

A Practical Approach for Transforming Conceptual Design into a Computational Model in SIMIO

Juan L. Sarli¹ and Maria J. Blas²

¹ Universidad Tecnológica Nacional-Facultad Regional Santa Fe, Santa Fe, Argentina

² Instituto de Desarrollo y Diseño INGAR (UTN-CONICET), Santa Fe, Argentina
jsarli@frsf.utn.edu.ar, mariajuliablas@santafe-conicet.gov.ar

Abstract. Drawing on Robinson's (2014) framework, this work introduces a methodology to establish traceability between the design model and the computational implementation within the SIMIO simulator. The proposal addresses the inherent gap between conceptual modeling theory and the technical execution of simulation models in commercial-off-the-shelf (COTS) software. By leveraging SIMIO's object-oriented paradigm, the approach maps conceptual components through ad-hoc traceability guidelines designed to ensure the structural and logical integrity of simulation artifacts.

Keywords: Discrete-Event Simulation, Model Traceability.

Enfoque Práctico para Transformar el Diseño Conceptual en un Modelo Computacional en SIMIO

Resumen. Basado en Robinson (2014), se propone una metodología para establecer trazabilidad entre el modelo de diseño y el computacional implementado en el simulador SIMIO. La propuesta es el puntapié inicial para abordar la brecha entre la teoría de modelado conceptual y la implementación de modelos de simulación basados en software comercial. El enfoque utiliza las capacidades de orientación a objetos de SIMIO para mapear componentes conceptuales, por medio de la definición de lineamientos de trazabilidad ad-hoc para asegurar la integridad de los artefactos de simulación.

Palabras clave: Simulación de Eventos Discretos, Trazabilidad de Modelos.

1 Introducción

Según Robinson (2014), en Modelado y Simulación (M&S) basado en eventos discretos, la actividad de modelado conceptual consiste en la definición de una abstracción no basada en software de un sistema real, definiendo objetivos, componentes y simplificaciones antes de la simulación. La construcción de un modelo de simulación se define en base a un flujo de artefactos evolutivos: descripción del sistema, modelo conceptual, modelo de diseño y, finalmente, el modelo computacional.

En este trabajo, se presenta una propuesta para brindar trazabilidad entre los artefactos modelo de diseño y modelo computacional, en base a una implementación en SIMIO¹. La motivación de la metodología de trazabilidad surge de la necesidad de mapear el diseño de modelos de simulación teóricos (basados en conceptos y relaciones bien definidas) a herramientas de software comerciales (las cuales no siempre brindan soporte a las nociones fundamentales de M&S, sino más bien aportan a su ejecución y a la visualización de resultados). Este enfoque constituye los pasos iniciales para abordar una problemática de mapeo teórico-práctico, aplicando herramientas existentes para resolver un desafío no trivial y novedoso en el modelado conceptual. Se toma SIMIO como herramienta de aplicación debido a sus nociones de orientación a objetos, facilidad de extensión de conceptos y encapsulamiento de responsabilidades (Prochaska et al., 2017). Adicionalmente, el lugar de trabajo donde se lleva a cabo este estudio posee licencia académica para dicha herramienta; lo que garantiza un contexto adecuado para la conducción de esta investigación.

El resto del trabajo se encuentra estructurado de la siguiente manera. La Sección 2 presenta los principales elementos utilizados durante el mapeo junto con la metodología definida. La Sección 3 describe una prueba de concepto. Finalmente, la Sección 4 se encuentra dedicada a la discusión de resultados parciales y trabajos futuros.

2 Metodología y Elementos de Mapeo

La Tabla 1 presenta los elementos de los artefactos vinculados a modelado conceptual según se detalla en (Álvarez et al., 2024) y (Blas et al., 2025) siguiendo la propuesta de (Robinson, 2014). Para cada caso, se detalla el artefacto en el cual se define el elemento (modelo conceptual -MC- o de diseño -MD-).

Tabla 1. Elementos del Modelado Conceptual.

Elemento	Artef.	Descripción
<i>Entrada</i>	MC	Información necesaria para ejecutar el modelo, pudiendo corresponder a factores experimentales o parámetros. Deben referir a algún elemento del modelo.
<i>Salida</i>	MC	Información vinculada al objetivo que se obtiene al ejecutar el modelo. Cada salida debe asociarse con una propiedad de un elemento del modelo.
<i>Entidad</i>	MD	Elemento principal del modelo, caracterizado por sus atributos. Se relaciona con las demás entidades por medio de eventos.
<i>Atributo</i>	MD	Propiedad tipificada que caracteriza una entidad, la cual toma valores durante la ejecución del modelo.
<i>Evento</i>	MD	Ocurrencia instantánea que puede alterar el estado del modelo, dependiendo de su efecto. Un evento es primario si su efecto modifica el estado del modelo, en otro caso se lo considera secundario.
<i>Rutina de Evento</i>	MD	Actualiza el estado del modelo cuando se dan los eventos primarios. Las rutinas de eventos primarios embeben los efectos de los eventos secundarios asociados.

¹ <https://www.simio.com/>

Es importante destacar que la Tabla 1 solo presenta los elementos asociados a artefactos (previos al desarrollo del modelo computacional) trazables por la metodología. Los restantes elementos del modelo conceptual (como ser, simplificaciones, suposiciones, objetivo, alcance del modelo, y nivel de detalle) no se incluyen ya que sientan las bases para el modelo de diseño. Por otra parte, los elementos individualmente identificados como actividades y demoras de una entidad son excluidos debido a que su definición guía la especificación de eventos y rutinas de eventos. Estos últimos dos (incluidos en Tabla 1), definen la componente dinámica de cada entidad, por lo que son trazables al flujo de ejecución computacional.

Por su parte, la Tabla 2 presenta los elementos principales de SIMIO según se detalla en (Prochaska et al., 2017) y (Houck et al., 2019).

En base a lo definido con anterioridad, se formularon las reglas de mapeo que se detallan en la Tabla 3.

Tabla 2. Elementos disponibles en SIMIO para la construcción de modelo computacional.

Elemento	Descripción
<i>Model Entity</i>	Objeto que representa una entidad dinámica que se mueve y es procesada en el modelo. Es un modelo completo en sí mismo. Tiene inteligencia y comportamiento definido mediante estados y atributos. Posee una lógica interna propia a través de los eventos y procesos.
<i>Fixed Object</i>	Elemento con ubicación permanente dentro de la vista de <i>Facility</i> , definido dentro de la <i>Standard Library</i> . Se encarga de procesar las entidades. Tiene su propia lógica interna con estados, procesos y eventos asociados.
<i>Property</i>	Característica de <i>Model Entity</i> o <i>Fixed Object</i> que representa un atributo estático del objeto. Se define en tiempo de diseño como entrada del modelo y se utiliza como un valor de solo lectura durante la ejecución (por ejemplo: para configurar el escenario).
<i>State</i>	Característica de <i>Model Entity</i> o <i>Fixed Object</i> que representa un atributo del objeto que evoluciona en el tiempo. Varía de forma discreta o continua a lo largo de la ejecución.
<i>Link</i>	Camino por el cual una entidad se mueve entre objetos dentro del modelo. Es una especialización del elemento <i>Fixed Object</i> .
<i>Node</i>	Punto de inicio o fin de un <i>Link</i> . Es una especialización del elemento <i>Fixed Object</i> .
<i>Trigger</i>	Punto de entrada de un <i>Fixed Object</i> para invocar un <i>Process</i>
<i>Output Statistic</i>	Variable de salida definida por el usuario.
<i>Process</i>	Secuencia de pasos activada por un evento. Al iniciar libera un token como hilo de ejecución. Los tokens se pueden extender por medio de los elementos <i>Property</i> y <i>State</i> .

Tabla 3. Reglas de mapeo.

#	Descripción
1	Toda <i>Entrada</i> define una <i>Property</i> .
2	Una <i>Salida</i> define un <i>State</i> si es atómica, como un contador estadístico.
3	Una <i>Salida</i> define un <i>Output Statistic</i> si requiere de una fórmula.
4	Una <i>Entidad</i> define un <i>Model Entity</i> si es procesada por el modelo, pero no conforma su lógica.

5	Una <i>Entidad</i> define un <i>Fixed Object</i> si procesa otras entidades y es parte de la lógica del modelo (entidad controlable por el modelo).
6	Una <i>Entidad</i> define un <i>Link</i> si refiere a características del camino que debe seguir un <i>Model Entity</i> .
7	Todo <i>Atributo</i> define un <i>State</i> .
8	Todo <i>Evento</i> define un <i>Trigger</i> dentro de un <i>Fixed Object</i> .
9	Toda <i>Rutina de Evento</i> que son primarios define un <i>Process</i> .

3 Prueba de Concepto

Se plantea un escenario de simulación reducido que tiene por objetivo estimar la cantidad de vehículos por tipo fabricados (*Salida #1*) y la cantidad de vehículos en cola en un registro de patentes (*Salida #2*). El patentamiento aplica a tres tipos de vehículos (*Entidad #1*): moto, auto, o camión (*Atributo #1*). El país puede ser Argentina o Brasil (*Atributo #2*), según su fabricación. Los vehículos arriban (*Entidad #2*) con una distribución uniforme 60-120 minutos (*Entrada #1*), luego van a un proceso de registro de patente (*Entidad #3*) con una distribución triangular <60, 100 y 140> minutos (*Entrada #2*). Los vehículos ya patentados abandonan la fábrica (*Entidad #4*). Cada vez que se produce un arribo de vehículo (*Evento #1*) y un ingreso a la cola de la *Entidad #3* (*Evento #2*), se dispara la rutina de evento *definirTipoVehiculoypais* (*Rutina de Evento #1*) y *contarTipoVehiculoArribadoARegistro* (*Rutina de Evento #2*).

Las Figuras 1 a 3 detallan la implementación de este escenario en SIMIO.

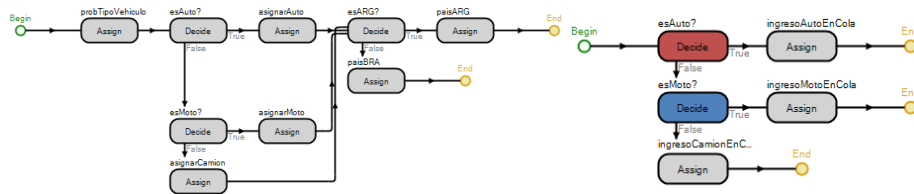


Fig. 1. (a) Izquierda: *Proceso* que implementa la *Rutina de Evento #1* para el tipo de vehículo y país. Los pasos *asignarAuto*, *asignarMoto*, y *asignarCamion* setean el valor del *Atributo #1* y actualizan el valor de la *Salida #1*. Los pasos *paisARG* y *paisBRA* setean el valor del *Atributo #2*. **(b)** Derecha: *Proceso* que implementa la *Rutina de Evento #2* para contar tipo de vehículo arribado a registro de patente. Los pasos denominados “*ingreso*” actualizan la *Salida #2*.

Views	Name	Object Type	Description
Elements	AutosArribados	Output Statistic Element	Cantidad de autos que arribaron a patentamiento
	MotosArribadas	Output Statistic Element	Cantidad de motos que arribaron a patentamiento
	CamionesArribados	Output Statistic Element	Cantidad de camiones que arribaron a patentamiento
Properties	AutosEnEspera	State Statistic Element	Cantidad de autos en espera de patentamiento
	MotosEnEspera	State Statistic Element	Cantidad de motos en espera de patentamiento
	CamionesEnEspera	State Statistic Element	Cantidad de camiones en espera de patentamiento

Fig. 2. Métricas de salida, donde *AutosArribados*, *MotosArribadas*, *CamionesArribados* definen la *Salida #1*. *AutosEnEspera*, *MotosEnEspera*, *CamionesEnEspera* definen la *Salida #2*.

Alvarez, G., Sarli, J., Blas, M. J. (2023). Una Propuesta de Extensión de un Framework para el Desarrollo de Modelos Conceptuales para Simulación. Actas del 2023 Congreso Nacional de Ingeniería Informática / Sistemas de Información (CONAIISI).

Blas, M. J., Meichtry Regner V. E., Gonnet, S. (2025). Design of Discrete Event Simulations. Simposio de Modelado y Simulación (SIMS).

Houck, D., Whitehead, C. (2019). INTRODUCTION TO SIMIO. Proceedings of the 2019 Winter Simulation Conference. DOI: 10.1109/WSC40007.2019.9004692

Prochaska, K., Thiesing, R. M. (2017). Introduction to Simio. Proceedings of the 2017 Winter Simulation Conference. DOI: 10.1109/WSC.2017.8248147

Robinson, S. (2014). Simulation: The Practice of Model Development and Use. 2nd ed. Palgrave MacMillan.