

Modelado Basado en Agentes: un Enfoque desde la Ingeniería de Sistemas

María Pereda, Jesús M. Zamarreño*

Departamento de Ingeniería de Sistemas y Automática, EII, Universidad de Valladolid, C/ Doctor Mergelina s/n, 47011, Valladolid, España

Resumen

El modelado basado en agentes (ABM, Agent Based Modeling) es una técnica de modelado que está siendo explotada con gran éxito en áreas como la ecología, ciencias sociales, economía, etc. Sin embargo, su uso como técnica de modelado en el campo de la Automática es más bien testimonial. En este artículo mostramos cómo se puede abordar el modelado basado en agentes desde el punto de vista de la Ingeniería de Sistemas y Automática y las particularidades que tiene como herramienta de modelado. Asimismo, proponemos una descripción matemática de los modelos basados en agentes que ilustramos con un par de ejemplos *Copyright © 2015 CEA. Publicado por Elsevier España, S.L.U. Todos los derechos reservados.*

Palabras Clave:

Agentes, Modelado dinámico, Ingeniería de sistemas, Espacio de estados, Representaciones conceptuales.

1. Introducción

El modelado basado en agentes (MBA) es una técnica de modelado de sistemas complejos cuyo uso se ha incrementado notablemente en las últimas dos décadas como herramienta de modelado en diferentes campos de aplicación, desde las ciencias sociales, economía, ecología, etc. Esta técnica se basa en modelar los sistemas desde el punto de vista de los elementos que los componen; los agentes, y las relaciones entre ellos. Una de las principales ventajas de esta técnica de modelado es que existe una correspondencia directa entre los elementos en el sistema y los elementos en el modelo, y las relaciones entre ellos en el sistema y las relaciones entre elementos en el modelo (Galán et al., 2009). Se trata pues de un proceso de modelado de abajo hacia arriba, en el que las propiedades macroscópicas de los sistemas son consecuencia (*emergen*) de las relaciones entre los agentes que forman el sistema, de forma opuesta a las técnicas de modelado tradicionales basadas en parámetros concentrados. Esta característica permite estudiar los sistemas desde el punto de vista de sus elementos constitutivos.

El modelado basado en agentes se basa, como su propio nombre indica, en el modelado del sistema a partir de agentes. No existe una definición universalmente aceptada de agente. Sin embargo, tomando como base (Macal and North, 2006), se puede concluir que un agente sería un elemento individual e identificable, con un conjunto de características o atributos, y

reglas que gobiernan su comportamiento y su capacidad de decisión. Este agente estaría situado en un entorno con el que interactúa, además de con otros agentes. Un término muy utilizado también es el de sistema multi-agente (MAS, Multi-Agent System), que, según (Torsun, 1995), consiste en un conjunto de agentes autónomos, heterogéneos e independientes que viven en un entorno, interactúan con él y entre ellos, con sus propias metas, capacidades y conocimiento. La aplicación de los sistemas multi-agente puede hacerse atendiendo a diversos puntos de vista (Luck et al., 2005), siendo los más interesantes, desde el punto de vista de la Automática, el estudio de los agentes como origen de tecnología (incluyendo técnicas y algoritmos para tratar con interacciones en entornos dinámicos y abiertos) y el uso de los agentes como simulación (para representar sistemas dinámicos complejos). El modelado basado en agentes se incluiría en esta última categoría.

El estudio de los sistemas multi-agente desde el punto de vista del origen de tecnología ha sido tratado desde la Ingeniería de Sistemas por parte de numerosos investigadores: (Dong et al., 2008), (Lo, 2012), (Innocenti et al., 2007), (Hu et al., 2013), (Wen et al., 2013), (Yu and Wang, 2013), (Zhu, 2014), por citar algunos, donde el interés principal es analizar las interacciones y, en muchos casos, la sincronización de agentes, que pueden ser desde simples robots hasta elementos coordinados de un sistema de control.

Sin embargo, en este artículo queremos centrarnos en los sistemas multi-agente desde el punto de vista del modelado basado en agentes, que puede definirse como un método computacional que permite al investigador crear, analizar y expe-

* Autor en correspondencia.

Correos electrónicos: maria.pereda@autom.uva.es (María Pereda), jesusm@autom.uva.es (Jesús M. Zamarreño)

rimentar con modelos compuestos de agentes que interactúan dentro de un entorno (Gilbert, 2008). Esta técnica de modelado se aplica de forma natural en aquellos sistemas que están compuestos a bajo nivel por multitud de elementos (modelados como agentes) que interactúan entre sí y con el entorno; por ejemplo, en procesos biológicos (fermentación, depuración, etc.) donde se modelaría a nivel de microorganismo (Pereda and Zamarreño, 2011), en procesos químicos modelando a nivel de molécula, procesos de cristalización (Wilensky, 2002), de difusión (Wilensky, 2007), etc. Para todos estos procesos, existen modelos de conocimiento o de primeros principios que se obtienen a partir de balances de masa y energía principalmente así como de otra serie de relaciones empíricas o experimentales cuyos resultados concuerdan con la realidad hasta cierto punto, pero que en ocasiones carecen de una base teórica que las sustente. No obstante, existen comparativas con otras técnicas de modelado como las basadas en dinámica de sistemas (Borshchev and Filippov, 2004), (Schieritz and Milling, 2003), (Izquierdo et al., 2008) o directamente con modelos basados en ecuaciones diferenciales (Van Dyke Parunak et al., 1998), (Rahmandad and Sterman, 2008) donde pueden verse las ventajas y desventajas de cada enfoque así como las razones para elegir una u otra alternativa en función del sistema a modelar.

Con el modelado basado en agentes se modelan los elementos microscópicos que constituyen el sistema así como la dinámica de estos a partir del conocimiento que se dispone sobre su comportamiento y, como resultado de las interacciones de los agentes, *emerge* un comportamiento macroscópico del sistema que se valida sobre la realidad observable. Este concepto, el de emergencia, es clave en el modelado basado en agentes, y se puede definir como la aparición de patrones macroscópicos estables a partir de las interacciones locales entre agentes. Uno de los primeros modelos que ilustraron el fenómeno emergente es el modelo Boids, desarrollado por (Reynolds, 1987) para simular el comportamiento colectivo de una bandada de pájaros.

Los modelos basados en agentes son modelos computacionales cuyo análisis debe hacerse a través de simulación. Actualmente se tienen diversas alternativas software (MASON (Luke et al., 2003), Repast (Collier, 2003), Swarm (Mina et al., 1996), NetLogo (Wilensky, 1999), etc.), pudiendo utilizarse también lenguajes de programación de propósito general, paquetes científicos especializados como Matlab (MATLAB, 2010) o simples hojas de cálculo para la creación de este tipo de modelos.

En los primeros años de aplicación de esta técnica de modelado no existía un estándar para la descripción de este tipo de modelos que favoreciera su difusión en la comunidad investigadora. Hoy en día, el estándar más extendido y asentado es el protocolo ODD (Overview, Design concepts, and Details) (Grimm et al., 2006). Su objetivo es estandarizar las descripciones de los modelos basados en agentes para hacer descripciones más comprensibles y completas, reduciendo los problemas de irreproducibilidad de los modelos. El protocolo ODD se organiza en torno a tres componentes principales: Visión general, Conceptos de diseño, y Detalles. Estas secciones, que deben ser escritas en un determinado orden, abarcan siete elementos que deben ser documentados con la suficiente profundidad y

claridad permitiendo que el modelo sea replicable por terceros: Propósito, Entidades, Variables de estado y escalas, Visión del proceso y planificación, Conceptos de diseño, Inicialización, Datos de entrada, y Submodelos. Sus autores han continuado publicando actualizaciones del protocolo con ejemplos de aplicación (Grimm et al., 2010).

En este trabajo se aporta una visión desde la Ingeniería de Sistemas para la representación de modelos basados en agentes. Este punto de vista proporciona un enfoque adecuado para la representación de los agentes, el entorno, y las relaciones de los agentes con otros agentes, y agentes con el entorno, los cuales conducen a comportamientos emergentes. Desde este punto de vista, los agentes son sistemas dinámicos cuyos estados evolucionan en el tiempo dentro de un dominio en el espacio de estados, como resultado de las relaciones complejas entre agentes y agentes con el entorno. Por lo tanto, el objetivo de este trabajo es abordar el modelado basado en agentes desde el punto de vista de la Ingeniería de Sistemas y Automática y proponer una representación de modelos basados en agentes que facilite posteriores desarrollos.

El artículo está estructurado de la siguiente manera. En esta primera sección se ha dado una panorámica del modelado basado en agentes, indicando sus características y su relación con el área de la Ingeniería de Sistemas y Automática. En la sección 2 se aborda la forma de representar este tipo de modelos de forma que sea útil como notación general en el área. En la sección 3 se muestran un par de ejemplos sencillos de modelado basado en agentes que ilustran la utilización de la notación propuesta y permiten centrarse en la dinámica de los agentes. Por último, en la sección 4 se presentan las principales conclusiones de este trabajo.

2. Representación de Modelos Basados en Agentes

La descripción matemática de los modelos basados en agentes no ha sido un tema muy estudiado aunque sí se pueden encontrar algunos estudios relacionados. Por ejemplo, (Hinkelmann et al., 2010) proponen un complemento al protocolo ODD consistente en una estructura algebraica para describir un modelo basado en agentes como un sistema dinámico usando funciones polinómicas. Con esta propuesta, la representación matemática es más precisa pero menos intuitiva y más compleja al estar basada en polinomios. En (Leombruni and Richiardi, 2005) se estudia cómo a partir de la formulación matemática de la dinámica de los agentes se puede determinar el comportamiento macroscópico de las variables agregadas, pero sus autores concluyen que su resolución algebraica no es factible en la práctica. No obstante, aunque en estos y otros artículos aparecen formulaciones matemáticas de los modelos basados en agentes, o bien estas no son completas y generales, o bien no se adaptan a las necesidades propias del campo de la Ingeniería de Sistemas y Automática.

El objetivo de la representación propuesta en este trabajo es acercar la técnica de modelado basado en agentes al campo de la Ingeniería de Sistemas y Automática. Puede ser considerada como un complemento al protocolo ODD, centrada en la descripción dinámica de los estados. En este sentido, desde el punto de

vista de la Ingeniería de Sistemas, los elementos principales del protocolo ODD que describen la dinámica vendrían recogidos en Entidades, Variables de estado y escalas, y Submodelos. Para facilitar una representación no ambigua y ulteriores análisis del sistema, complementamos las descripciones textuales del protocolo ODD con una representación matemática afín al área.

La representación propuesta se compone de una descripción matemática de los estados dinámicos de los agentes, un conjunto de relaciones funcionales entre los estados y los parámetros que conducen a la dinámica de los modelos, y una formulación de las funciones de salida que conduce a las variables agregadas de interés en el modelo.

2.1. Descripción matemática de los agentes

Un MBA se compone de una población de agentes (agrupados en tipos o 'razas', o de agentes todos del mismo tipo). En el caso de haber varios tipos de agentes, estos los identificamos con el índice j , que toma valores desde 1 al número total de tipos (b).

Cada agente i_j de tipo j tendrá una serie de atributos dinámicos que definen su estado en cada instante y que puede ser representado mediante un vector de estados dinámicos $\vec{x}_{i_j}^j(t)$ en el instante de tiempo t . Para disponer de una notación más compacta, todos los agentes del mismo tipo j se agrupan en una matriz de vectores de estado $X^j(t)$.

Además, los agentes pueden estar descritos por atributos estáticos (invariables en el tiempo) que favorecen la heterogeneidad y que podemos representar con un vector $\vec{p}_{i_j}^j$. Estos parámetros pueden influir en la dinámica y el comportamiento de los agentes.

En algunos modelos, el número de agentes n_j podría variar durante la simulación; esto se expresará como una dependencia temporal: $n_j(t)$. Si el número de agentes es constante en el modelo, esta dependencia temporal puede ser obviada.

En resumen, el estado de un agente en tiempo t se representa mediante el vector $\vec{x}_{i_j}^j(t)$, siendo cada componente del vector el valor de un atributo dinámico del agente.

- $j = 1, \dots, b$ identifica el tipo de agente (raza)
- $i_j = 1, \dots, n_j(t)$ identifica el agente de tipo j
- $n_j(t)$ es el número de agentes de tipo j en el instante t
- $\vec{x}_{i_j}^j(t) \in \mathbb{R}^{m_j}$, m_j es el número de estados para el tipo j
- $\vec{p}_{i_j}^j \in \mathbb{R}^{q_j}$, q_j es el número de parámetros para el tipo j

Como se ha indicado antes, para disponer de una notación más compacta, podemos representar los estados del conjunto de agentes de tipo j como $X^j(t) = \left(\vec{x}_1^j(t) \dots \vec{x}_{n_j(t)}^j(t) \right)$ donde $X^j(t) \in M_{m_j \times n_j(t)}(\mathbb{R})$

De forma similar, representamos los parámetros del conjunto de agentes de tipo j como $P^j = \left(\vec{p}_1^j \dots \vec{p}_{n_j(t)}^j \right)$ donde $P^j \in M_{q_j \times n_j(t)}(\mathbb{R})$

Finalmente, el entorno en el que se desenvuelven los agentes estará parametrizado de alguna forma (por ejemplo, las dimensiones del mundo en el que viven los agentes). Este conjunto de parámetros que definen el entorno lo agrupamos en un vector $\vec{\alpha}$ cuyos elementos tomarán valores en un subespacio \mathbb{S} de

\mathbb{R}^l siendo l el número de parámetros con el que se define el entorno. Es decir, el entorno vendrá descrito como $\vec{\alpha} \in \mathbb{S} \subset \mathbb{R}^l$.

Por último, en la Ingeniería de Sistemas, muchas veces nos interesará considerar el modelo como un bloque que puede conectarse con otros para formar un sistema más grande, con lo cual habrá señales externas que afecten al comportamiento del modelo. Estas señales externas las representaremos con el vector $\vec{u} \in \mathbb{R}^E$, siendo E el número de entradas externas.

2.2. Métodos dinámicos

La dinámica del modelo a nivel macroscópico es consecuencia de la dinámica de los agentes a nivel microscópico. La dinámica de los agentes está representada por la evolución de sus estados (atributos dinámicos) y es causada por la propia evolución del agente o por interacciones del agente con otros agentes o con el entorno. Asimismo, los cambios de estado de un agente pueden provenir de múltiples causas por lo que podemos descomponer la dinámica en diversos métodos (o submodelos en el estándar ODD).

Es decir, un agente i_j de raza j tiene una interacción 'a' con otros agentes y/o con el entorno. Esta interacción se expresa por medio de un método 'a' que depende de los estados y parámetros de todos los agentes (en general) y de los parámetros del entorno. Esta interacción puede originar un cambio en algunos de los estados de los demás agentes, no solamente de él mismo. Los métodos también pueden representar una consecuencia del paso del tiempo en vez de representar una interacción; por ejemplo, el incremento de edad de un agente.

Expresar en formulación matemática estas interacciones no es un tema trivial, debido a que las interacciones de unos agentes pueden afectar a otros agentes.

Nosotros sólo estamos interesados en las relaciones funcionales entre estados y parámetros que se desprenden de estas interacciones. En caso de requerirse una completa descripción formal de los métodos, se debería emplear una descripción algorítmica o utilizar algún lenguaje formal, como el lenguaje Z (Potter et al., 1996).

Para tener en cuenta las múltiples interacciones que pueden suceder en cada periodo de tiempo, consideraremos tres pasos ordenados que deben ejecutarse en cada instante t del modelo. En primer lugar, se realiza una copia de la matriz de estados de los agentes (Z^j) y del número de agentes (c_j) en el instante de tiempo t (1).

$$\{Z^j = X^j(t); \quad c_j = n_j(t)\} \quad (1)$$

donde $j = 1, \dots, b$.

Después (2), los estados evolucionan como consecuencia de la aplicación de los métodos, es decir, los estados de los agentes se actualizan (F_{i_j, a_j}^j es el método que representa la interacción a_j causada por el agente i_j de tipo j). Este procedimiento se realiza sobre todos los agentes de tipo j y por cada método de cada tipo de agente.

$$\{\vec{z}^k; k = 1, \dots, b\} = F_{i_j, a_j}^j(\{Z^k; k = 1, \dots, b\}, \{P^k; k = 1, \dots, b\}, \vec{u}, \vec{\alpha}) \quad (2)$$

donde $\bar{Z}^k \in M_{m_k \times \bar{c}_k}(\mathbb{R})$ son los nuevos estados modificados de todos los agentes y $\bar{c}_k \in \mathbb{N}$ es el nuevo número de agentes de tipo k como consecuencia de la aplicación del método. La ecuación 2 debe aplicarse por cada posible interacción $a_j = 1, \dots, A_j$ donde $A_j \in \mathbb{N}$ es el número de interacciones del tipo j , y sobre todos los agentes $i_j = 1, \dots, c_j$ de cada tipo $j = 1, \dots, b$. Por lo tanto, el número de veces que se aplica la ecuación 2 depende del número de agentes y del número de acciones que debe realizar cada uno. Después de la ejecución de cada método, es necesario actualizar los estados $Z^k = \bar{Z}^k$ y el número de agentes $c_k = \bar{c}_k$ para tener en cuenta todas las posibles interacciones (por ejemplo, la acción de un agente puede modificar el estado de otro agente) y la creación o destrucción de agentes como consecuencia de estas acciones, antes de aplicar el siguiente.

En la aplicación de los procedimientos es importante que el orden de aplicación sobre los diversos agentes no sea siempre el mismo (para evitar dar prioridades a unos agentes sobre otros). Para ello, el software especializado en MBA, como Net-Logo (Wilensky, 1999), trata al conjunto de agentes en orden aleatorio; es decir, no los considera que estén en un determinado orden.

Finalmente, una vez que se hayan aplicado todos los métodos, los estados en tiempo $t+1$ se actualizan basándose en las copias (3).

$$\{X^j(t+1) = \bar{Z}^j; \quad n_j(t+1) = \bar{c}_j\} \quad (3)$$

donde $j = 1, \dots, b$.

2.3. Variables agregadas

En cada iteración de la simulación, se calculan las variables agregadas $\vec{y}(t)$ que recogen el comportamiento global del modelo. Estas variables se expresan por medio de relaciones entre los estados de los agentes y los parámetros de los agentes y del entorno.

Esta relación funcional en el instante de tiempo t puede relacionar también instantes de tiempo pasados, e incluso el valor pasado de la variable agregada (como en un acumulador).

$$\vec{y}(t) = \vec{g}(\vec{y}(t-1), \{X^j(\tau); \quad j = 1, \dots, b; \quad \tau = 0, \dots, t\}, \{P^j; \quad j = 1, \dots, b\}, \vec{\alpha}) \quad (4)$$

donde $\vec{y}(t) \in \mathbb{R}^S$, siendo S el número de variables agregadas.

2.4. Simulación

La representación propuesta en los apartados anteriores facilita una descripción coherente con las herramientas disponibles en el área de la Ingeniería de Sistemas y Automática. Asimismo, esa descripción es la base de la simulación del sistema, elemento principal de análisis del sistema, ya que, en general, los métodos dinámicos representados por las funciones F_{i_j, a_j}^j suelen ser demasiado complejos como para realizar ningún tipo de análisis matemático, eso sin tener en cuenta las múltiples

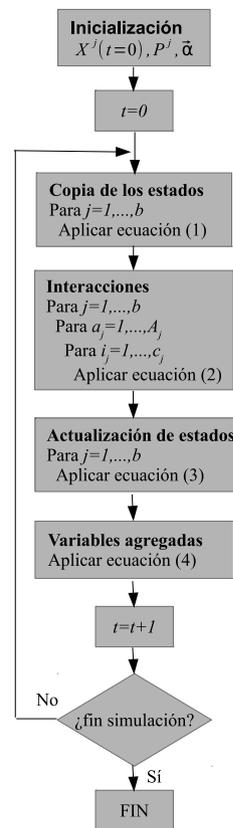


Figura 1: Diagrama de flujo de la simulación MBA.

interacciones entre agentes. La simulación del sistema MBA consiste en un proceso típico de inicialización de variables (de estado y parámetros), y aplicación de las ecuaciones (1), (2), (3) y (4) de forma iterativa hasta la condición de finalización de la simulación. Este proceso está representado en la figura 1.

3. Ejemplos

En esta sección se presentan un par de ejemplos ilustrativos descritos mediante la representación de la sección anterior. Se han añadido algunos comentarios para clarificar algunas particularidades que aparecen en los ejemplos.

3.1. Modelo de Segregación de Schelling

Aunque no sea de interés directamente en el campo de la Automática, se ha elegido este primer ejemplo por su sencillez y por razones históricas, al ser uno de los primeros modelos basados en agentes en las ciencias sociales.

En la década de los 70, Thomas Schelling propuso un modelo de población sencillo para ilustrar cómo, incluso con suposiciones mínimas sobre las preferencias vecinales de los individuos, una ciudad integrada evolucionaría a una ciudad segregada, incluso cuando todos los individuos fueran partidarios de la integración. Schelling colocó monedas de centavo y 10 centavos en un tablero de ajedrez representando agentes de dos

grupos de la sociedad y fue moviéndolos de acuerdo a un conjunto de reglas. El tablero representaba una ciudad, y cada casilla del tablero una casa o vivienda. El vecindario de un agente en una localización del tablero consistía en sus casillas adyacentes. Las reglas determinaban la felicidad de cada agente en su localización actual. Si el agente era infeliz, se movería a otra localización en el tablero.

Este modelo (Wilensky, 1997) aparece en la biblioteca de ejemplos de NetLogo (Wilensky, 1999). Se trata de una versión del modelo de segregación de Thomas Schelling, también conocido como *Schelling tipping model* (Schelling, 1969).

En la implementación de NetLogo (software en el cual los agentes básicos son representados como tortugas), los agentes tortuga se diferencian en la característica del color; es decir, tenemos tortugas rojas y tortugas verdes, pero el comportamiento de todos ellos es el mismo. Por lo tanto, haciendo uso de la representación propuesta, se va a definir un único tipo de agente, la tortuga, con tres estados: sus coordenadas en un entorno bidimensional (*coordenada_x*, *coordenada_y*) y un estado de felicidad (*feliz?*), como se muestra en la tabla 1. Para separar las tortugas en las dos grupos, rojas y verdes, se va a utilizar el parámetro *color*. Ambos grupos de agentes contarán con un porcentaje de vecinos similares a ellos tal que su estado sea de felicidad; esto se expresa con el parámetro *% - similar - deseado* (tabla 2).

La aplicación de la notación propuesta da lugar a la descripción que se desarrolla en los siguientes apartados.

3.1.1. Descripción de los agentes

Dado que solamente existe una raza de agente, $b = 1$ y $j = 1$. El modelo estará compuesto de n_1 agentes, cada uno de los cuales ($i_1 = 1, \dots, n_1$) estará definido por un vector de estados $\vec{x}_{i_1}^1 \in \mathbb{R}^3$, donde

$$\vec{x}_{i_1}^1 = \begin{pmatrix} \text{coordenada } x \\ \text{coordenada } y \\ \text{feliz?} \end{pmatrix}$$

La matriz de vectores de estado del sistema vendrá definida como $X^1(t) = (\vec{x}_1^1 \dots \vec{x}_{n_1}^1)$ donde $X^1(t) \in M_{3 \times n_1}(\mathbb{R})$

Para una formulación más cómoda de los métodos dinámicos, por conveniencia se realizan las particiones ($x_1^1(t)$, $x_2^1(t)$, $x_3^1(t)$) de $X^1(t)$, que permiten la selección de cada estado de los agentes de forma aislada, de tal forma que podemos expresar la matriz de estados como

$$X^1(t) = \begin{pmatrix} x_1^1(t) \\ x_2^1(t) \\ x_3^1(t) \end{pmatrix}$$

donde $x_1^1(t), \dots, x_3^1(t) \in M_{1 \times n_1}(\mathbb{R})$.

De igual forma, cada agente ($i_1 = 1, \dots, n_1$) está definido por un vector de parámetros $\vec{p}_{i_1}^1 \in \mathbb{R}^2$, el cual está compuesto por

$$\vec{p}_{i_1}^1 = \begin{pmatrix} \text{color} \\ \% - \text{similar} - \text{deseado} \end{pmatrix}$$

Al igual que con los estados, podemos expresar los parámetros de todos los agentes agrupándolos en una matriz $P^1 = (\vec{p}_1^1 \dots \vec{p}_{n_1}^1)$ donde $P^1 \in M_{2 \times n_1}(\mathbb{R})$ y realizar una partición de dicha matriz para seleccionar de forma conjunta (sobre todos los agentes) cada parámetro,

$$P^1 = \begin{pmatrix} p_1^1 \\ p_2^1 \end{pmatrix}$$

donde $p_1^1, p_2^1 \in M_{1 \times n_1}(\mathbb{R})$.

Finalmente, el entorno en el que se mueven los agentes está representado por un mundo bidimensional ($\vec{\alpha} \in \mathbb{R}^2$),

$$\vec{\alpha} = \begin{pmatrix} \text{numero} - \text{celdas} - \text{ancho} \\ \text{numero} - \text{celdas} - \text{alto} \end{pmatrix}$$

3.1.2. Métodos dinámicos

En este modelo se tiene un único método ($F_{i_1,1}^1$) para cada agente en el instante t . En el caso de NetLogo, se modela el comportamiento de dos tipos de tortugas en una laguna mítica. Las tortugas de color rojo y las tortugas de color verde se llevan bien entre sí. Pero cada tortuga quiere asegurarse de que vive cerca (medido a partir de sus coordenadas x_1^1, x_2^1) de alguna de 'las suyas' (indicado por el parámetro de color p_1^1); es decir, cada tortuga roja quiere vivir cerca de, al menos, algunas (medido por el parámetro *%-similar-deseado*; p_2^1) tortugas rojas, y cada tortuga verde quiere vivir cerca de al menos algunas (medido por p_2^1) tortugas verdes. Si las tortugas no tienen los suficientes (comparado con p_2^1) vecinos del mismo color que ellas (indicado por p_1^1), son infelices (indicado por el estado x_3^1), y por lo tanto saltarán a una celda cercana (nuevo valor de los estados x_1^1, x_2^1) dentro de los límites del entorno (definido por $\vec{\alpha}$). Si son felices (indicado por x_3^1), no harán nada. La simulación se detiene cuando todas las tortugas son felices.

Por lo tanto, y teniendo en cuenta que el número de agentes es constante siempre, podemos aplicar (1), que para este caso particular se reduciría a

$$Z^1 = X^1(t) \quad (5)$$

De forma equivalente a las particiones realizadas sobre la matriz de estados, las realizamos sobre las copias Z^1 y \bar{Z}^1

$$Z^1 = \begin{pmatrix} z_1^1 \\ z_2^1 \\ z_3^1 \end{pmatrix} \quad \bar{Z}^1 = \begin{pmatrix} \bar{z}_1^1 \\ \bar{z}_2^1 \\ \bar{z}_3^1 \end{pmatrix}$$

donde $z_1^1, \dots, z_3^1 \in M_{1 \times n_1}(\mathbb{R})$ y $\bar{z}_1^1, \dots, \bar{z}_3^1 \in M_{1 \times n_1}(\mathbb{R})$.

Aplicamos (2) teniendo en cuenta la descripción del comportamiento de los agentes, que se puede representar en un único método,

$$\begin{pmatrix} \bar{z}_1^1 \\ \bar{z}_2^1 \\ \bar{z}_3^1 \end{pmatrix} = F_{i_1}^1(z_1^1, z_2^1, z_3^1, p_1^1, p_2^1, \vec{\alpha}) \quad (6)$$

que se debe aplicar sobre cada agente $i_1 = 1, \dots, n_1$.

Finalmente, aplicamos (3) para actualizar los estados del sistema en el siguiente instante de tiempo.

$$X^1(t+1) = \bar{Z}^1 \quad (7)$$

3.1.3. Variables agregadas

En este modelo se calculan dos variables agregadas, por lo que el vector $\vec{y}(t)$ es bidimensional ($\vec{y}(t) \in \mathbb{R}^2$).

La primera variable agregada $y_1(t)$ es la media del porcentaje de tortugas similares en el vecindario de cada tortuga, lo cual

Tabla 1: Estados de las tortugas

Tipo de agente	Estado	Descripción
Tortuga	<i>coordenada x</i>	coordenada x de la celda en la que el agente está posicionado en el entorno
Tortuga	<i>coordenada y</i>	coordenada y de la celda en la que el agente está posicionado en el entorno
Tortuga	<i>feliz?</i>	(variable booleana) verdadero si un porcentaje (%-similar-deseado) igual o mayor de los vecinos de esa tortuga son del mismo color que ella

Tabla 2: Parámetros

Tipo de agente	Parámetro	Descripción
Tortuga	<i>color</i>	el color de las tortugas puede ser rojo o verde
Tortuga	<i>% - similar - deseado</i>	porcentaje de tortugas del mismo color que cada tortuga desea en su vecindario

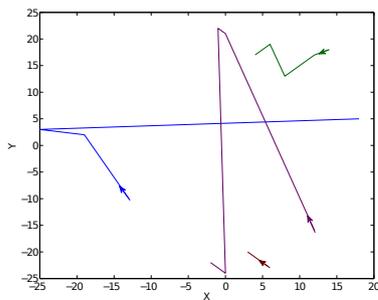
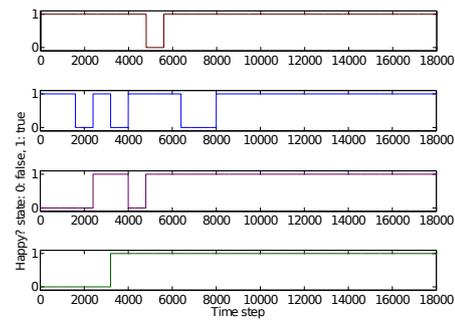


Figura 2: Evolución de las posiciones de 4 tortugas.

Figura 3: Evolución del estado *feliz?* de 4 tortugas.

depende de las coordenadas de los agentes ($x_1^1(t)$, $x_2^1(t)$) y el parámetro *color* (p_1^1), lo cual expresamos con la ecuación

$$y_1(t) = g_1(x_1^1(t), x_2^1(t), p_1^1) \quad (8)$$

La segunda variable agregada, $y_2(t)$, será el porcentaje de tortugas infelices en relación al número total de tortugas, para lo cual necesitamos el estado de felicidad de todas las tortugas ($x_3^1(t)$), lo que expresaremos con la ecuación

$$y_2(t) = g_2(x_3^1(t)) \quad (9)$$

3.1.4. Simulación

Para visualizar el comportamiento dinámico del modelo presentado en las secciones anteriores, vamos a realizar una simulación con 800 tortugas (400 verdes y 400 rojas), siendo el parámetro *% - similar - deseado*=62% para todas las tortugas. Como ejemplo, la evolución dinámica de los estados de 4 tortugas elegidas al azar se muestra en las figuras 2 y 3. En estas figuras se puede ver el comportamiento dinámico de las tortugas según se mueven por el entorno en busca de sus objetivos así como la evolución dinámica del estado de felicidad que va pasando por un transitorio hasta que todas las tortugas son *felices* en el estacionario.

3.2. Modelo de calor en una habitación

Se ha elegido un segundo ejemplo, también sencillo, pero con mayor relación con la Automática, donde además se puede

ver que un modelo basado en agentes podría ser utilizado también como base para el análisis y diseño de reguladores.

Se ha tomado como base un modelo de la librería de NetLogo, un modelo (Wilensky, 1998) que simula el funcionamiento de un termostato ubicado en una habitación con una fuente de calor controlada. La distribución de calor en la habitación se simula mediante agentes (bolas de calor) que se mueven por una habitación bidimensional según una reglas determinadas. El modelo no trata de ser físicamente realista, sino que pretende servir de ejemplo sencillo con el que mostrar las posibilidades del modelado basado en agentes. Hemos modificado el modelo original en dos aspectos: incorporación de una dinámica más realista (primer orden) en la respuesta del termostato, y posibilidad de desconectar el termostato para poder centrarnos en la dinámica del modelo en lazo abierto (Pereda and Zamarreño, 2014).

Haciendo uso de la representación propuesta, se define un tipo de agente que denominaremos *bolas de calor*. Las bolas de calor son agentes móviles definidos por tres estados; sus dos coordenadas en un entorno bidimensional (*coordenada_x*, *coordenada_y*) y su dirección (*direccion*) (tabla 3).

3.2.1. Descripción de los agentes

En este modelo, como se ha expuesto anteriormente, tendremos un tipo de agente: bolas de calor; $j = 1$. Su estado dinámico viene definido por sus coordenadas y por su dirección de movimiento, por lo que $\vec{x}_{i_1}^1 \in \mathbb{R}^3$ donde $i_1 = 1, \dots, n_1(t)$, sien-

Tabla 3: Estados de los agentes para el modelo de calor

Tipo de agente	Estado	Descripción
Bola de calor	<i>coordinada_x</i>	coordinada x de la bola de calor
Bola de calor	<i>coordinada_y</i>	coordinada y de la bola de calor
Bola de calor	<i>direccion</i>	dirección de movimiento

do $n_1(t)$ el número total de bolas de calor. El número de agentes puede variar a lo largo del tiempo debido a la generación de nuevos agentes (creados por la fuente de calor) y a la desaparición de los mismos (por su dispersión fuera del entorno de la habitación). El vector de estados de cada bola de calor sería, por tanto,

$$\vec{x}_i^1 = \begin{pmatrix} \textit{coordinada_x} \\ \textit{coordinada_y} \\ \textit{direccion} \end{pmatrix}$$

Por comodidad, agrupamos todos los estados de todas las bolas de calor en una matriz de estados $X^1(t) \in M_{3 \times n_1}(\mathbb{R})$, formado por $X^1(t) = (\vec{x}_1^1 \dots \vec{x}_{n_1(t)}^1)$.

Para facilitar el acceso a cada estado de todos los agentes, podemos definir una partición de dicha matriz,

$$X^1(t) = \begin{pmatrix} x_1^1(t) \\ x_2^1(t) \\ x_3^1(t) \end{pmatrix}$$

donde $x_1^1(t), x_2^1(t), x_3^1(t) \in M_{1 \times n_1(t)}(\mathbb{R})$.

También debemos parametrizar el entorno. El entorno vendrá caracterizado por las dimensiones de la habitación (suponemos un entorno bidimensional), la calidad del aislamiento de las paredes, y las dimensiones y posición del termostato. Suponemos asimismo que la fuente de calor se encuentra fija en el centro de la habitación. Por lo tanto, el entorno vendrá definido por un vector $\vec{\alpha} \in \mathbb{R}^7$ expresado como

$$\vec{\alpha} = \begin{pmatrix} \textit{ancho} - \textit{habitacion} \\ \textit{alto} - \textit{habitacion} \\ \textit{aislamiento} - \textit{paredes} \\ \textit{ancho} - \textit{termostato} \\ \textit{alto} - \textit{termostato} \\ \textit{posX} - \textit{termostato} \\ \textit{posY} - \textit{termostato} \end{pmatrix}$$

Finalmente, la señal externa podemos considerarla como una potencia $u \in \mathbb{R}$ que provocará la aparición de nuevas bolas de calor.

3.2.2. Métodos dinámicos

Los agentes *bolas de calor* tienen un comportamiento que provoca su desplazamiento por el entorno.

Para establecer la descripción dinámica del comportamiento, aplicamos (1) para operar con las copias de los estados en el instante de tiempo t , teniendo en cuenta que el número de agentes es variable.

$$\{Z^1 = X^1(t); \quad c_1 = n_1(t)\} \quad (10)$$

Para aplicar (2), en primer lugar, realizamos una partición

$$Z^1 = \begin{pmatrix} z_1^1 \\ z_2^1 \\ z_3^1 \end{pmatrix}$$

donde $z_1^1, \dots, z_3^1 \in M_{1 \times c_1}(\mathbb{R})$.

La ecuación dinámica de los estados de las bolas de calor consiste en el comportamiento dinámico del desplazamiento de las bolas de calor. Este movimiento es muy simple. Las bolas de calor realizan un desplazamiento unitario en una dirección dada, lo que provoca un cambio de z_1^1, z_2^1 a \bar{z}_1^1, \bar{z}_2^1 . En caso de alcanzar los límites de la habitación (indicado por α_1, α_2) las bolas de calor podrán rebotar hacia el interior de la habitación (cambiando la dirección z_3^1 a un nuevo valor \bar{z}_3^1) o bien seguir desplazándose hacia fuera efectivamente desapareciendo (modificándose el número de agentes, de c_1 a \bar{c}_1 y las dimensiones de \bar{Z}^1) en función de una probabilidad relacionada con el aislamiento de las paredes (α_3). Asimismo, pueden aparecer nuevos agentes generados por la fuente de calor; el número de nuevos agentes generados dependerá del valor de la señal externa u y provocará la modificación de c_1 a \bar{c}_1 y las dimensiones de \bar{Z}^1 . Esta dependencia funcional se expresa como

$$\begin{pmatrix} \bar{z}_1^1 \\ \bar{z}_2^1 \\ \bar{z}_3^1 \end{pmatrix} = F_{i,1}^1(z_1^1, z_2^1, z_3^1, u, \alpha_1, \alpha_2, \alpha_3) \quad (11)$$

Finalmente, aplicamos (3) para actualizar los estados del sistema en el siguiente instante de tiempo.

$$\{X^1(t+1) = \bar{Z}^1; \quad n_1(t+1) = \bar{c}_1\} \quad (12)$$

3.2.3. Variables agregadas

Para estimar la temperatura medida por el sensor del termostato, supondremos una dinámica de primer orden en el propio sensor y realizaremos una contribución proporcional al número de bolas de calor localizadas sobre la superficie del termostato (a mayor densidad de bolas de calor, mayor temperatura en la habitación). Por lo tanto, esta variable agregada ($y(t) \in \mathbb{R}$, temperatura, dependerá de su valor anterior ($y(t-1)$), de la posición de las bolas de calor ($x_1^1(t), x_2^1(t)$), y de la posición (α_6, α_7) y dimensiones (α_4, α_5) del termostato. Esta dependencia funcional se expresa en la siguiente ecuación.

$$y(t) = g(y(t-1), x_1^1(t), x_2^1(t), \alpha_4, \alpha_5, \alpha_6, \alpha_7) \quad (13)$$

3.2.4. Simulación

Se simuló el modelo con una temperatura inicial en la habitación de 20 °C (lo que corresponde con un determinado número de bolas de calor que son distribuidas aleatoriamente por la habitación). Si nos fijamos en el comportamiento de tres de estos agentes, podemos ver la evolución de sus estados durante los primeros instantes de simulación: en la figura 4 se muestra su posición a lo largo del tiempo y en la figura 5 se puede ver la dirección en grados donde se aprecia el cambio de dirección que han sufrido los tres agentes en el momento que *chocaron* contra la pared. Si no ponemos ningún controlador (ni fuente de calor) podríamos ver (figura 6) que la temperatura en la habitación (variable agregada) va disminuyendo a lo

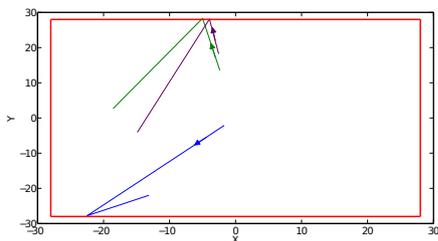


Figura 4: Evolución de la posición de los agentes.

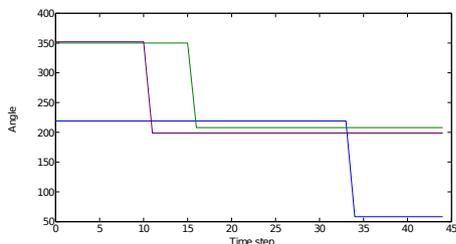


Figura 5: Evolución de la dirección a lo largo del tiempo.

largo del tiempo ya que no hay creación de nuevos agentes y eventualmente los agentes existentes van traspasando la pared y desapareciendo en función del parámetro de aislamiento.

Si activamos un controlador sencillo todo/nada, estaríamos configurando un lazo de regulación como se ve en la figura 7 donde el modelo basado en agentes ha sido integrado como proceso a controlar. En este caso, la simulación arroja el resultado mostrado en la figura 8 ya que cada vez que el controlador activa la fuente de calor, aparecen nuevos agentes que provocan el incremento de la temperatura.

4. Conclusiones

En este trabajo se ha expuesto el modelado basado en agentes como estrategia de modelado válida para determinado tipo de sistemas donde su comportamiento a alto nivel es resultado de las interacciones de elementos de bajo nivel (representados como agentes). Se ha abordado el tema de la representación desde el enfoque de la Ingeniería de Sistemas poniendo el énfasis en el comportamiento dinámico de los agentes fruto de la evolución

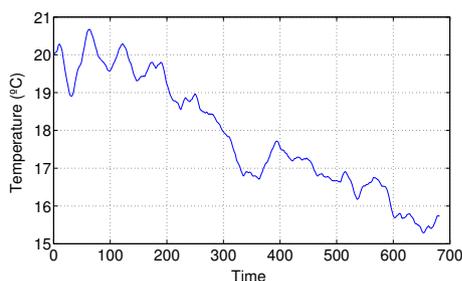


Figura 6: Evolución de la temperatura (variable agregada) a lo largo del tiempo, sin fuente de calor.

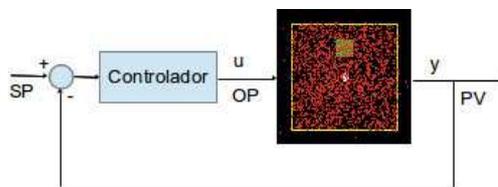


Figura 7: Lazo de control del modelo basado en agentes.

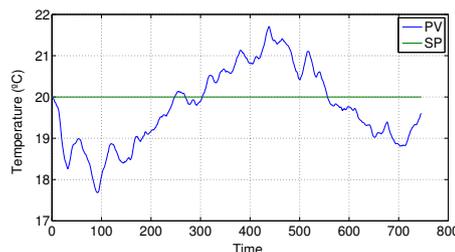


Figura 8: Evolución de la temperatura (variable agregada) a lo largo del tiempo en modo controlado.

temporal de sus estados. La representación que se propone no pretende sustituir al actual protocolo ODD, sino complementarlo, sobre todo para su uso en el campo de la ingeniería de sistemas y automática. En este trabajo se han presentado un par ejemplos ilustrativos sencillos de esta técnica.

English Summary

Agent-Based Modelling: an Approach from the Systems Engineering.

Abstract

Agent-Based Modelling (ABM) is a modelling technique with great success in fields like ecology, social sciences, economy, etc. However, it is not so widespread in the Automatic field. In this paper, we present how to deal with ABM from the point of view of the System Engineering and Automatic Control field and the specific issues to take into account as modelling technique. Besides, we propose a mathematical description that is illustrated through two simple examples.

Keywords:

Agents, Dynamic modelling, Systems engineering, State space, Conceptual representations.

Agradecimientos

Los autores agradecen el soporte a la Universidad de Valladolid bajo la beca *Ayuda para la Formación de Personal Investigador*, y al Ministerio de Economía y Competitividad, bajo el proyecto *Metodología de diseño de estrategias de control jerárquico y distribuido basadas en MPCs para el control total de sistemas integrados y redes de distribución* (DPI 2012-39381-C02-02).

Referencias

- Borshchev, A., Filippov, A., July 2004. From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. In: Proceedings of the 22nd International Conference of the System Dynamics Society, Oxford, England.
- Collier, N., 2003. RePast: An Extensible Framework for Agent Simulation. <http://repast.sourceforge.net/> (last visited August 2013).
- Dong, J., xin Yin, Y., xiang Peng, K., 2008. Industrial process coordinated and controlled based on multi-agent technology. *Systems Engineering - Theory & Practice* 28 (10), 119 – 124.
DOI: [http://dx.doi.org/10.1016/S1874-8651\(10\)60004-X](http://dx.doi.org/10.1016/S1874-8651(10)60004-X)
- Galán, J. M., Izquierdo, L. R., Izquierdo, S. S., Santos, J. I., del Olmo, R., López-Paredes, A., Edmonds, B., 2009. Errors and artefacts in agent-based modelling. *Journal of Artificial Societies and Social Simulation* 12 (1), 1.
URL: <http://jasss.soc.surrey.ac.uk/12/1/1.html>
- Gilbert, G. N., 2008. Agent-based models. Quantitative applications in the social sciences. Sage.
- Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S. K., Huse, G., Huth, A., Jepsen, J. U., Jorgensen, C., Mooij, W. M., Muller, B., Pe'er, G., Piou, C., Railsback, S. F., Robbins, A. M., Robbins, M. M., Rossmanith, E., Ruger, N., Strand, E., Souissi, S., Stillman, R. A., Vabo, R., Visser, U., DeAngelis, D. L., 2006. A standard protocol for describing individual-based and agent-based models. *Ecological Modelling* 198, 115–126.
- Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J., Railsback, S. F., Nov. 2010. The ODD protocol: A review and first update. *Ecological Modelling* 221 (23), 2760–2768.
DOI: <http://dx.doi.org/10.1016/j.ecolmodel.2010.08.019>
- Hinkelmann, F., Murrugarra, D., Jarrah, A. S., Laubenbacher, R. C., 2010. A mathematical framework for agent based models of complex biological networks. *Computing Research repository abs/1006.0408*.
URL: <http://dblp.uni-trier.de/db/journals/corr/corr1006.html#abs-1006-0408>
- Hu, H.-X., Liu, A., Xuan, Q., Yu, L., Xie, G., 2013. Second-order consensus of multi-agent systems in the cooperation-competition network with switching topologies: A time-delayed impulsive control approach. *Systems & Control Letters* 62 (12), 1125 – 1135.
DOI: <http://dx.doi.org/10.1016/j.sysconle.2013.09.002>
- Innocenti, B., López, B., Salvi, J., 2007. A multi-agent architecture with cooperative fuzzy control for a mobile robot. *Robotics and Autonomous Systems* 55 (12), 881 – 891, robotics and Autonomous Systems in the 50th Anniversary of Artificial Intelligence Campus Multidisciplinary in Perception and Intelligence.
DOI: <http://dx.doi.org/10.1016/j.robot.2007.07.007>
- Izquierdo, L., Galán, J. M., Santos, J. I., del Olmo, R., 2008. Modelado de sistemas complejos mediante simulación basada en agentes y mediante dinámica de sistemas. *Empiria: Revista de metodología de ciencias sociales* 16, 85–112.
- Leombruni, R., Richiardi, M., Sep. 2005. Why are economists sceptical about agent-based simulations? *Physica A: Statistical Mechanics and its Applications* 355 (1), 103–109.
DOI: <http://dx.doi.org/10.1016/j.physa.2005.02.072>
- Lo, S. K., 2012. A collaborative multi-agent message transmission mechanism in intelligent transportation system - a smart freeway example. *Information Sciences* 184 (1), 246 – 265.
DOI: <http://dx.doi.org/10.1016/j.ins.2011.08.024>
- Luck, M., McBurney, P., Shehory, O., Willmott, S., 2005. Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing). AgentLink.
- Luke, S., Balan, G. C., Panait, L., Cioffi-Revilla, C., Paus, S., 2003. MASON: A Java Multi-Agent Simulation Library. In: Macal, C. M., North, M., Sallach, D. (Eds.), Proceedings of Agent 2003, Conference on Challenges in Social Simulation. Argonne National Laboratory.
- Macal, C. M., North, M. J., Dec. 2006. Tutorial on agent-based modeling and simulation part 2: How to model with agents. In: Winter Simulation Conference, 2006. WSC 06. Proceedings of the. pp. 73–83.
DOI: <http://dx.doi.org/10.1109/wsc.2006.323040>
- MATLAB, 2010. version 7.10.0 (R2010b). The MathWorks Inc., Natick, Massachusetts.
- Minar, N., Burkhart, R. and Langton, C., Askenazi, M., 1996. The swarm simulation system: A toolkit for building multi-agent simulations. Santa Fe Institute working paper 96-06-042. Swarm available at <http://www.swarm.org> (last visited August 2013).
- Pereda, M., Zamarreño, J. M., Jun. 2011. Agent-based modeling of an activated sludge process in a batch reactor. In: 2011 19th Mediterranean Conference on Control & Automation (MED). IEEE, Corfu, pp. 1128–1133.
DOI: <http://dx.doi.org/10.1109/MED.2011.5983027>
- Pereda, M., Zamarreño, J. M., 2014. "Thermostat II" (Version 3). CoMSES Computational Model Library. Retrieved from: <https://www.openabm.org/model/4234/version/3> (last visited June 2014).
- Potter, B., Sinclair, J., Till, D., 1996. Introduction to Formal Specification and Z (2nd Edition). Prentice Hall PTR.
- Rahmandad, H., Sterman, J., 2008. Heterogeneity and network structure in the dynamics of diffusion: Comparing agent-based and differential equation models. *Management Science* 54 (5), 998–1014.
DOI: <http://dx.doi.org/10.1287/mnsc.1070.0787>
- Reynolds, C. W., Aug. 1987. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Computer Graphics* 21 (4), 25–34.
DOI: <http://dx.doi.org/10.1145/37402.37406>
- Schelling, T. C., 1969. Models of segregation. *The American Economic Review* 59 (2), 488–493.
- Schieritz, N., Milling, P. M., 2003. Modeling the forest or modeling the trees. comparison of sd and ab simulation. In: Proceedings of the 21st International Conference of the System Dynamics Society.
- Torsun, I., 1995. Foundations of Intelligent Knowledge-Based Systems. Library and Information Science. Academic Press Limited.
- Van Dyke Parunak, H., Savit, R., Riolo, R. L., 1998. Agent-based modeling vs. equation-based modeling: A case study and users' guide. In: Sichman, J. S. a., Conte, R., Gilbert, N. (Eds.), Multi-Agent Systems and Agent-Based Simulation. Vol. 1534 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 10–25.
DOI: http://dx.doi.org/10.1007/10692956_2
- Wen, G., Hu, G., Yu, W., Cao, J., Chen, G., 2013. Consensus tracking for higher-order multi-agent systems with switching directed topologies and occasionally missing control inputs. *Systems & Control Letters* 62 (12), 1151 – 1158.
DOI: <http://dx.doi.org/10.1016/j.sysconle.2013.09.009>
- Wilensky, U., 1997. NetLogo Segregation model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/Segregation> (last visited March 2013).
- Wilensky, U., 1998. NetLogo Thermostat model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/Thermostat> (last visited March 2013).
- Wilensky, U., 1999. NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/> (last visited March 2013).
- Wilensky, U., 2002. NetLogo Crystallization Basic model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/CrystallizationBasic> (last visited March 2013).
- Wilensky, U., 2007. NetLogo Solid Diffusion model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/models/SolidDiffusion> (last visited March 2013).
- Yu, L., Wang, J., 2013. Robust cooperative control for multi-agent systems via distributed output regulation. *Systems & Control Letters* 62 (11), 1049 – 1056.
DOI: <http://dx.doi.org/10.1016/j.sysconle.2013.08.005>
- Zhu, J., 2014. Stabilization and synchronization for a heterogeneous multi-agent system via harmonic control. *Systems & Control Letters* 66 (0), 1 – 7.
DOI: <http://dx.doi.org/10.1016/j.sysconle.2013.12.019>