

## 4 Agentes de Software

Um agente de software pode ser definido como um objeto complexo com atitude [35]. Um agente de software é governado pelo seu estado e seu comportamento. O estado do agente é descrito pelo seu conhecimento e expressado através de componentes mentais como *crenças*, *objetivos* e *planos* [36]. Já o seu comportamento é composto e governado por uma série de características comportamentais, chamadas propriedades de agência. Uma lista de propriedades de agência, bem como suas descrições, podem ser encontradas em [37]. Dentre estas, existem três que são necessárias para que uma entidade seja considerada um agente de software: autonomia, interação e adaptação.

Agentes de software também podem ser classificados quanto ao seu grau de inteligência. Agentes com uma maior capacidade de cognição são classificados como agentes *cognitivos*, ao passo que os agentes menos cognitivos são classificados como agentes *reativos* [38]. É interessante notar que agentes podem ter tanto características reativas como cognitivas – agentes puramente cognitivos ou reativos são apenas os dois extremos de um eixo imaginário onde podem ser posicionados todos os agentes de software já construídos ou ainda por construir.

A tecnologia de agentes de software é frequentemente citada na literatura como um avanço natural da tecnologia de objetos [39]. Porém, a falta de ferramental para o desenvolvimento orientado a agentes ainda é um imenso empecilho – não há uma linguagem de modelagem eficaz (UML e algumas extensões a ela já propostas [40] já provaram ser ineficientes para a modelagem multi-agente principalmente devido às características de programação concorrente que o paradigma de agentes possui), não há metodologia de desenvolvimento suficientemente testada e, por fim, não há uma linguagem orientada a agentes, o que força que qualquer software modelado com agentes tenha que ser posteriormente “traduzido” ou para o paradigma orientado a objetos ou de programação estruturada, gerando re-trabalho, ainda que parcial, de modelagem

[41]. As pesquisas nessa área continuam, pois acredita-se que seja mais fácil modelar software em termos de agentes e papéis do que em termos de objetos e funcionalidades [42].