóvel e seguro de automóvel. O contexto avaliado pela empresa seguradora poderia ser, por exemplo, número de sinistros na localidade, histórico do contratante, avaliação de mercado e tempo de contratação. O contexto avaliado pelos clientes poderia ser, por exemplo, quantidade de empresas que oferecem o seguro, condições de pagamento e tempo máximo de ressarcimento prometido.

Neste anexo, será descrito apenas a modelagem da instância do DynaCIP para o domínio de negociação com empresas aéreas, ficando como trabalho futuro a implementação de uma aplicação para o domínio do problema modelado.

Na instância foram modeladas duas empresas aéreas, a AirLines1 e a AirLines2, ambas oferecem passagens aéreas para as mesmas categorias, Crianças de colo, crianças e adultos em classes econômica e executiva. As duas definem a ocupação nominal de um assento para cada passagem vendida, exceto para

crianças de colo que não ocupam assento. Cada tipo de passagem tem um preço nominal que irá sofrer ajustes após a análise das informações de contexto.

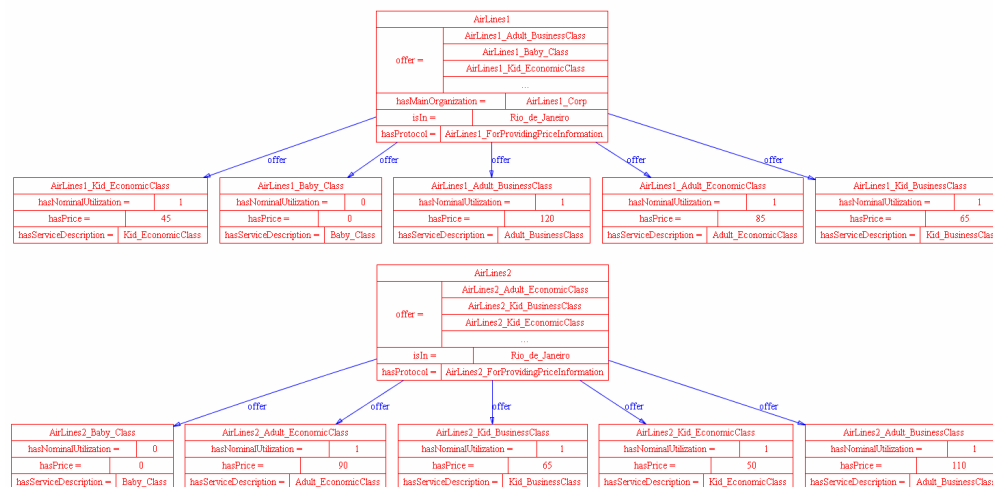


Figura 47. Instância de Airlines1 e Airlines2.

A empresa Airlines1 faz parte de uma organização corporativa, a AirLinesCorp, que possui um protocolo dar desconto de dois por cento aos passageiros que pertençam ao seu programa de milhagem. Além disso, a AirLines1 possui um protocolo com três diretivas com prioridades de leitura para prover os preços de seus bilhetes aéreos. A diretiva com a maior prioridade, determina um aumento de cinco por cento no preço da passagem caso só ela ofereça o voo para o local solicitado. Já a diretiva com segunda maior prioridade, define um desconto de dois por cento caso existam mais de cinco empresas aéreas oferecendo o voo para o local solicitado. A terceira e ultima diretiva, utiliza duas sub-diretivas para definir um desconto de um por cento quando a ocupação da aeronave estiver entre setenta e cinco e cem por cento de sua capacidade máxima e o passageiro fizer parte do programa de milhagem.

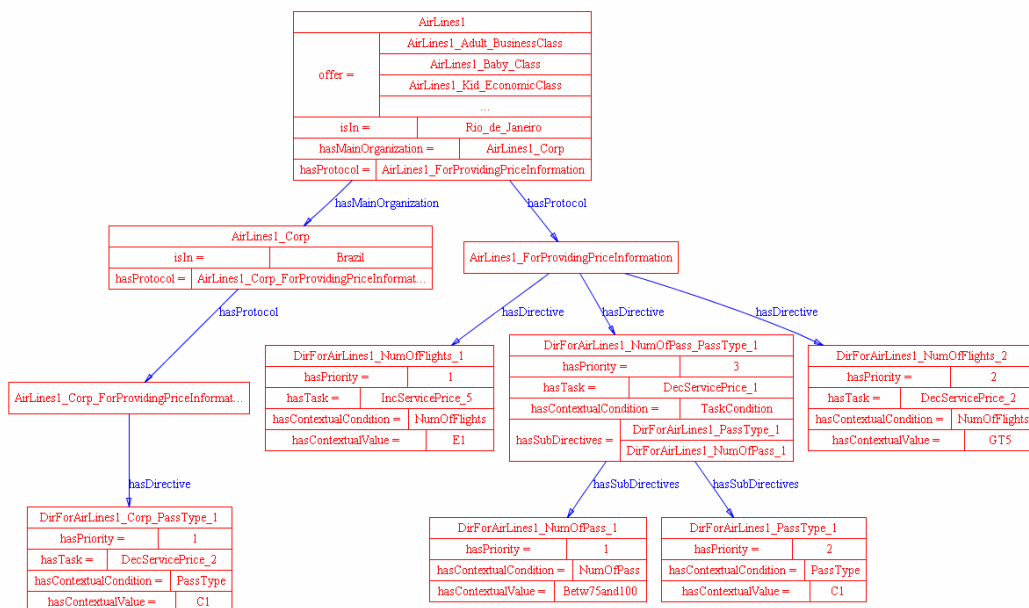


Figura 48. Instância de Airlines1 e suas Diretivas.

A empresa aérea Airlines2 possui um protocolo para definir seus preços baseados em três diretivas. A diretiva de maior prioridade define um desconto de um por cento quando o número de vôos oferecidos para uma localidade for maior do que quatro. A diretiva de segunda maior prioridade define um desconto de dois por cento quando a ocupação da aeronave estiver menor do que vinte e cinco por cento de sua capacidade máxima. A terceira e última diretiva define um desconto de dois por cento quando o passageiro fizer parte do programa de milhagem.

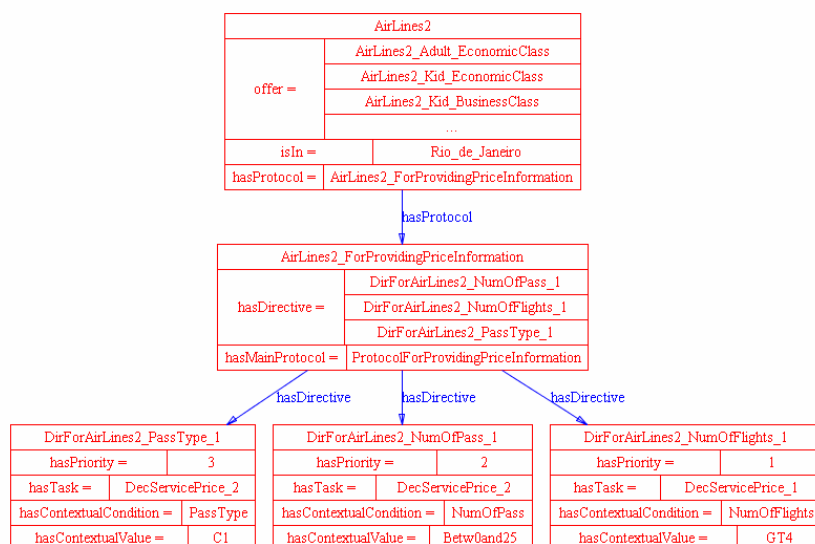


Figura 49. Instância de Airlines2 e suas Diretivas.

Foram modelados dois clientes João e Maria. João possui três diretivas para definir onde comprar sua passagem. A diretiva de maior prioridade, diz que se o menor número de conexões for maior do que quatro, deve ser escolhida a empresa aérea que oferecer o menor número de conexões. Na segunda diretiva de maior prioridade, se o menor tempo de voo for maior do que um dia, deve ser escolhida a empresa aérea que tiver menor tempo de voo. Na terceira diretiva, define-se a escolha da empresa aérea que oferecer o menor preço em qualquer outra situação.

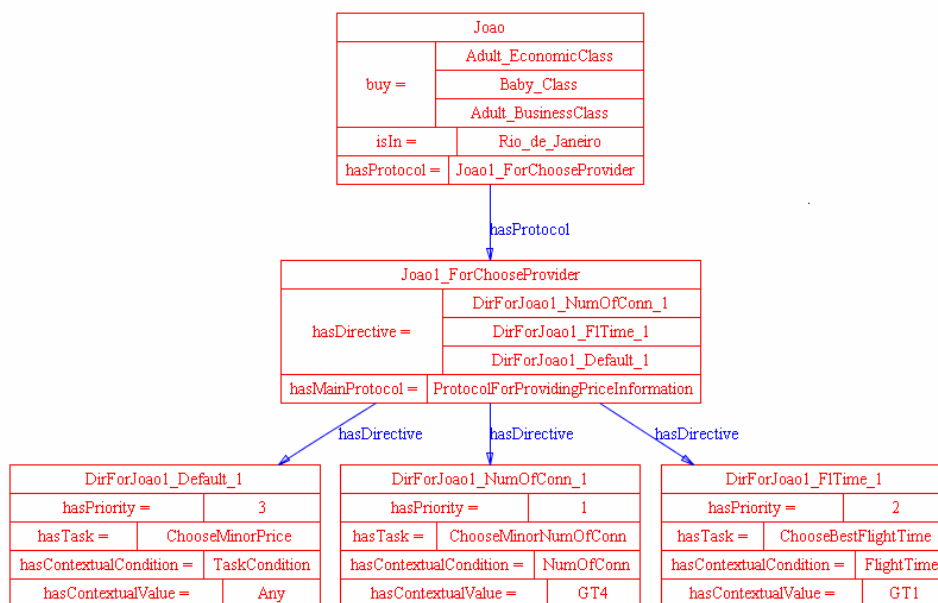


Figura 50. Instância de João e suas Diretivas.

Maria também possui três diretivas para definir a escolha de uma empresa aérea. Na diretiva de maior prioridade, se o menor número de conexões for maior do que um, a empresa que oferecer o menor número de conexões será escolhida. Na diretiva de segunda maior prioridade, se o menor tempo de voo for maior do que um dia, a empresa de menor tempo de voo deverá ser escolhida. Na terceira diretiva, a empresa de menor tempo de voo deverá ser escolhida em qualquer outra situação.

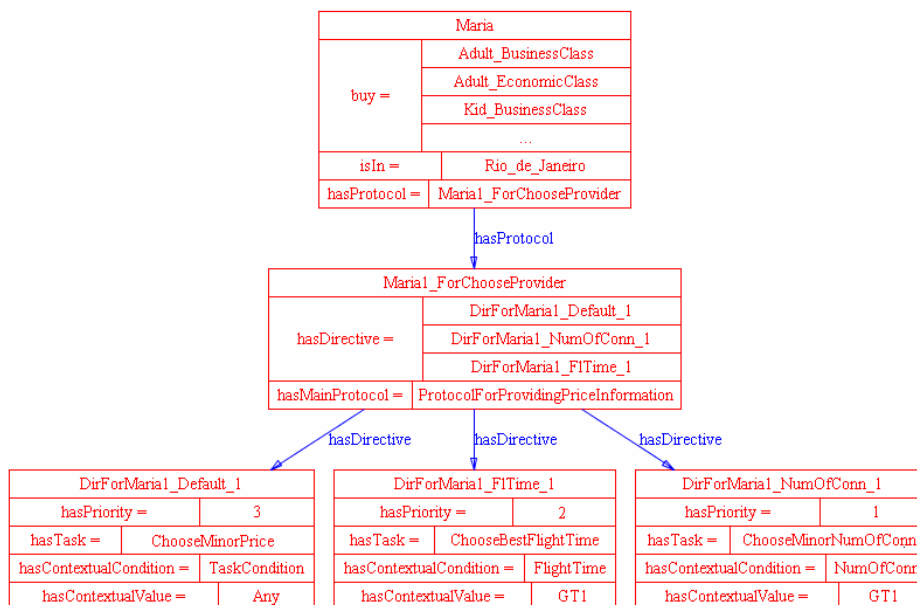


Figura 51. Instância de Maria e suas Diretivas.

Para analisar esta modelagem, podemos imaginar a seguinte situação: “O cliente João e Maria solicitam uma proposta para as empresas aéreas Airlines1 e AirLines2 de uma passagem aérea para um adulto na classe econômica do Rio de Janeiro para Sydney no dia 15/Nov. João faz parte do programa de milhagem da Airlines1 e não faz parte da AirLines2”.

AirLines1 (protocolo para definir o número de conexões e o tempo entre origem e destino) : Rio-Bue-Auk-Syd (4 con), (~2 dias), (ocup 80%)
AirLines1 (protocolo para definir o preço da passagem) : Preço Base=0.085*(84000 milhas) = R\$7140
 -Dir0 Passtype==C1 => Dec 2% : Preço= R\$6997,20
 -Dir1 NumOfFlights==1 => Inc 5% : Preço= R\$6997,20
 -Dir2 NumOfFlights>5 => Dec 2% : Preço= R\$6997,20
 -Dir3 Passtype==C1 e NumOfPass > 75% => Dec 1% : Preço= R\$6927.23
Airlines1 proposal: 4conexões, aprox 2 dias e preço R\$6927.23

AirLines2 (protocolo para definir o número de conexões e o tempo entre origem e destino) : Rio-La-Syd (3 con), (~3 dias), (ocup 20%)
AirLines2 (protocolo para definir o preço da passagem) : Preço Base=0.090*(84000 milhas) = R\$7560
 -Dir1 NumOfFlights>4 => Dec 1% : Preço= R\$7560
 -Dir2 NumOfPass < 25% => Dec 2% : Preço= R\$7408,80
 -Dir3 PassType==C1 => Dec 2% : Preço= R\$7408,80
Airlines2 proposal: 3conexões, aprox 3 dias e preço R\$7408,80

João (protocolo para definir a empresa aérea) :
 -Dir1 MinorNumOfConn>4 => Escolher empresa com menor número de conexões (Airlines2);
 -Dir2 MinorFlightTime>1 => **Escolher empresa com menor duração de tempo (Airlines1);**
 -Dir3 Qualquer outra condição => Escolher empresa com menor preço (Airlines1);

Maria (protocolo para definir a empresa aérea) :
 -Dir1 MinorNumOfConn>1 => **Escolher empresa com menor número de conexões (Airlines2);**
 -Dir2 MinorFlightTime>1 => Escolher empresa com menor duração de tempo (Airlines1);
 -Dir3 Qualquer outra condição => Escolher empresa com menor preço (Airlines1);

Figura 52. Análise do resultado de uma possível negociação com empresas aéreas.

Para a situação descrita, o João escolheria a empresa AirLines1, pois o menor número de conexões é três não satisfazendo a primeira diretiva. Porém o menor tempo de voo é maior do que um dia, logo a diretiva dois é satisfeita escolhendo a empresa que oferece o menor tempo de voo, a AirLines1. Já a cliente Maria, escolheria a empresa AirLines2, pois a primeira diretiva seria satisfeita onde menor número de conexões é maior do que um, sendo escolhida a empresa que oferece o menor número de conexões, a empresa AirLines2.

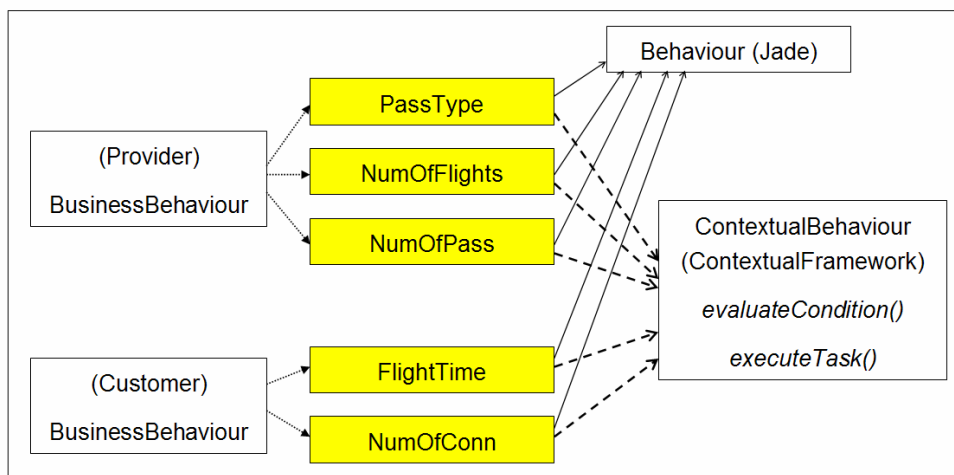


Figura 53. Comportamentos Contextuais de uma possível negociação com empresas aéreas.

A implementação da negociação de bilhetes de passagens aéreas deverá utilizar uma instância da ontologia como a descrita neste anexo e implementar as classes de Comportamentos Contextuais do framework, definindo comportamentos para se analisar informações de contexto, como o tipo e classificação de passageiros quanto a sua fidelidade, número de passageiros e ocupação da aeronave e número de vôos oferecidos para a rota solicitada. Pelo lado cliente as informações de contexto a serem analisadas para a escolha de uma empresa aérea podem ser, por exemplo, o tempo de voo e sua duração entre origem e destino e o número de conexões entre origem e destino solicitados.