Lámpsakos | N°. 14 | pp. 25-32 | julio-diciembre | 2015 | ISSN: 2145-4086 | Medellín–Colombia

Transformación del Q-Learning para el Aprendizaje en Agentes JADE

Q-Learning Transformation for Training on JADE Agents

Nayma Cepero-Pérez, Ing.
Instituto Superior Politécnico José Antonio Echeverría (Cujae)
La Habana, Cuba
ncepero@ceis.cujae.edu.

Mailyn Moreno-Espino, PhD.
Instituto Superior Politécnico José Antonio Echeverría (Cujae)
La Habana, Cuba
my@ceis.cujae.edu.

(Recibido el 10-04-2015. Aprobado el 10-06-2015)

Citación de artículo, estilo IEEE:
N. Cepero-Pérez, M. Moreno-Espino, "Transformación del Q-Learning para el Aprendizaje en Agentes JADE", Lámpsakos, N°. 14, pp. 25-32, 2015

Resumen. El aumento de la interacción entre los sistemas informáticos ha modificado la forma tradicional de analizarlos y desarrollarlos. La necesidad de la interacción entre los componentes del sistema es cada vez más importante para poder resolver tareas conjuntas, que de forma individual serían muy costosas o incluso imposibles de desarrollar. Los sistemas multi-agente ofrecen una arquitectura interesante y completa para ejecutar tareas distribuidas que cooperan entre sí. La creación de un sistema multi-agente o un agente requiere de gran esfuerzo, por lo que se han adoptado métodos como los patrones de implementación. El patrón Proactive Obsever_ JADE permite crear los agentes e incluirle en cada uno comportamientos dotados de inteligencia que pueden evolucionar utilizando técnicas de aprendizaje automático. El aprendizaje por refuerzo es una técnica del aprendizaje automático que permite a los agentes aprender a través de interacciones de prueba y error, en un ambiente dinámico. El aprendizaje por refuerzo en sistemas multi-agente ofrece nuevos retos derivados de la distribución del aprendizaje, como pueden ser la necesidad de la coordinación entre agentes o la distribución del conocimiento, que deben ser analizados y tratados.

Palabras clave: agentes, aprendizaje reforzado, JADE, patrones de implementación.

Abstract. Increased interaction between computer systems has modified the traditional way to analyze and develop them. The need for interaction between the system components is increasingly important to solve joint tasks, which individually would be very expensive or even impossible to develop once. Multi-agent systems offer an interesting and complete distributed architecture to execute tasks cooperate. The creation of a multi-agent system or an agent requires great effort so methods have been adopted as the deployment patterns. The pattern creates Proactive Obsever_JADE agents and include in each endowed with intelligence behaviors can evolve using machine learning techniques. The reinforcement learning is a machine learning technique that allows agents to learn through trial and error interactions in a dynamic environment. Reinforcement learning in multi-agent systems offers new challenges arising from the distribution of learning, such as the need for coordination between agents or distribution of knowledge, which should be analyzed and treated.

Keywords: agents, reinforced learning, JADE, patterns implementation.

1. INTRODUCCIÓN

El avance alcanzado en la tecnología y las comunicaciones en el mundo actual, han propiciado que se dediquen grandes esfuerzos orientados hacia la simulación del pensamiento humano. En este empeño una de las ciencias que más se destaca es la Inteligencia Artificial (IA).

De los estudios desarrollados por la IA y como parte de la evolución de la ingeniería de software surgió un novedoso campo de investigación y desarrollo: los agentes. Los agentes han sido el tema de intensas investigaciones durante años, constituyendo uno de los enfoques computacionales más prometedores para abordar cuestiones que requieren un modelado flexible, abstracto y un razonamiento elevado [1].

Cuando se comenzó a trabajar el tema de agentes, la meta fundamental fue crear lenguajes y protocolos para que los agentes inteligentes compartieran conocimiento. En este ámbito era común que se trataran a los agente como entes con características que usualmente se aplican en los seres humanos, ya que se pretende que los agentes imiten su comportamiento [2].

El paradigma de agente ha encontrado aceptación en diferentes áreas como las tecnologías de la información, incluyendo las Redes de Computadoras, Ingeniería de Software, Programación orientada a objetos, Inteligencia Artificial, entre otros. Aunque no existe un concepto en cuanto a qué es un agente, muchos autores han hecho sus definiciones. Un intento de ello y con el objetivo de unificar los esfuerzos para el desarrollo de las tecnologías para agentes inteligentes puede encontrarse en la Fundación para Agentes Inteligentes Físicos (FIPA).

FIPA es una organización internacional que se dedica a promover la industria de los agentes inteligentes desarrollando abiertamente especificaciones que soporten la interoperabilidad entre los agentes [3].

Los agentes con su habilidad social forman sistemas con más de un agente, es decir un Sistema Multi-Agente (SMA) que son producto de la necesidad de desarrollar aplicaciones complejas, que a un agente individual le tomaría demasiado tiempo resolver [4].

Basándose en las especificaciones de FIPA y con la meta de simplificar el desarrollo de un SMA surge Java Agents DEvelopment Framework (JADE), un marco de trabajo estándar personalizado para el desarrollo de aplicaciones multi-agente distribuidas [5]. En JADE un agente inteligente puede ser definido como una entidad computacional capaz de solucionar problemas de operaciones en ambientes dinámicos y abiertos [6].

La creación de un SMA o incluso de un agente, requiere de grandes esfuerzos, por lo que se han adoptado métodos como los patrones para facilitar el trabajo con JADE, la creación de agentes e incluir comportamientos proactivos.

Con el patrón Proactive Observer_JADE se facilita el trabajo con la plataforma de desarrollo JADE [7], se incorporan comportamientos proactivos; además se permite al usuario realizar cambios en el comportamiento de los agentes de forma dinámica, lo que es un aspecto primordial en el desarrollo de un SMA [8].

Los agentes creados con el patrón antes mencionado no tienen un comportamiento dotado de una inteligencia que pueda evolucionar utilizando técnicas de aprendizaje automático.

Teniendo en cuenta los planteamientos anteriores los agentes creados con el patrón Proactive Observer_JADE no aprenden para cambiar su comportamiento en función del aprendizaje. Por este motivo, es necesario agregar a estos agentes operaciones que les permitan adquirir conocimiento del medio donde trabajan, para revertirlo en un mejor funcionamiento de las actividades que realizan los agentes [9].

En el presente trabajo se aborda el problema del aprendizaje por refuerzo en los agentes que se implementan sobre la plataforma de agentes JADE. El objetivo del trabajo es transformar un algoritmo de aprendizaje reforzado que se adecue con el patrón Proactive Observer_JADE.

2. DESCRIPCIÓN DE LOS AGENTES

La historia de la computación según Wooldrige [1] ha estado marcada por cinco tendencias fundamentales: la ubicuidad, la interconexión, la delegación, la inteligencia y la orientación a humano.

Estas tendencias han permitido el surgimiento del paradigma de agentes. Los agentes intentan simular el comportamiento humano, teniendo como objetivos

interactuar, cooperar, coordinar y negociar con los demás agentes, de manera similar a como lo hacen los humanos.

En diferentes momentos se han comparado a los agentes con los objetos, pero los agentes no son objetos con otro nombre, debido a que un agente es un sistema racional que toma decisiones propias, que puede actuar de forma proactiva y no solo reactiva. Un agente se desempeña de forma diferente, dependiendo de la situación en que se encuentre, porque tiene una intención propia para cumplir una meta determinada de diseño [10, 11].

Franklin [12] brinda una definición de agentes en la que menciona que entre las propiedades principales de estos se encuentran: la autonomía, estar orientados a metas, ser colaborativos, flexibles, autoiniciables, con continuidad temporal, comunicativos, adaptativos y móviles. Agrega que un agente autónomo es un sistema situado en un ambiente que percibe y actúa sobre él en el tiempo, según su agenda propia, y de esta manera produce efectos que él mismo podrá sentir en el futuro [12].

Russell y Norvig [13] tienen una visión más flexible de los agentes, como una herramienta para analizar sistemas y no como una característica abstracta que divida al mundo en agentes y no-agentes.

De manera general la autora de este trabajo considera que las anteriores definiciones son válidas, con distintos niveles de amplitud y reflejando aspectos diferentes, sin entrar en contradicción una definición con la otra. Por tanto, se puede decir que las características fundamentales que poseen los agentes son: autonomía, habilidad social, reactividad y proactividad. Cuyas características se pueden resumir de la siguiente manera [1]:

- *Autonomía:* un agente opera sin la intervención directa de humanos y debe tener una cierta clase de control sobre sus acciones y su estado interno.

- *Habilidad social:* los agentes tienen la capacidad de interactuar con otros agentes mediante algún mecanismo de comunicación. Esto le permite lograr objetivos que por sí solos no puede lograr.

- *Reactividad:* los agentes perciben su entorno y responden en un tiempo razonable a los cambios que ocurren en él. El agente puede estar en estado pasivo la mayor parte del tiempo y despertar al momento que detecte ciertos cambios.

- *Proactividad:* el comportamiento proactivo es tomar iniciativa para mejorar las circunstancias actuales o crear nuevas [14].

Según [15] la proactividad tiene tres atributos fundamentales: comienza por sí misma, está orientada al cambio y se enfoca en el futuro. La proactividad se divide en tres niveles: individual, de equipo y organizacional [16].

2.1. El Marco de Desarrollo JADE

Se han desarrollado muchas plataformas con la finalidad de permitir la interconexión y ejecución de los agentes de un sistema multi-agente. Algunas siguen el estándar FIPA y otras se basan en arquitecturas de agente propias [5].

Java Agents DEvelopment Framework es un entorno de trabajo estándar personalizado para el desarrollo de aplicaciones multi-agente distribuidas. Su desarrollo está basado en las especificaciones de FIPA y tiene como meta simplificar el desarrollo de SMA mientras que asegura el cumplimiento de los estándares a través de un sistema de servicios y agentes comprensibles al programador [5]. Trabaja en aquellos aspectos que no están relacionados con el funcionamiento interno de los agentes y que son independientes de la aplicación, como el transporte de mensajes (su codificación y análisis) y el ciclo de vida de los agentes [17].

JADE puede ser considerado como un entorno de trabajo que implementa una plataforma de agentes y apoya el desarrollo de un SMA. La plataforma de agentes JADE se enfoca en el funcionamiento de un sistema de agentes distribuidos con el lenguaje Java. En particular, su arquitectura de comunicación trata de ofrecer mensajería flexible y eficiente, eligiendo el mejor transporte disponible de forma transparente y el aprovechamiento del estado de la técnica de la tecnología de objetos distribuidos, embebidos en un ambiente de tiempo de ejecución Java [5].

2.2. Patrón Proactive Observer_JADE

Este patrón utiliza los principios del patrón de comportamiento Observer [18], maneja como base el protocolo de interacción Subscribe de FIPA [19].

Para lograr una implementación genérica del patrón con agentes, se deben crear fundamentalmente dos agentes: Observer y Observable. Los agentes que cumplirán con estos roles tendrán la capacidad de comunicarse y actuar autónomamente, pudiendo hasta cambiar su comportamiento en dependencia de las situaciones a las que se enfrenten [7].

El agente con el rol de Observable debe tener una lista interna de los agentes Observer para poder alertarlos de los cambios que detecta. Además debe esperar los mensajes de subscripción de los agentes Observer para poder sumarlos a la lista mencionada y enviar la respuesta de la subscripción. Por último debe tener la capacidad de enviar un mensaje a los Observers con los datos necesarios del cambio encontrado.

El agente Observer tiene que conocer los Observables existentes en el entorno para decidir a cuáles subscribirse. Además debe actualizar su estado cuando le llegue un mensaje con los datos que describen el cambio ocurrido y actuar en consecuencia.

3. APRENDIZAJE AUTOMÁTICO

El aprendizaje automático se basa en diseñar algoritmos que permitan a una computadora aprender, tratando de encontrar regularidades estadísticas u otros patrones en los datos. Por tanto, muchos algoritmos pueden parecerse a un un humano cuando aborda una tarea de aprendizaje [20].

El aprendizaje automático es la habilidad de las computadoras para entender los datos, gestionar resultados e inferir conocimiento de información incierta, es la fuerza detrás de muchas revoluciones recientes en la computación. Los filtros de correo spam, asistentes personales en los teléfonos inteligentes y vehículos que se auto-manejan están basados en los avances en las investigaciones del aprendizaje automático. Desafortunadamente, aunque la demanda de estas capacidades se ha ido acelerando, cada nueva aplicación necesita de un gran esfuerzo. Hasta un equipo de expertos especialmente entrenado en aprendizaje automático realizan progresos lentos por la falta de herramientas para construir estos sistemas [9].

Los sistemas de aprendizaje automático aprenden a ejecutar acciones a partir de los datos. Si se tiene una aplicación que pensamos puede ser apoyada con aprendizaje automático, el primer problema que enfrentamos es la selección del algoritmo que se va a utilizar. La clave es no perderse entre todo el espacio, sino percatarse de que existe una combinación de solo tres componentes [21]:

Aprendizaje = representación + evaluación + optimización

- Representación: un clasificador debe estar representado en algún lenguaje formal que la computadora pueda manejar. Por el contrario, escoger una representación para un aprendiz es equivalente a escoger un grupo de clasificadores que sean posibles de aprender.

- Evaluación: una función de evaluación es necesaria para distinguir buenos clasificadores de los malos. La función de evaluación usada internamente por el algoritmo puede diferir del externo que queremos que el clasificador optimice.

- Optimización: finalmente necesitamos un método para buscar entre los clasificadores en el lenguaje para el de mayor score. La selección de la técnica de optimización es clave para la eficiencia del aprendiz, y también ayuda a determinar el clasificador producido si la función de evaluación tiene más de un mínimo.

Las técnicas de aprendizaje automático (Machine Learning (ML, por sus siglas en inglés) se definen como la habilidad computacional de una máquina de mejorar su rendimiento basándose en resultados anteriores [22].

Algunas tareas de aprendizaje automático están orientadas hacia el diseño de robots móviles que aprenden a navegar según su propia experiencia, minar datos históricos en registros médicos para aprender cuales pacientes futuros responderán mejor a ciertos tratamientos, como construir motores de búsqueda que automáticamente se personalice hacia los intereses de los usuarios, la clasificación de documentos, entre otras [21].

3.1. Aprendizaje reforzado

El Aprendizaje Reforzado es el problema al que se enfrenta un agente que aprende a través de interacciones de prueba y error, en un ambiente dinámico. El aprendizaje por refuerzo es una técnica de aprendizaje automático que se aplica principalmente en entornos activos y dinámicos. En aprendizaje por refuerzo, el agente percibe el entorno que tiene a su alrededor y ejecuta acciones que lo modifican en un proceso iterativo de prueba y error. Cada vez que el agente ejecuta una acción sobre el entorno, éste le informa de su nuevo estado y de un refuerzo, que es alguna medida de la distancia al objetivo [23].

En el modelo del Aprendizaje Reforzado que se muestra en la Fig. 1 [24] se observa cómo un agente se conecta a su entorno a través de la percepción y la acción. En cada paso de interacción el agente recibe como entrada "i", alguna indicación del estado actual del entorno "s", el agente entonces selecciona una acción "a" generada como un resultado. La acción cambia el estado del entorno y el valor de esta transición de estado se le comunica al agente a través de una señal reforzada "r". El comportamiento del agente B, debe escoger acciones que tienden a incrementar la sumatoria de los valores de la señal reforzada [23]. Puede aprender comportarse con el tiempo por pruebas y errores sistemáticos, guiados por una amplia variedad de algoritmos.

3.2. Algoritmo Q-learning.

El algoritmo de Q-Learning [23] es un método libre del modelo que aprende a partir de conocimiento adquirido del entorno en cada acción ejecutada en cada estado, es decir, a partir de la experiencia generada durante la exploración del entorno.

Esta experiencia se representa mediante tuplas de experiencia que contienen:

- s, el estado actual en el que se encontraba el agente al ejecutar la acción.
- r, la acción ejecutada por el agente.
- s´, el estado al que se ha llegado ejecutando la acción.
- r´, el refuerzo obtenido por ello.

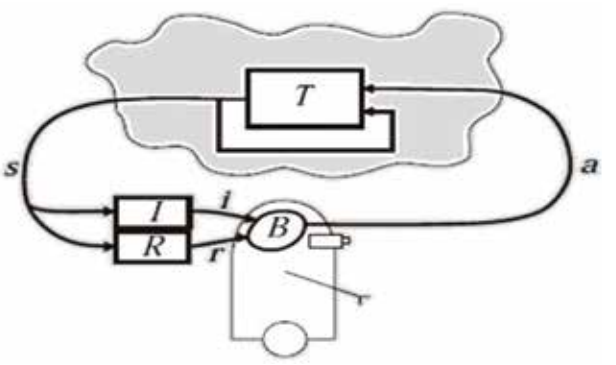

Fig. 1. Representación esquemática del aprendizaje reforzado

Dicha experiencia se encuentra en la función que mapea un estado y una acción ejecutada con un valor de la calidad de dicha acción en dicho estado.

El algoritmo de Q-Learning se basa en estimar la función de valor-acción, mediante la experiencia acumulada en los distintos episodios de aprendizaje. Habitualmente la representación de la funciones, se hace en una tabla de valores. Sin embargo, es posible aproximarla mediante técnicas de aproximación de funciones como redes de neuronas o árboles de decisión. El hecho de que la función dependa de dos parámetros, estado y acción, da lugar a dos alternativas posibles a la hora de aproximar su valor: en la primera alternativa, se usa un aproximado que toma como entrada tanto el estado, como la acción, obteniendo un valor. En la segunda alternativa, si el conjunto de acciones es reducido, se puede usar un aproximado para cada posible acción, obteniendo tantos aproximados como acciones sean posibles.

4. ALGORITMO MODIFICADO DE AJUSTE CON PATRÓN PROACTIVE OBSERVER_ JADE

Como parte de la investigación desarrollada para este trabajo se estudiaron los diferentes algoritmos que se han diseñado para implementar el Aprendizaje Reforzado. Anteriormente se realiza una descripción del algoritmo Q-learning, el mismo es un método libre de modelo (lo que quiere decir que no necesita un modelo previo para aprender) porque aprende a partir de conocimiento adquirido del entorno en cada acción ejecutada en cada estado, es decir, a partir de la experiencia generada durante la exploración del entorno. El Q-learning es el algoritmo que se utiliza para implementar un SMA para entornos dinámicos, más dinámicos que la situación problemática des-

crita previamente en la Introducción de este trabajo [9, 21, 24-26]. Después del estudio realizado se decidió que partiendo del algoritmo Q-learning se le aplicarán modificaciones que permitan adaptarlo al problema que se está tratando. Las modificaciones desarrolladas respetan la estructura básica del Q-learning, pues también es un método libre de modelo que aprende de la interacción del usuario con el sistema y no de un modelo definido anteriormente. Para las adaptaciones del algoritmo, se toma como base que el aprendizaje se logra mediante la interacción del sistema con el usuario. A continuación se presenta la descripción del algoritmo:

Se construye una tabla llamada: "Aprendizaje" que contiene una relación directa entre un término Keyword y la relevancia.

El Keyword es extraído del documento, específicamente del título, por ejemplo "Java, SDK" contiene 2 Keyword, y la relevancia es un valor numérico individual para cada palabra.

La tabla "Aprendizaje" al inicio está vacía, pues como se ha mencionado se va construyendo durante la interacción del usuario con el componente.

La relevancia es un valor que depende del refuerzo, que a su vez depende de las acciones realizadas por el usuario.

Para una mayor comprensión de lo anteriormente expuesto, supongamos la siguiente situación: cuando se inicia el componente, en la clase interfaz "Selection_UI", al usuario se le muestra toda la información que es nueva en la carpeta que se está vigilando; aquí el usuario puede guardar, ignorar o pedir sugerencias.

Cuando el usuario decide guardar información esta se almacena en la tabla "Aprendizaje", donde se separan las palabras y se les asigna un refuerzo a cada palabra de manera individual y posteriormente se le asigna refuerzo al título completo. El refuerzo ante esta acción es 1, es válido aclarar que este valor se puede variar según los intereses del usuario, solo se tiene que cambiar una función que asigne el refuerzo de otra manera.

En el caso de que el usuario ignore la información que se le está mostrando, se cierra la interfaz y el agente no aprende nada nuevo.

Cuando el usuario pide sugerencias se levanta la interfaz "Suggestion_UI", el algoritmo se conecta con la tabla "Aprendizaje" y según los intereses del usuario que se hayan ido almacenando, se le muestra al usuario las sugerencias de la información, ordenada de mayor a menor según el orden de relevancia de los documentos. En esta interfaz el usuario podrá guardar o cancelar las sugerencias, si guarda la información que se le está sugiriendo, el valor del refuerzo aumenta en 0.5 porque además de realizar bien la acción recibe un premio extra por haberla sugerido correctamente y por ser interés del usuario.

En el caso de que el usuario cancele las sugerencias que se le están mostrando, se cierra la interfaz y el agente no aprende nada nuevo. Y no se realiza ninguna otra acción hasta que no llegue un grupo nuevo de información.

4.1. Seudocódigo del Algoritmo:

- Informar que el agente se encuentra listo para ejecutar el episodio;

- Para cada documento D new;

Mostrar información al usuario;

» Almacenar la información de interés del usuario;

- Separar las palabras del título de D;
- Asignar el refuerzo;
- Calcular la relevancia de las palabras;
- Calcular la relevancia para el título;
- Definir el Umbral (para mostrar sugerencias);
- Almacenar información en la Base de Datos;

» Informar que el agente terminó el entrenamiento

- Mientras se copie un D nuevo repetir el ciclo de Mostrar información al usuario;

Mostrar sugerencias al usuario;

- Almacenar la información de interés del usuario;
- Separar las palabras del título de D;
- Asignar el refuerzo;

- Calcular la relevancia de las palabras;
- Calcular la relevancia para el título;
- Almacenar información en la Base de Datos;

- Repetir el ciclo de Mostrar información al usuario siempre que haya un D nuevo;

- Terminar cuando no haya más información nueva.

4.2. El algoritmo en el agente Observer

Después de realizadas las modificaciones al Q-learning original para transformarlo en un algoritmo que se pudiera adecuar a la situación planteada, es momento entonces de incorporarlo al agente Observer. La manera en la que se propone hacerlo es a través de un comportamiento, un comportamiento no es más que una clase Java que le dice al agente como actuar dentro del SMA. Este comportamiento puede ser modificado en tiempo de ejecución lo que resulta muy importante en el desarrollo de un SMA. En la Fig. 2 se muestra en un diagrama la idea general del funcionamiento antes explicado, existe un paquete que contiene al patrón, otro con el agente o agentes Observer y otro con el comportamiento.

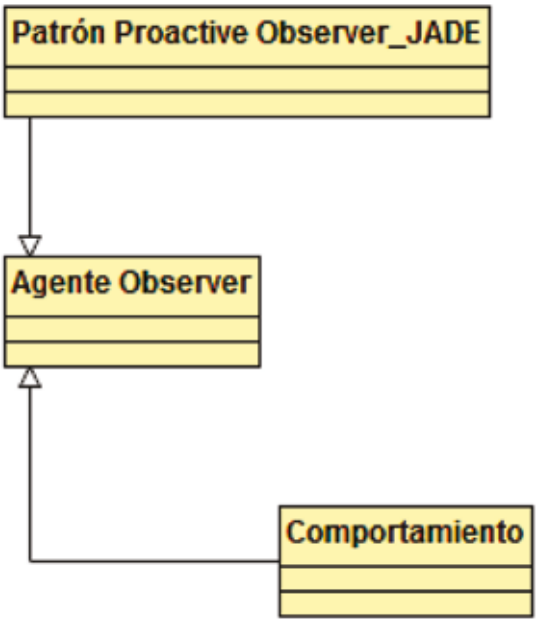

Fig. 2. Comportamiento incorporado en el Agente Observer

5. TRABAJOS FUTUROS

Como trabajo futuro se propone realizar diferentes casos de estudio que permitan validar mediante la experimentación la funcionalidad de la propuesta desarrollada. Para evaluar la mejora del funcionamiento de los agentes Obsever dentro del SMA, permitiendo medir la evolución del aprendizaje de los agentes.

6. CONCLUSIONES

De manera general se puede concluir que el aprendizaje reforzado es una vía efectiva para incluir en los agentes comportamientos dotados de inteligencia.

Se transformó el algoritmo Q-learning para adecuarlo a la situación problemática planteada, logrando que el algoritmo ajustado permita a los agentes aprender. También se plantea la manera de incluir el algoritmo dentro de los agentes como comportamiento.

REFERENCIAS

[1] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd. John Wiley & Son, 2009. ISBN: 9780470519460.

[2] FIPA, *FIPA Agent Management Specification.* Foundation for Intelligent Physical Agents, 2003. URL: http://www.fipa.org/specs/fipa00023/

[3] FIPA, *FIPA Communicative Act Library Specification*. Foundation For Intelligent Physical Agents, 2003. URL: http://www.fipa.org/specs/fipa00037/SC00037J.html

[4] J. Ferber. “Multi-agent systems: an introduction to distributed artificial intelligence”. Addison-Wesley, 1999. ISBN 0-201-36048-9

[5] F. L. Bellifemine, G. Caire & D. Greenwood. “Developing Multi-Agent Systems with JADE”. 1st. Chichester: John Wiley & Sons, Ltd, 2007. ISBN: 978-0-470-05747-6. URL: http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470057475.html

[6] F. Bellifemine, F. Bergenti, G. Caire & A. Poggi, "Jade-A Java Agent Development Framework", in Multi-Agent Programming, R. Bordini, et al. (Ed.). Springer US., 2005, pp. 125-147. URL: http://link.springer.com/chapter/10.1007%2F0-387-26350-0_5

[7] M. Moreno, A. Carrasco, A. Rosete & M. D. Delgado, "Patrones de Implementación para Incluir Comportamientos Proactivos," *Polibits*, Vol. January-June 2013, no. 47, pp. 73-87.

[8] B. Henderson-Sellers, "From Object-Oriented to Agent-Oriented Software Engineering Methodologies", in *Software Engineering for Multi-Agent Systems III*, R. Choren, et al. (Ed.). Springer Berlin Heidelberg, 2005, pp. 1-18. URL: http://link.springer.com/chapter/10.1007%2F978-3-540-31846-0_1

[9] *DARPA Envisions the Future of Machine Learning*, 2013. URL: http://www.darpa.mil/news-events/2013-03-19a

[10] T. Ishida, L. Gasser & M. Yokoo. "Organization self-design of distributed production systems", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 4, no. 2, pp. 123-134, 1992. DOI: 10.1109/69.134249

[11] OMG, *Agent Platform Special Interest Group: Agent Technology – Green Paper*, 2000. URL: http://www.objs.com/agent/agents_Green_Paper_v100.doc

[12] S. Franklin &A. Graesser, "Is It an agent, or just a program?: A taxonomy for autonomous agents", in *Intelligent Agents III Agent Theories, Architectures, and Languages*, J. Müller, M. Wooldridge, and N. Jennings (Ed.). Springer Berlin Heidelberg, 1997, pp. 21-35. URL: http://link.springer.com/chapter/10.1007%2FBFb0013570

[13] S. Russell & P. Norvig. "Artificial Intelligence: A Modern Approach". 3rd, illustrated. Prentice Hall, 2010. ISBN: 978-0136042594

[14] J. M. Crant. "Proactive Behavior in Organizations," *Journal of Management*, Vol. 26, no. 3, pp. 435-462, 2000. DOI: 10.1177/014920630002600304

[15] A. M. Grant & S. J. Ashford. "The dynamics of proactivity at work," *Research in Organizational Behavior*, Vol. 28, pp. 3-34, 2008. DOI:10.1016/j.riob.2008.04.002

[16] S. K. Parker, U. K. Bindl & K. Strauss. "Making Things Happen: A Model of Proactive Motivation," Journal of Management, Vol. 36, no. 4, pp. 827-856, 2010. DOI: 10.1177/0149206310363732

[17] F. Bellifemine, A. Poggi & G. Rimassa, *JADE – A FIPA-compliant agent framework*, 1999. URL: http://sharon.cselt.it/projects/jade/papers/PAAM.pdf

[18] E. Gamma, R. Helm, R. Johnson & J. Vlissides, *Design Patterns: Elements of Reusable Object-oriented Software*. Pearson Education, 2004. ISBN: 978-0201634983

[19] FIPA, *FIPA Subscribe Interaction Protocol Specification*, Foundation For Intelligent Physical Agents, 2003. URL: http://www.fipa.org/specs/fipa00095/PC00095A.pdf

[20] R. Barr & C. Rovee-Collier. "Encyclopedia of the Sciences of Learning". Estados Unidos: Springer, 2012. URL: http://www.springer.com/us/book/9781441914279

[21] P. Domingos. "A few useful things to know about machine learning". *Commun. ACM*, Vol. 55, no. 10, pp. 78-87, 2012. DOI: 10.1145/2347736.2347755

[22] A. Shhab, G. Guo & D. Neagu, *A Study on Applications of Machine Learning Techniques in Data Mining*, 2005. URL: http://pythia.inf.brad.ac.uk/paper/BNCODWorkshop.pdf

[23] L. P. Kaelbling, M. L. Littman & A. W. Moore. "Reinforcement Learning: A Survey". *Artificial Intelligence Research*, Vol. 4, no. pp. 237-285, 1996. DOI: 10.1.1.134.2462

[24] R. S. Sutton & A. G. Barto. "Reinforcement Learning: An Introduction". Cambridge, Massachusetts: The IMT Press, 1998. ISBN: 978-0262193986

[25] INTECHOPEN, "New Advances in Machine Learning", InTech, 374p, 2010. ISBN 978-953-307-034-6

[26] C. Germain-Renaud, A. Cady, P. Gauron, M. Jouvin, Ch. Loomis, et al., en *The Grid Observatory,* IEEE Computer Society Press. *IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*. Newport Beach, United States, May 2011.