Modelo de Actores como Modelo de Agentes

Sergio Rodríguez Calvo

Junio de 2017

Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

Abstract. *En el presente trabajo se muestran dos modelos que tienen similitudes pero que aplican en mundos distintos, estos son el modelo de agentes y el modelo de actores. El primero de ellos, el modelo de agentes, consiste en un conjunto de componentes, iguales o diferentes entre ellos, que actuan de forma racional y, en conjunto, tienen un comportamiento complejo. El segundo de ellos, el modelo de actores, aplica en el mundo virtual o de la información exclusivamente, es decir, se ejecutan en computadores y se comunican entre ellos realizando cada uno pequeñas tareas, permitiendo que los sistemas sean robustos, escalables y tolerantes a fallos.*

1. Introducción

En la naturaleza a menudo se observan comportamientos complejos y, en muchos casos, pueden llegar a considerarse ciertamente inteligentes. Al realizar un estudio en profundidad de los mismos se aprecia como el sistema se compone de seres, individuos o entes cuya operativa es muy simple. Un ejemplo de ello lo tenemos en una colonia de hormigas, el cual es un sistema complejo que resuelve problemas de envergadura, sin embargo, los individuos de una colonia de hormigas son insectos que tienen labores específicas y que aplican en todos los casos sea cuales sean las circustancias. Es un sistema que cuenta con distintos tipos de agentes por grupos, hormigas en este caso, y cada grupo tiene un enfoque o una labor diferente.

No se puede decir que una hormiga sea inteligente, o al menos, no tiene un grado elevado de la misma. En esencia, consiste en unos sentidos muy

básicos mediante los cuales percibe el entorno y de su análisis tiene lugar una acción, actuación o respuesta, racional.

Este paradigma ha sido observado por las personas y de su estudio ha surgido un paradigma o enfoque para resolver diferentes problemas en diferentes áreas. Esto se conoce como Agentes Inteligentes, aunque puede encontrarse con otro nombre como Agentes Racionales dada la dificultad que supone definir el concepto inteligencia.

Una de las virtudes que presenta un enfoque de este tipo es la posibilidad de ser masivamente paralelo. Hoy día, resultado de años de evolución y sofisticación de los sistemas de la información, se necesita construir sistemas que tengan gran capacidad, que sean escalables y elásticos en función de la carga que soporte el sistema en cada momento, y que también sean tolerantes a fallos, siendo incluso capaces de saber reponerse en caso de que se produzcan errores o fallos.

De todo ello, ha surgido una corriente que busca construir sistemas en pequeños componentes. E incluso, que esos pequeños componentes sean descritos como pequeños agentes de la información con tareas simples, que en conjunto componen un sistema que realiza tareas con muy buen rendimiento. Se le conoce como modelo de actores, y consiste en pequeños trozos de código software que ante la llegada de un mensaje realizan una tarea, y finalmente, devuelven un resultado.

Todo esto no es nuevo, si no que supone aprovechar un modelo que se creo en la decada de los 70, y que es ampliamente aplicado desde entonces en el mundo de las telecomunicaciones. Consiste en un modelo matemático que resuleve muy bien problemas de alta concurrencia.

2. Modelo de Agentes

De cara a cumplir con el objetivo de este trabajo, que consiste en plantear un modelo de actores como un modelo de agentes en el cual los actores tengan capacidad racional, previamente es necesario definir e introduccir qué es un modelo de agentes.

No existe una definición única de qué es un modelo de agentes, al igual que ocurre en todas las areas de la Inteligencia Artificial. Pero, aquí se va considerar la definición propuesta por Rusell & Norvig [3]:

> *Dada una sucesión de percepciones, un agente racional ideal debe realizar una acción que maximice la medida de éxito, a partir de*

las evidencias que obtiene de dicha sucesión y del conocimiento que posee.

A continuación, se describe con más detalle qué es un agente, así como su capacidad racional.

2.1. Qué es un agente

Un agente inteligente, o agente racional, es una entidad que percibe su entorno, realiza un procesamiento sobre dicha percepción, y actua en consecuencia en el mismo entorno.

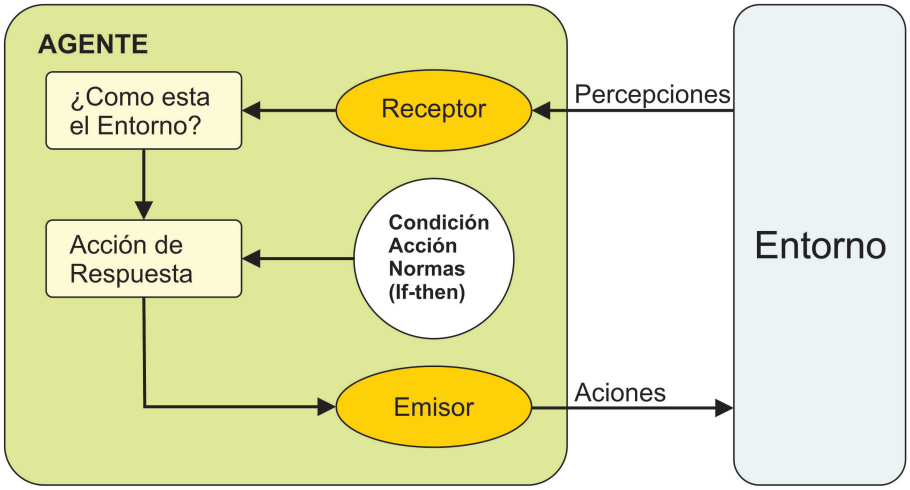

Figura 1: Esquema de un agente.

Se dice que un agente es inteligente cuando realiza una acción sobre su entorno de forma racional, entendiéndose racional como correcta o beneficiosa. En muchas ocasiones se busca una maximización con la obtención del resultado.

Un agente puede ser físico o virtual. Posteriormente se presentará un ejemplo de agente virtual, conocido como actor.

Como resumen, podemos decir que un agente debe cumplir las siguiente características:

- Percibe el entorno y aprende de él.

- Se adapta a cambios en su entorno.
- Realiza acciones correctas a partir de la información percibida de su entorno.
- Ayuda a alcanzar el objetivo común a todos los agentes que componen el sistema.

Según su comportamiento, los agentes se pueden clasificar en 6 tipos diferentes [4], estos son:

- Agentes reactivos.
- Agentes reactivos basados en modelo.
- Agentes basados en objetivos.
- Agentes basados en su utilidad.
- Agentes que aprenden.
- Agentes de consultas.

Dado el tipo de arquitecturas que se necesitan hoy día para cumplir una serie de criterios que garanticen que los sistemas son altamente concurrentes, el tipo de agente que nos interesa a partir de aquí son los agentes reactivos.

Dependiendo del diseño del sistema, que se describe en el siguiente capítulo, los actores a utilizar son los de tipo reactivo. Pueden incluso estar basados en modelo. Se necesita que el agente, en nuestro caso, y como se verá posteriormente, sea un actor reactivo porque debe responder a cambios. Generalmente, no se necesitan actores que deban conseguir cumplir objetivos propios.

También cabe destacar que los actores tendrán un comportamiento a menudo social, entendido como que un actor típicamente necesitará de otro para completar una tarea.

Pero antes de entrar en detalle con el modelo de actores, así como, el estudio de la capadidad racional de los mismos, se presenta la capacidad racional en agentes clásicos.

2.2. Capacidad racional en agentes

El objetivo es conseguir un agente que razone adecuadamente según el medio en el que se encuentre y conseguir dos cosas:

- Poder medir en qué grado logra ser racional.
- Cumplir con el Principio de Racionalidad Restringida de Herbert Simon.

En el primer punto, las medidas de rendimiento de un agente pueden variar según el agente, e incluso, no saber que hacer, engañarse a sí mismo, etc. Deben ser adecuadas para cada tipo de agente y al entorno en el que va a actuar.

El segundo punto, se debe conocer en qué consiste la racionalidad, y en este caso la misma depende de 4 factores:

- Medir el rendimiento que define el éxito.
- Conocimiento del medio en el que se encuentra el agente.
- Acciones que el agente puede ejecutar.
- Secuencia de percepciones del agente.

Un agente, por tanto, tiene que maximizar su rendimiento, para actuar de manera correcta, teniendo en cuenta las percepciones recibidas hasta ese momento.

3. Modelo de Actores

En la construcción de sistemas de información se plantea un nuevo mundo, en el cual las aplicaciones tengan una alta capacidad de respuesta, es decir, que estas sean altamente concurrentes. Esto se debe a que se necesita mantener la atención y el interés de los usuarios que acceden a ellos.

Las diferencias entre estos dos mundos, el mundo que conocemos hasta ahora y el que se plantea desde hace unos pocos años, cuenta con las siguientes diferencias [2]:

- Una única computadora frente a un cluster de computadores.

- Un único núcleo de procesamiento frente a múltiples de ellos.
- Alto coste de las memorias RAM frente a memorias RAM de bajo coste.
- Alto coste del almacenamiento físico frente a bajo coste del mismo.
- Redes lentas frente a redes de comunicaciones ultra rápidas.
- Baja concurrencia de usuarios frente a alta concurrencia.
- Conjunto pequeño de datos frente a grandes conjuntos de datos.
- Latencia en segundos frente a latencia en milisegundos.

Este modelo, el modelo de actores, fue creado en 1973 por Carl Hewitt y consiste en un modelo matemático de computación concurrente cuya primitiva universal es el actor. En este caso, un actor es una proción de código software que puede ejecutarse múltiples veces, e incluso, de forma paralela.

Esta idea puede ser llevada más allá gracias a una implementación adecuada de un toolkit que permita construir sistemas distribuidos siguiendo un modelo de actores y que a su vez sea fiel al manifiesto reactivo [1]. Pero, no sólo es una mejora, si no que se hace necesario dado el entorno en el que se desarrolla y se pretende ejecutar, un computador.

Seguir el manifiesto reactivo ayuda a optimizar los recursos, lo que unido al modelo de actores resulta en una combinación necesaria para cumplir con el objetivo de concurrencia y alta capacidad de respuesta.

Existen muchos toolkits o frameworks que implementan este modelo o la mayor parte de él. Uno de uso muy extendido es Akka, construido sobre la JVM y que permite ser utilizado tanto en Java como en Scala. Una muy buena combinación se obtiene utilizando programación funcional, la cual tiene numerosas ventajas frente a paradigmas como orientado a objeto, pero que no se comentan aquí al no estar dentro del objetivo de este documento.

3.1. Qué es un actor

Un actor, como se ha comentado previamente, es la primitiva universal de concurrencia dentro del modelo de actores. Esto es, un bloque de código por el que pasa un hilo de ejecución siempre y cuando tenga un mensaje pendiente.

Su topología consta de un buzón o cola de mensajes, y el propio actor (código). Existe un dispatcher que es el responsable de dar paso a un actor con la entrega de un mensaje del buzón. En ese caso, el actor ejecuta la acción para la que está programado. Una vez finaliza su ejecución, y dependiendo de la operativa, devolverá una respuesta, o realizará otras acciones, tales como reenviar el mensaje, crear otro actor para delegar en él, etc.

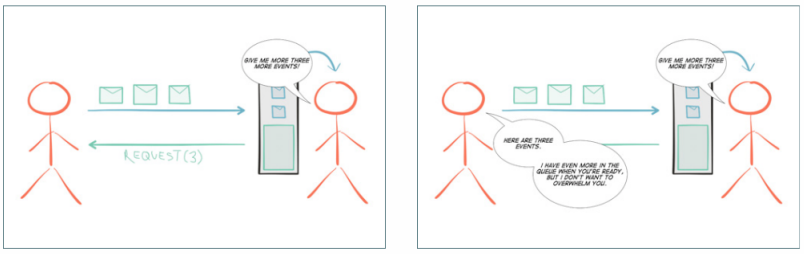

Figura 2: Comunicación entre agentes mediante técnica de *back-pressure*.

Un actor no debe guardar estados, ya que varios hilos de ejecución pasarán por el actor, y en caso de que llegue un mensaje que dependa de un mensaje anterior, el actor podría haber procesado otros mensajes previamente que han podido dejar el estado de forma que no permita alcanzar el resultado esperado, teniendo por tanto un efecto no deseado.

Es fácilmente distribuir actores, ya que al ser un paradigma orientado a mensajería, no es importante el contexto de ejecución, todo su contexto viene dado por el propio mensaje. Existe en muchos frameworks o toolkits que implementan un modelo de actores la figura del enrutador, que es la parte del toolkits que permite alojar los actores en máquinas remotas facilitando el balanceo de carga y facilitando también el aumento de la capacidad de un sistema de forma dinámica.

3.2. Sistemas distribuidos

Un sistema distribuido se define como un conjunto de computadores conectados entre sí. Esto es la solución para poder construir sistemas reactivos.

En un primer momento, los procesadores, construidos con la conocida como tecnologá del silicio, iban doblando su potencia cada 18 meses. Esto es lo que se conoce como Ley de Moore. Por ello, y gracias también al avaratamiento de la tecnología y al aumento de la capacidad de almacenamiento físico

y de memoria RAM, no era necesario una alternativa a la construción de sistemas monolíticos, es decir, sistemas desplegados en un único computador y que contara con todos los servicios.

Tampoco la sociedad utilizaba internet y los sistemas web para tantas tareas como ahora, por lo que con un único sistema monolítico y un computador con suficiente recursos (potencia y memoria) era suficiente.

A lo largo de los años se ha ido alcanzando los límites de la tecnología y nuevos paradigmas o enfoques han ido surgiendo. Por ejemplo, casi simultáneamente a los sistemas distribuidos se hizo necesario duplicar el numero de procesadores. Es decir, contar con varios núcleos de procesamiento en un mismo procesador.

Los sistemas distribuidos, por tanto, suponen llevar más allá el enfoque aplicado con el enfoque multinúcleo, y aplicarlo también a nivel de computadores aprovechando las redes de comunicaciones existentes entre ellos.

3.3. Manifiesto Reactivo

El manifiesto reactivo es un documento en el que se recoge los principios que debe tener en cuenta todo sistema que pretenda ser reactivo. Esto es, ser flexible, con bajo acoplamiento y escalables.

Estos principios son los siguientes:

- Responsividad: Los sistemas responden de forma adecuada. Se requiere de tiempos de respuesta rápidos y consistentes.
- Resilencia: Los sistemas permanecen responsivos incluso en situación de fallo. Entendiendo esto como alta disponibilidad. Esto ocurre gracias a la replicación, contención, aislamiento y la delegación.
- Elasticidad: Los sistemas permanecen responsivos incluso ante variaciones de alta carga de trabajo.
- Orientación a mensajes: Los sistemas reactivos intercambian mensaje. No mantienen estados, si no que el estado es el mensaje. Puede verse como un sistema orientado a eventos u órdenes.

En este tipo de paradigma, aplicar el manifiesto reactivo se hace necesario, ya que presenta numerosas ventajas, como por ejemplo, mejor aprovechamiento de los recursos.

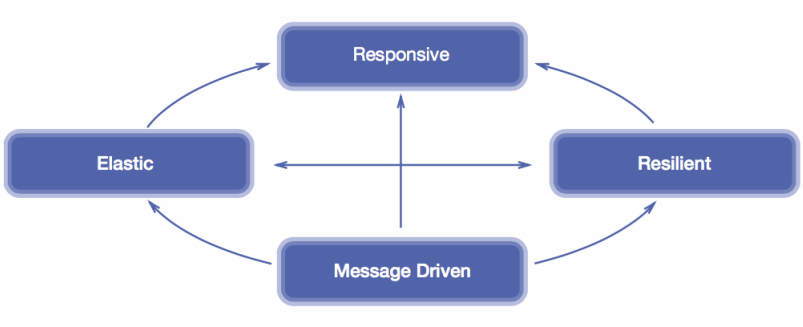

Figura 3: Relación entre los cuatro principios reactivos.

Por tanto, decir que se quiere construir un sistema siguiendo el modelo de actores, es equivalente a decir que se pretende constuir un sistema reactivo y distribuido.

3.4. Rol de los actores en sistemas distribuidos (y reactivos)

Quizás, en este punto aún no quede del todo claro por qué utilizar actores en sistemas distribuidos. Es cierto que, un sistema distribuido puede o no estar implementado utilizando actores, pero si que es de ayuda utilizarlos como manera de simplificar esta tarea.

Los actores permiten que el sistema una vez ha procesado y actuado ante un mensaje o evento, quede libre para hacer cualquier otra cosa. Esa es en esencia la mayor ventaja que proporciona usar actores. Pero tambiÃĺn facilitan mucho la tarea a la hora de mantener integridad y consistencia en los datos.

Sin entrar en detalles, con actores se permite supervisar a otros actores de forma que se pueden aplicar patrones como Sagas, que ayudan a que el sistema ante un fallo no deje el estado de los datos de una forma inconsistente.

O, por ejemplo, se puede usar para establecer un patrón de fallo rápido, conocido como Circuit Break, que ayuda a no saturar servicios ante la detección de un problema de respuesta en el mismo.

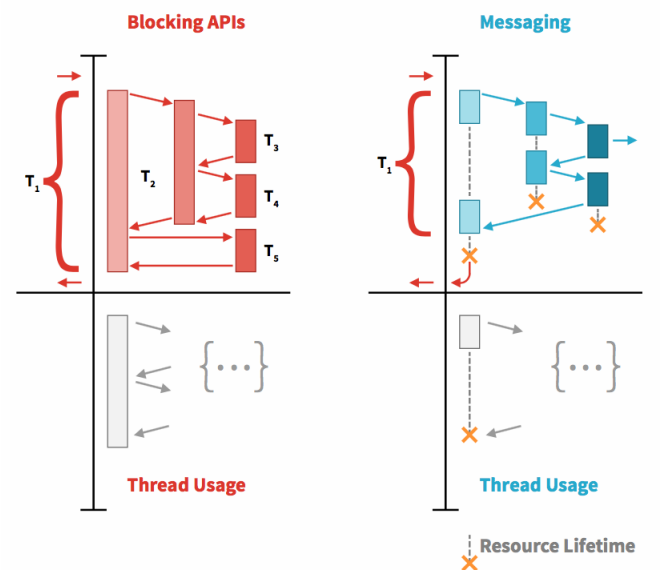

Figura 4: Ejemplo de uso de hilos eficientes en sistemas orientados a mensajería frente a sistemas tradicionales.

3.5. Capacidad racional en actores

Existen muchos tipos de actores, clasificados según su operativa. Existen actores cuya misión es simplemente hacer de enrutador y/o balanceador. Existen otros cuya misión es hacer de conector y encerrar así la lógica de comunicación con determinados servicios que usan protocolos específicos.

No se va a catalogar a esos actores como racionales, ya que aunque se obtiene de ellos una respuesta adecuada en cada momento, no aporta especial valor ya que eso mismo se podría hacer con otros sistemas que no son considerados como modelo de agente o de actores.

Sin embargo, los actores pueden utilizarse, por ejemplo, para implementar los servicios que permiten a una entidad financiera hacer transferencias entre clientes. En este caso, se dan dos operaciones desde un punto de vista muy básico, que son:

- Retirar el importe del cliente que realiza la transferencia previa comprobación de que cuenta con el saldo suficiente.
- Agregar dicho importe en el cliente que recibe la transferencia.

En un sistema que implemente dichos requisitos, y desarrollado con actores, deben existir los siguientes actores que se encarguen de realizar la tarea de permitir que se hagan las transferencias sin que se pierda dinero en el proceso. Tarea que por otro lado no es simple.

Se necesitan, por ejemplo, de acotres que deleguen en otros tareas más atómicas y se encarguen de supervisar que todos han ido correctamente. Tipicamente, estas tareas se desarrollan en paralelo, y si una tarea B que está asociada a otra tarea A falla, pero la tarea A no, hay que compensar la tarea A.

Otro caso más complejo, se da en aquellas entidades que cuentan con un sistema legado que cuenta con lógica en la propia base de datos. Este caso tiene la dificultad de que ante un fallo en el patrón que se acaba de describir provoca que un procedimiento de alerta en la base de datos salte inmediatamente, incluso aunque los actores se encarguen del problema intenando compensar.

En este caso, se puede realizar implementaciones más complejas que consisten en grupos de actores permanentemente activos y divididos en regiones, de manera que se mantiene a cada actor vivo todo el tiempo para evitar que este tipo de problemas ocurra.

En todos estos casos, los actores ejecutan tareas complejas y dan la respuesta adecuada, es decir, la respuesta racional en todos los casos, los de éxitos, y también los de fallo.

4. Ventajas e Inconvenientes en el uso de actores

Todas las ventajas del uso de actores han sido descritas a lo largo de este documento. Se enumeran las más importantes aquí:

- Uso eficiente de los recursos en los servidores.
- Facilidad de replicación de componentes, servidores y actores.
- Mayor capacidad y versatilidad de evolución de una plataforma al ser intercambiables los módulos.
- Capacidad de soportar más concurrencia, e incluso, ser elástico ante cambios.

Pero, también se presentan inconvenientes en su utilización:

- Necesidad de paradigma funcional en el desarrollo de los actores para ser más testable.
- Imposibilidad de garantía de transacción y necesidad de nuevas técnicas para garantizar la integridad de los datos.
- Mayor interoperabilidad e integración.

5. Conclusión

Este nuevo paradigma está siendo cada vez más usado en la industria tecnológica, o incluso, en aquellas industrias que no tienen origen tecnológico pero que se ven obligados a pivotar hacia dicho sector. Ofrece una solución al mayor problema de hoy día en los sistemas expuestos en Internet, dar servicio a cada vez más usuarios.

En parte, no se ha hecho más que reutilizar técnicas o paradigmas utilizados en otros áreas o industrias. Como en este caso, que se presentan similitudes entre un modelo de agentes y un modelo de actores.

En el futuro, este parece ser una de las pocas salidas donde poder avanzar y con una alta probabilidad evolucionar aún más perfeccionando modelos y técnicas ampliamente conocidas. Incluso, extendiendo su uso a sectores o campos en los que no se hace aún uso de agentes o actores.

Referencias

[1] Roland Kuhn Martin Thompson Jonas BonÃľr Dave Farley. *Manifiesto Reactivo*. `http://www.reactivemanifesto.org/es`.

[2] Hugh McKee. *Designing Reactive Systems: The Role of Actors in Distributed Architecture*. 1995.

[3] Stuart Russell y Peter Norvig. *Artificial Intelligence: A Modern Approach*. 1995.

[4] Wikipedia. *Tipos de agentes*. `https://es.wikipedia.org/wiki/Agente_inteligente_(inteligencia_artificial)`.