

T-World: un entorno gráfico, flexible y portable para la enseñanza e investigación de agentes inteligentes

Sergio Burdisso, Guillermo Aguirre, and Marcelo Errecalde

Universidad Nacional de San Luis, Departamento de Informática
Ejército de los Andes 950, San Luis.

`sergio.burdisso@gmail.com,gaguirre@unsl.edu.ar,merreca@unsl.edu.ar`

Resumen Los entornos controlados pueden ser muy útiles en la investigación y la enseñanza de la Inteligencia Artificial, en general, y los Agentes Inteligentes en particular. Un entorno clásico es *Tileworld*, donde es posible ajustar diferentes controles del funcionamiento del ambiente de manera tal que se pueda evaluar el desempeño de las distintas arquitecturas y estrategias de control de los agentes. Más allá del atractivo que ofrece *Tileworld* como banco de pruebas, poco interés se ha dedicado a proveer implementaciones que aprovechen todas las características y posibilidades que ofrece el mismo para la investigación y docencia en el área de Agentes Inteligentes. En este contexto, este trabajo propone y describe a *T-World*, un entorno gráfico, flexible y portable para la enseñanza e investigación de agentes inteligentes. *T-World* ofrece un entorno de trabajo visualmente atractivo e independiente de la plataforma que permite combinar distintas configuraciones de ambientes y facilita la implementación e interacción con distintos tipos de agentes.

Keywords: campos de prueba, *Tileworld*, agentes inteligentes

1. Introducción

El entorno en el que se desempeña un agente es muy importante al momento de hacer una caracterización del mismo. Al menos en una de las tantas definiciones posibles de agente [6] se resalta la importancia que tiene el entorno para explicar lo que es un agente. Es posible asegurar que el requisito fundamental impuesto a un agente es que pueda percibir su entorno y actuar sobre éste. La *percepción* de un agente está formada por aquello que reciben las entradas (sensores) en un instante de tiempo. De ese modo, se puede considerar que la *secuencia de percepciones* del agente es el historial de todo lo que el agente ha percibido. En general se puede decir que la acción elegida por el agente en cada instante depende de la secuencia de percepciones recibida hasta el momento, y caracterizar el comportamiento de un agente como una función matemática que hace corresponder a cada secuencia de percepciones una acción.

En la investigación científica, la *experimentación controlada* [2] es muy importante al evaluar el impacto que tienen en el desempeño de los sistemas tanto

las características de los programas como las variaciones del entorno. Como ejemplo, al momento de hacer la evaluación del diseño de una CPU mediante una *tarea de referencia* (*benchmark* en inglés) hay que elegirla cuidadosamente para obtener resultados de interés. Si lo que se desea medir es la velocidad de procesamiento, la multiplicación de matrices es una buena tarea de referencia ya que si se consigue un buen desempeño con problemas con multiplicación de matrices, se puede predecir que se logrará también un buen desempeño en una clase más amplia de problemas, las tareas con intensivo procesamiento numérico.

En el campo de la Inteligencia Artificial una tarea de referencia es el denominado Mundo de los Bloques (*Block World* en inglés) descrito por primera vez por Gerald Sussman como parte de su trabajo de doctorado [10]. En este problema, tres bloques (etiquetados A, B y C) están ubicados sobre una mesa. El agente debe apilar los bloques (moviendo de a un bloque por vez) de tal manera que A esté encima de B, y B encima de C.

Si el problema comienza como se ve en la Figura 1(1), vemos que alcanzar el

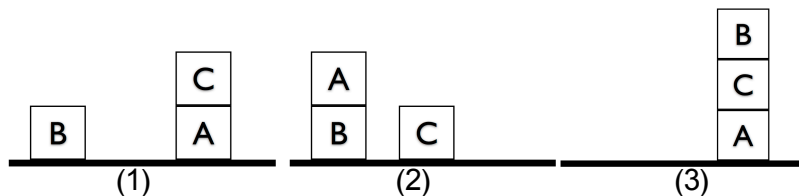


Figura 1. Mundo de bloques

objetivo no se limita sencillamente a considerar dos sub-objetivos parciales como: I) A sobre B y II) B sobre C. Si primero se consigue I) (como se muestra en la Figura 1(2)), para conseguir II) hay que deshacer lo anterior y algo similar ocurre si se comienza con el sub-objetivo II) alcanzado (ver Figura 1(3)). Este escenario simple, permite visualizar la llamada “anomalía de Sussman” que, al igual que la multiplicación de matrices, representa una clase importante de problemas como lo es la de aquellos que involucran interacciones entre sub-objetivos conjuntivos; además tiene otra propiedad atractiva: es fácil de describir.

La investigación y docencia en el área de agentes inteligentes ha debatido largamente acerca de la importancia de contar con tareas de referencia de este tipo que sirvan para analizar, experimentar y entender las particularidades que surgen de combinar distintos tipos de agentes con distintos tipos de entornos. Esto no es una tarea sencilla, si consideramos que el problema debería ser fácil de describir y entender, pero simultáneamente capaz de representar todas las dificultades que se pueden encontrar en un problema del mundo real. Por otra parte, el problema debería ser tal que permita especificar distintos tipos de dificultades que puede presentar un ambiente tales como dinamismo, incertidumbre, observabilidad parcial, etc. De esta manera, el problema no favorece su resolu-

ción con un determinado tipo de agentes, sino que permite introducir aspectos que pueden requerir habilidades reactivas con mínimo tiempo de respuesta, o pro-activas con razonamiento de alto nivel y capacidades de planificación a largo plazo.

Una tarea de referencia que parece mostrar varias de estas propiedades es el *Tileworld*. Esta tarea, ha permitido evaluar distintas arquitecturas de agentes en trabajos de investigación científica pero, lamentablemente, no ha tenido la misma difusión como herramienta didáctica y de experimentación para alumnos que realizan cursos del área de agentes inteligentes. Desde nuestra perspectiva, una de las razones para esta falencia ha sido la falta de disponibilidad de entornos gráficos amigables, interactivos, que respondan a los estándares de visualización que es usual encontrar hoy en día en juegos de computadoras. Estos entornos deberían ser portables, fáciles de programar y deberían proveer, además del soporte para la práctica docente, de todas las herramientas para la investigación científica que han sido tradicionalmente utilizadas en este problema.

En este contexto, la principal contribución de este trabajo es la presentación y descripción de *T-World*, un entorno gráfico, flexible y portable para el *TileWorld* que sirva para la enseñanza e investigación de agentes inteligentes. A tal fin, el resto del artículo se organiza de la siguiente manera. La sección 2 describe el *Tileworld* (mundo de baldosas) y en la sección 3 se describen algunas experiencias con el uso del *Tileworld* en trabajos de investigación. Nuestra implementación del *Tileworld*, denominada *T-World* se presenta en la sección 4 haciéndose hincapié en la forma en que los distintos tipos de ambientes pueden implementarse en este entorno. Finalmente, en la última sección se presentan las principales conclusiones de nuestro trabajo y posibles trabajos futuros.

2. El mundo de baldosas (*Tileworld*)

El *Tileworld* (en castellano *mundo de baldosas*) [5] se propuso inicialmente como un ambiente experimental para la evaluación de la arquitectura de los agentes. Es un entorno simulado, formado por una grilla bi-dimensional, en el cual existen agentes, huecos, baldosas y obstáculos [4]. Un agente (o robot) puede moverse en cuatro direcciones: arriba, abajo, derecha e izquierda y si está junto a una baldosa la puede empujar. Un obstáculo es un grupo inamovible de celdas de la grilla, que no pueden ser atravesadas por los agentes. Los huecos deben ser llenados con baldosas y el objetivo es llenar tantos huecos como sea posible. El *Tileworld* es un ejemplo de entorno dinámico; inicialmente, de manera aleatoria, se encuentra en uno de los estados posibles y luego, dependiendo de una serie de parámetros establecidos por el usuario, va cambiando en el tiempo de manera discreta, ya que los huecos aparecen y desaparecen aleatoriamente. El usuario puede establecer distintos parámetros como la frecuencia con que aparecen y desaparecen tanto las baldosas como los obstáculos o bien los huecos; también es posible elegir la manera en que desaparecen los huecos: brusca (completamente) o suave (decremento gradual del tamaño). En el *Tileworld* los huecos aparecen de manera aleatoria y se mantienen de acuerdo a su expectativa de vida, a menos

que desaparezcan por el accionar del usuario. El intervalo entre la aparición de huecos sucesivos es denominado el *tiempo de gestación* de los huecos. El desempeño de un agente en el *Tileworld* se mide haciendo correr este banco de pruebas durante un número predeterminado de pasos en el tiempo y midiendo la cantidad de huecos que el agente llenó de manera exitosa. El desempeño (utilidad u) de un agente en una corrida particular r se define de la siguiente manera:

$$u(r) = \frac{\text{número de huecos llenados durante } r}{\text{número de huecos que aparecieron en } r}$$

Esto brinda una medida de desempeño normalizada en el rango de 0 (el agente no tuvo éxito en llenar ningún hueco) a 1 (el agente tuvo éxito en llenar cada hueco que apareció). Más allá de su simplicidad, el *Tileworld* permite examinar importantes capacidades de los agentes. Quizás la más importante sea la habilidad del agente para reaccionar a cambios en el entorno y de explotar las oportunidades que surgen.

Como un ejemplo supongamos que un agente está llevando una baldosa hacia

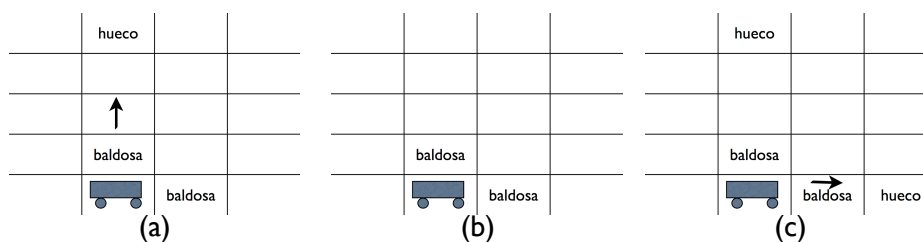


Figura 2. Consideración de oportunidades

un hueco (Figura 2 (a)), cuando de pronto el hueco desaparece (Figura 2 (b)). En ese momento el objetivo pierde sentido y lo mejor que podría hacer el agente, si notó el cambio, es repensar su objetivo original. Para ilustrar lo que significa reconocer oportunidades, supongamos que en la misma situación original aparece un hueco a la derecha del agente (Figura 2 (c)). Es más probable que el agente sea capaz de llenar el segundo hueco (antes que desaparezca) por la sencilla razón que tiene que empujar la baldosa un solo lugar, en lugar de tres. Los experimentos pueden ser “controlados” (variados) de acuerdo al: a) *Dinamismo*, la tasa en la cual aparecen nuevos huecos, b) *Hostilidad*, la tasa en la cual aparecen los obstáculos, c) *Utilidad variable*, diferencias en la puntuación de los huecos y d) *Duración fija/variable*, ¿los huecos tienen un tiempo de vida preestablecido o desaparecen gradualmente?. Estas son las principales “perillas” que se deben regular cuando se hacen los experimentos, de manera que las comparaciones son realizadas dentro de un determinado rango de variabilidad.

3. Experiencias con el uso de Tileworld

En esta sección se revisan brevemente dos trabajos de investigación basados en el uso de variantes de Tileworld. Estos trabajos son de interés porque ejemplifican la forma de realizar y evaluar los distintos experimentos, cómo las variaciones de los parámetros influyen en el desempeño de las diferentes clases de agentes y cómo se pueden visualizar gráficamente esas diferencias.

Evaluación de PRS sobre Tileworld. El trabajo [3] de Kinny y Georgeff (K&G) entre otros objetivos principales, evalúa la factibilidad de medir experimentalmente la efectividad de un agente inmerso en un ambiente simulado. K&G establecen algunos conceptos útiles que también han sido incorporados en nuestra investigación. Los huecos aparecen en celdas libres seleccionadas aleatoriamente y existen por un período de tiempo conocido como *expectativa de vida*; pero pueden desaparecer antes debido al accionar del agente. El tiempo real durante el cual existe un hueco es conocido como *tiempo de vida*. El intervalo de tiempo entre la aparición de un hueco y el siguiente es conocido como *tiempo de gestación*. Cada hueco tiene un valor específico, el *puntaje*. La expectativa de vida, el tiempo de gestación y el puntaje se toman a partir de distribuciones aleatorias independientes.

El sistema de experimentación de K&G se basó en PRS [1], un sistema de razonamiento de tiempo real usado para gobernar la tarea del robot situado en el Tileworld. Para medir el desempeño del agente en esta clase de ambientes cambiantes, es necesario establecer la *frecuencia de cambio del mundo*, que será denotada con γ . El valor de γ es la relación entre la frecuencia de reloj del Tileworld y la frecuencia de reloj usada por el agente. Este parámetro mide la dinámica de cambio percibida por el agente durante el experimento.

Para obtener resultados comparables entre distintas instancias experimentales, como medida de *efectividad del agente* se toma un valor ϵ , que se calcula como la relación entre el puntaje total de los huecos tapados y el puntaje total de los huecos que aparecieron durante el experimento. El rango de valores de ϵ es $[0;1]$, alcanzándose el máximo valor cuando el agente logra tapar todos los huecos que aparecen. A medida que γ crece, el agente siente que la expectativa de vida de los huecos disminuye, lo que conduce a una pérdida de eficiencia (ϵ disminuye). Precisamente esa relación es la escogida para graficar el comportamiento de las distintas clases de agente mediante un sistema de coordenadas donde, en el eje Y se representa el valor de ϵ y en el eje X el valor de $\log_{10}\gamma$. En esta *curva de desempeño del agente*, se pueden identificar tres secciones distintas. Para valores pequeños de γ el agente consigue muy buen desempeño ya que, como el entorno varía poco, puede cubrir todos los huecos que aparecen. A medida que γ crece la efectividad ϵ decrece rápidamente. Finalmente para los valores más grandes de γ , la curva se acerca a cero pero no llega a tocarlo debido a que siempre el agente encuentra algún hueco que puede llenar. Realizando varias repeticiones para cada conjunto de valores en los parámetros se consigue *caracterizar* a los distintos tipos de agentes.

Modelos Markov y BDI sobre Tileworld En el trabajo [8] de Simari y Parsons (S&P) se hace una evaluación empírica de los modelos BDI y MDP sobre el entorno *Tileworld*. En el modelo BDI se consideraron distintos tiempos de replanificación (parámetro p) que mide el tiempo requerido por el agente para cambiar de plan. La versión (simplificada) del *Tileworld* creada por Martijn Schut [7] fue la elegida para hacer los experimentos. En este caso, se prescinde de las baldosas ya que para sumar puntos sólo se requiere que el agente llegue al hueco; además, se considera que el agente tiene un conocimiento exacto del estado del mundo sin ningún costo de razonamiento.

En el caso del modelo llamado *Proceso de Decisión Markov* (o MDP por sus siglas en inglés), se intenta identificar en cada paso la decisión óptima y el mundo es modelado tomando en cuenta cada acción posible en cada estado posible. Claramente aquí, se visualizan los problemas de *escalabilidad* de este modelo para resolver (en forma óptima) el problema del *Tileworld*. En un mundo de tamaño n (grilla de $n \times n$), hay n^2 posiciones posibles para el agente y 2^{n^2} configuraciones posibles para la presencia (o no) de los huecos. Un *estado* en este mundo es un par (P, H) , donde $P = (i, j)$, $0 \leq i, j \leq n - 1$ y H representa una configuración posible de huecos sobre la grilla. De esa manera cuando n es igual a 6 existen $2,5 \times 10^{12}$ estados distintos, por lo que $n = 4$ o $n = 5$ son los valores más grandes que pueden ser manejados en una computadora con recursos razonables. Para enfrentar el problema de *explosión de estados* existen distintas aproximaciones las cuales han tenido distinto grado de éxito. Lo importante a considerar es que aún para tamaños pequeños del *Tileworld* existe una gran cantidad de datos para procesar.

Para comparar el desempeño de los modelos propuestos se consideraron tres parámetros: dinamismo, determinismo y accesibilidad, variándose solamente uno de ellos y dejando los otros dos constantes. Respecto al *dinamismo*, las conclusiones que alcanzan S&P son similares a las obtenidas por K&G. Para el *determinismo*, se tuvo en cuenta el efecto de la probabilidad de éxito de las acciones en la efectividad. Comprobaron empíricamente que la efectividad del agente es mejor cuando sus acciones son más confiables. Finalmente la *accesibilidad* considera la influencia del número de celdas que el agente “ve” a su alrededor. En todos los casos la efectividad aumentó cuando creció el radio de celdas consideradas.

4. Descripción de la aplicación

El diseño e implementación de *T-World* dentro de nuestro grupo de investigación, surgió por la necesidad de contar con una versión de *Tileworld* que reuniera ciertos aspectos que consideramos deseables. Por un lado, que fuese *portable* y *flexible* para ser usada fácilmente sin depender de un lenguaje de programación particular y permitiendo incorporar motores de razonamiento con *programas de agente* escritos en diferentes lenguajes. También se debía poder considerar la aplicación de diversos parámetros al mismo tiempo y, buscando que las simulaciones resulten atractivas, que también fuese *visual* pensando en las posibilidades de su uso en la enseñanza.

En la Figura 3 se observa una instantánea de la simulación de un experimento. Consideramos que es importante para usar T-World en docencia que la simulación sea similar al contexto de un juego y que se aproxime a un experimento real; por ejemplo se puede observar el efecto visual del robot empujando las baldosas. También se ve la reacción el robot cuando choca con un obstáculo.

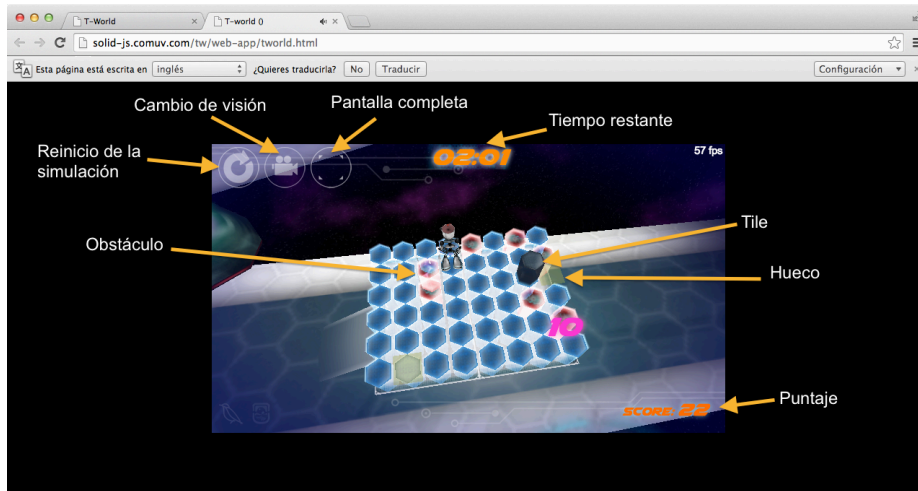


Figura 3. Ventana de la simulación con T-World

Como lenguaje de programación se usó JavaScript debido a la posibilidad de ejecutarlo dentro de un navegador. Los programas que están escritos en lo que se conoce como JavaScript *del lado del cliente*, son interpretados y ejecutados sin necesidad de contar con un servidor web. Las distintas piezas de código funcionan como manejadores (*handlers* en inglés) de eventos, ya que responden a las entradas de los usuarios siguiendo el modelo de programación orientada a eventos y de acuerdo a las instrucciones del programa producen las modificaciones en la ventana del navegador. Las acciones de los usuarios son percibidas por el navegador y el programa atiende los correspondientes eventos.

El *contenido* que se observa en las distintas páginas web tiene el estilo general provisto por el código HTML de la misma, y habitualmente también se le agregan mejoras mediante la utilización de hojas de estilo CSS que hacen aportes a la *presentación*. Al usar JavaScript se agrega *comportamiento* al contenido y la presentación de las páginas web. El rol de JavaScript es mejorar la experiencia de navegación del usuario, haciéndole más fácil obtener o transmitir información. Eso se consigue, creando efectos visuales tales como transformaciones de imágenes que sutilmente guíen al usuario y ayuden la navegación de la página. También ocultando determinado contenido en algunos casos, y mostrando detalles en otros, a medida que el usuario “escarba” en el contenido.

4.1. Configuración del entorno

Dado que en el presente artículo pondremos especial énfasis en las posibilidades que ofrece *T-World* como herramienta didáctica, describiremos la configuración del entorno tomando como base algunas de las dimensiones clásicas de los ambientes especificadas en el libro más difundido para la enseñanza de agentes inteligentes [6]: observabilidad, número de agentes, determinismo, dinámica y conocimiento del ambiente. En la mayoría de los casos, estos valores se establecen con la ayuda de elementos de control gráfico como contadores creciente/decreciente (*spinners* en inglés) o deslizadores (*sliders* en inglés).

Totalmente observable vs parcialmente observable. Los ambientes totalmente observables son más sencillos, ya que no requieren mantener información compleja sobre el estado interno del agente para conocer el estado real del ambiente. Un ambiente parcialmente observable puede deberse a la presencia de ruido en el sentido, o a la falta de sensores adecuados para la tarea bajo consideración. En *T-World* la observabilidad parcial se puede incorporar de dos formas. Una forma de observación parcial consiste en limitar el radio de celdas que pueden ser percibidas (el agente es *ciego* cuando el radio es 0). La otra forma de especificar observación parcial es introduciendo *ruido y sensores incorrectos*, los cuales se especifican como tres valores en el porcentaje de ruido correspondientes a la percepción de huecos, baldosas y obstáculos.

Único agente vs multi-agente. Como ya se ha discutido en [6], en algunos casos *puede ser conveniente* considerar una entidad cualquiera como si fuera un agente y existen otras situaciones donde *necesariamente* se deben modelizar los otros agentes. En el caso de establecerse un entorno multi-agente, las alternativas para realizarlo suelen ser de dos tipos. Una es conformar un único equipo que funciona de manera cooperativa para conseguir un mismo objetivo y la otra alternativa es simular una competencia entre dos agentes de distintos equipos. En este último caso, dos agentes compiten por sumar puntos, el progreso de un agente en el puntaje implica para el otro agente un retraso en sus posibilidades de sumar puntos, lo que corresponde a un entorno multi-agente competitivo. Cuando se forma un único equipo los agentes deben coordinar el trabajo procurando sumar el máximo puntaje en el menor tiempo posible.

Determinístico vs estocástico. Si el próximo estado se determina unívocamente por el estado actual y la acción realizada por el agente, se dice que el ambiente es *determinístico*; en otro caso que es estocástico. En un entorno determinístico y totalmente observable, el agente no debe preocuparse por la incertidumbre del entorno. Si el entorno es parcialmente observable o la realidad es tan compleja que no se pueden considerar todos los aspectos no observados puede ser conveniente asumir que el entorno es estocástico. Se dice que el entorno es *incierto* si no es totalmente observable o no es determinístico. En *T-World* cuando se especifica un comportamiento estocástico es necesario establecer cómo se comportarán las acciones del agente. Por un lado se puede regular el porcentaje de confianza

con el cual las acciones finalizarán según lo previsto; además es necesario elegir uno de los posibles modelos de movimiento estocásticos.

Estático vs dinámico. Si el ambiente puede cambiar mientras el agente está deliberando, se dice que para el agente el ambiente es dinámico; en otro caso es estático. Los ambientes estáticos son mas sencillos, ya que no se necesita estar pendiente del entorno mientras está decidiendo sobre su próxima acción, ni tener en cuenta el paso del tiempo como sí es necesario en los ambientes dinámicos. Cuando el ambiente en sí mismo no cambia con el paso del tiempo, pero sí lo hace el puntaje que mide el desempeño, entonces se dice que el entorno es semi-dinámico. En T-World es posible optar por cualquiera de las tres posibilidades.

Conocido vs Desconocido. Hablando estrictamente, esta distinción no se refiere al ambiente en sí mismo sino al conocimiento que puede tener el agente (o el diseñador del agente) sobre las leyes de la física del ambiente, es decir, sobre los resultados de las operaciones (o las probabilidades de los resultados, si el entorno es estocástico). Notar que la distinción entre entornos conocidos y desconocidos no es la misma que entre ambientes total y parcialmente observables. En cualquier caso, en T-World no existe ningún impedimento para experimentar con agentes cuando el ambiente es conocido o desconocido.

4.2. Programa Agente

Dentro de *T-World* todos los agentes perciben el entorno del mismo modo, mediante una estructura en formato JSON, XML o hechos Prolog¹, con tres tipos de información: *externa al agente* (p.e. disposición de elementos “físicos” y de otros agentes en el mundo, tiempo transcurrido, etc.). Información *relativa al agente mismo* como ser niveles de energía, puntuación, posición actual, etc. *Conocimiento incorporado* con aquellos valores constantes asignados por el usuario, (p.e. probabilidades de resultados, dimensiones del escenario, valores y reglas que definen cómo evoluciona el mundo). Las acciones del agente también deben ser transmitidas al T-World mediante paquetes JSON.

De ese modo los usuarios sólo deben programar el motor de razonamiento, lo que se denomina el *programa agente*, ya que la percepción (y las acciones) se obtienen (transmiten) mediante paquetes JSON. El envío y recepción de paquetes JSON por parte del navegador sólo es posible mediante el uso de WebSockets (provistas por HTML5) que utiliza un protocolo propio (protocolo WebSocket RFC6455). Por este motivo fue necesario construir un Proxy en lenguaje C que utiliza sólo librerías estándares, de forma tal que pueda ser compilado de forma nativa en cualquier sistema operativo. Luego el usuario deberá conectar su programa agente con el Proxy, mediante un simple socket TCP.

¹ los formatos JSON y XML son estándares para el intercambio de datos usados por muchos lenguajes. Además, debido a la popularidad de PROLOG dentro del campo de la Inteligencia Artificial se lo incluyó especialmente

5. Conclusiones y trabajos futuros

Tomando en cuenta el importante papel que desempeñan los bancos de prueba para evaluar distintas estrategias de agentes inteligentes y reconociendo la importancia que tiene en educación el uso de la experimentación controlada, este trabajo propone y describe el entorno *T-World*. Esta versión de Tileworld se destaca por su portabilidad ya que puede ser accedida como una página web, por su flexibilidad permitiendo combinar distintas situaciones a las cuales enfrentar a los agentes y provista de una atractiva visualización que estimula su uso. *T-World* se puede acceder de manera remota como un URL en la dirección <http://tworld-ai.com> o bien solicitar a los autores el código y cargarlo como un archivo local para ejecutarlo en cualquier navegador.

Actualmente se están implementando mejoras que permiten la administración de usuarios y de los programas agente que los usuarios desarrollan; también está prevista la generación y resguardo de las distintas estadísticas que los mismos generan. Entre los usos inmediatos se realizarán pruebas para integrar *T-World* en diferentes líneas de trabajo dentro del grupo de investigación [9] y en docencia se considera su uso en la materia *Inteligencia Artificial* de la Lic en cs. de la Computación de la UNSL.

Referencias

1. Michael P Georgeff and Francois Felix Ingrand. Decision-making in an embedded reasoning system. In *Proceedings of the 11th international joint conference on Artificial intelligence-Volume 2*, pages 972–978. MKP Inc., 1989.
2. Steve Hanks, Martha E. Pollack, and Paul R. Cohen. Benchmarks, test beds, controlled experimentation, and the design of agent architectures. *AI Magazine*, 14(4):17–42, 1993.
3. D Kinny and M George. Commitment and effectiveness of situated agents. In *IJCAI-91*, pages 82–88, 1991.
4. Michael Lees. A history of the tileworld agent testbed. Technical report, School of Computer Science and Inf. Tech, University of Nottingham. UK, 2002.
5. Martha E. Pollack and Marc Ringuette. Introducing the tileworld: Experimentally evaluating agent architectures. In *In Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 183–189, 1990.
6. Stuart J. Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach (3. internat. ed.)*. Pearson Education, 2010.
7. Martijn Schut and Michael Wooldridge. Intention reconsideration in complex environments. In *Proceedings of the fourth international conference on Autonomous agents*, pages 209–216. ACM, 2000.
8. Gerardo I Simari and Simon D Parsons. *Markov Decision Processes and the Belief-Desire-Intention Model: Bridging the Gap for Autonomous Agents*. Springer, 2011.
9. Cecilia Sosa-Toranzo, Marcelo Errecalde, and Edgardo Ferretti. On the use of agreement technologies for multi-criteria decision making within a BDI agent. In *14th Ibero-American Conference on Artificial Intelligence (IBERAMIA)*, November 2014. Accepted as full paper.
10. Gerald Jay Sussman. *A Computer Model of Skill Acquisition*. PhD thesis, Laboratorio de Inteligencia Artificial, MIT, 1973.