

CONESCAPANHONDURAS2025paper131.pdf

 Institute of Electrical and Electronics Engineers (IEEE)

Document Details

Submission ID

trn:oid:::14348:477759018

Submission Date

Jul 31, 2025, 10:10 PM CST

Download Date

Aug 12, 2025, 6:29 PM CST

File Name

CONESCAPANHONDURAS2025paper131.pdf

File Size

138.4 KB

4 Pages




2,580 Words

16,014 Characters

3% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Top Sources

- 3%  Internet sources
- 3%  Publications
- 0%  Submitted works (Student Papers)

Integrity Flags




0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Top Sources

- 3%  Internet sources
- 3%  Publications
- 0%  Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Publication	R. Mroß, A. Schnakenbeck, M. Völker, A. Fay, S. Kowalewski. "Model Checking Gen...	1%
2	Publication	S. Ramaswamy, K.P. Valavanis. "Hierarchical time-extended Petri nets (H-EPNs) ba...	<1%
3	Internet	waset.org	<1%
4	Publication	Ngoc Huy Tran, Xuan Thuyen Le. "Building Augmented Reality Integrated Autom...	<1%
5	Internet	www.coursehero.com	<1%

GRAFCETJSON: Formato JSON para la representación de diagramas secuenciales GRAFCET

Abstract—This project presents GRAFCETJSON, a structured JSON format designed to digitally represent GRAFCET diagrams in accordance with the IEC 60848 standard. The format was developed to enable precise encoding of sequential structures, transitions, actions, timers, and conditions specific to the GRAFCET language. As part of the proposed system, a graphical interface was implemented using the GoJS library, modified to interpret and simulate diagrams from GRAFCETJSON files. This solution facilitates the visualization, editing, and validation of industrial processes, thus providing a formal and extensible foundation for the graphical representation of control systems, aimed at improving interoperability and efficiency in the design of automation systems.

Index Terms—GRAFCET, GRAFCETJSON, automation, sequential control, industrial processes, simulation, GoJS

I. INTRODUCCIÓN

La automatización industrial se ha consolidado como un elemento clave en la mejora de la eficiencia operativa, la seguridad y la competitividad de los sistemas productivos modernos. Dentro de este campo, la representación formal de la lógica de control secuencial es esencial para garantizar la trazabilidad, el mantenimiento y la implementación precisa de procesos automatizados. GRAFCET (*Graphe Fonctionnel de Commande Étape Transition*), normalizado por la IEC 60848, es uno de los lenguajes gráficos utilizados para esta tarea, debido a su capacidad para describir, de forma clara y estructurada, el comportamiento dinámico de sistemas secuenciales [1], [2].

A. Antecedentes

GRAFCET fue introducido en la década de 1970 como una evolución de metodologías para la especificación de automatismos, y desde entonces ha sido formalizado y utilizado como lenguaje gráfico para representar procesos de control secuencial de manera visual y normalizada, especialmente en el ámbito educativo y en entornos de simulación y documentación técnica [1], [3]. Su estructura, basada en etapas, transiciones, condiciones y acciones, permite establecer una correspondencia lógica entre el diseño conceptual y su implementación en controladores lógicos programables (PLCs). A lo largo de los años, diversas herramientas han sido desarrolladas para facilitar su edición y simulación.

Una de las herramientas más conocidas es CADE-SIMU, un software que comenzó a utilizarse a principios de la década del 2000 y que ha sido ampliamente adoptado, particularmente en ámbitos de enseñanza técnica y formación académica. Si

bien su simplicidad ha sido una ventaja para la enseñanza del lenguaje GRAFCET, su evolución tecnológica ha sido limitada. Carece de soporte para exportación estructurada de diagramas, interoperabilidad con otras plataformas digitales o adaptación a entornos de desarrollo modernos. Esta situación ha generado la necesidad de contar con soluciones que permitan representar diagramas GRAFCET de manera estructurada y compatible con herramientas gráficas más flexibles y actuales.

B. Problemática

Aunque GRAFCET ha sido reconocido por normativas internacionales como un lenguaje formal útil para la representación de procesos secuenciales, su implementación práctica en entornos industriales ha sido limitada, en parte debido a la falta de herramientas que permitan integrarlo de manera flexible en flujos de trabajo digitales. En la actualidad, no se dispone de un formato abierto, estandarizado y estructurado que posibilite la representación de diagramas GRAFCET en un formato interpretable por sistemas computacionales, ni que sea fácilmente integrable en entornos de edición, simulación o validación automática. La mayoría de herramientas disponibles operan con esquemas gráficos cerrados o exportaciones poco reutilizables, lo que impide su incorporación en procesos de diseño interoperables.

La necesidad de contar con una representación formal, digital y extensible de diagramas GRAFCET ha motivado el desarrollo de GRAFCETJSON, un formato en JSON propuesto en este trabajo, orientado a la codificación estructurada de diagramas de control secuencial. Este formato permite representar etapas, transiciones, condiciones y acciones conforme a las reglas de GRAFCET, y fue diseñado para ser interpretado por interfaces visuales como GoJS, con fines de edición, visualización y simulación interactiva. La propuesta busca superar las limitaciones estructurales de herramientas existentes, promover la interoperabilidad con sistemas modernos y facilitar la validación operativa de procesos automatizados.

En este contexto, el presente trabajo propone GRAFCETJSON, un formato estructurado basado en JSON, diseñado específicamente para representar diagramas GRAFCET en entornos digitales. Este formato permite codificar etapas, transiciones, condiciones, acciones y temporizadores de manera formal, estandarizada y reutilizable. Además, fue implementado y validado mediante una herramienta desarrollada en la librería GoJS, adaptada para interpretar y visualizar los

diagramas definidos bajo este esquema. La propuesta busca no solo superar las limitaciones de las herramientas actuales, sino también establecer una base interoperable para el diseño, edición y simulación de sistemas de control secuencial.

II. METODOLOGÍA

Para llevar a cabo el diseño de sistemas automatizados que posean un nivel de complejidad suficiente como para ser representados de forma estructurada mediante sus elementos fundamentales, es necesario contar con herramientas adecuadas tanto para su construcción como para su evaluación.

A. Formato JSON

Si bien la solución propuesta por Northwoods Software proporciona una base sólida, su utilidad se ve limitada por la cantidad de elementos que puede representar y por los métodos disponibles para evaluar los diseños generados.

En respuesta a estas limitaciones, este trabajo propone una extensión basada en la solución de Northwoods Software y en el uso de su biblioteca GoJS. Se desarrolla un formato en JSON junto con una herramienta de software propia, diseñada específicamente para cumplir con los objetivos de esta investigación. Esta herramienta busca ofrecer un entorno independiente de diseño que proporcione al usuario todas las capacidades necesarias para construir y evaluar GRAFCETs. Para lograr este objetivo, fue necesario expandir la plantilla de GRAFCET incluida en GoJS, incorporando nuevas funcionalidades que permitieran una mayor expresividad y control. A partir de la estructura resultante, se diseñó también un algoritmo de simulación capaz de evaluar el comportamiento del sistema modelado, lo cual permite verificar la validez del diseño y proporciona retroalimentación útil para el diseñador humano.

B. GRAFCET de Nivel Tecnológico

La API de GoJS proporciona documentación detallada que permite tanto la edición de plantillas existentes como la creación de diagramas completamente nuevos. A partir de esta API, se desarrolló una plantilla especializada para representar GRAFCETs a nivel tecnológico, superando así la representación meramente funcional que ya era posible con la plantilla base.

Extender la plantilla para modelar el nivel tecnológico implicó integrar un sistema de variables, junto con un modelo de acción y otro de receptividad o condición. El sistema de variables fue incorporado directamente dentro de los parámetros del modelo, utilizando la clave `modelData` como punto de almacenamiento. Dentro de esta clave, cada variable es representada como un objeto de JavaScript, al que se le asignan distintos parámetros necesarios tanto para el diseño como para la simulación del sistema.

Cada una de estas variables posee un nombre único, compuesto únicamente por caracteres alfanuméricos por motivos de simplicidad y compatibilidad visual. Además del nombre, es fundamental especificar el tipo de variable, diferenciando entre entradas, salidas y contadores, con el objetivo de evitar

errores durante su manipulación lógica o computacional. Para que el sistema de simulación funcione de forma eficiente, también se incorporaron parámetros adicionales que permiten evaluar su estado, gestionar temporizadores asociados o determinar cómo cambian en el tiempo. Todos estos parámetros son modificables durante el proceso de simulación, lo que permite que el diseñador interactúe de manera dinámica con las variables y aproveche su complejidad para obtener un diseño más realista.

Gracias a esta infraestructura, las acciones dentro del diseño pueden aprovechar directamente las funcionalidades del sistema de variables, lo que permite asignar tiempos, operar contadores y manipular variables binarias de forma dinámica o persistente, según lo establecido por la normativa de GRAFCET. Esta integración facilita la configuración de comportamientos fundamentales del sistema automatizado dentro del entorno gráfico.

En cuanto al modelo de receptividad o condición, se contemplaron dos enfoques complementarios. Por un lado, se permite al usuario introducir condiciones utilizando texto plano, lo cual brinda una interfaz sencilla pero poderosa para manejar expresiones lógicas relativamente complejas, incluyendo la negación de variables, evaluación de flancos y agrupación de términos (por ejemplo, expresiones como $(S1+S2) * -S3$). Por otro lado, se desarrolló una alternativa basada en estructuras tipo JSON, lo que permite una implementación visual más clara y estructurada por parte de GoJS, a la vez que facilita la interpretación automática de las condiciones durante la simulación.

C. Simulación

Para llevar a cabo la simulación del modelo, se optó por implementar un algoritmo basado en eventos, ya que este enfoque permite garantizar un buen rendimiento incluso en sistemas complejos, al mismo tiempo que ofrece escalabilidad a medida que crece el número de componentes o transiciones dentro del diseño. Este tipo de algoritmo resulta especialmente adecuado para representar sistemas secuenciales donde las transiciones y acciones se activan por interacción del usuario, evitando así el uso ineficiente de recursos que puede generar un enfoque cíclico o por pasos fijos.

El algoritmo utilizado está basado en el algoritmo propuesto en [4], el cual ha demostrado ser efectivo para representar el comportamiento de autómatas secuenciales y sistemas de control discretos bajo un esquema de eventos. Este permite evaluar de forma precisa la evolución del sistema a partir de las etapas activas y las condiciones dinámicas asociadas a cada transición, con una respuesta eficiente tanto en tiempo de diseño como durante la ejecución simulada.

Como se muestra en Fig. 1, el cálculo de las transiciones franqueables se realiza únicamente en tres momentos clave durante la simulación: al iniciar la simulación, inmediatamente después de ejecutar las acciones correspondientes a las etapas activas, y tras que se dé algún evento asíncrono, como, por ejemplo, que el usuario active una entrada o se efectúe una acción temporizada. Esta estrategia permite reducir el número

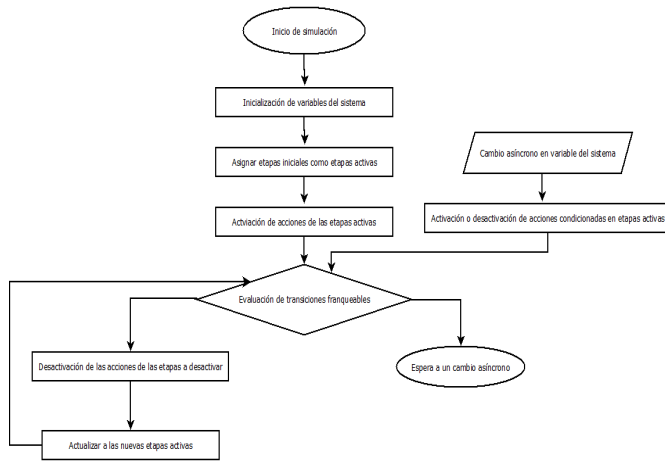


Fig. 1. Flujograma de la simulación de sistemas secuenciales bajo un esquema de eventos.

de evaluaciones innecesarias y mejora la eficiencia general del sistema de simulación.

Durante la evaluación de las transiciones franqueables, el procedimiento consiste en identificar aquellas que se encuentran conectadas a las etapas actualmente activas, analizar sus condiciones de receptividad y determinar su posibilidad de evolución conforme a las reglas del algoritmo. No obstante, debido a la variedad de estructuras posibles en un GRAFCET, como secuencias complejas, condiciones ubicadas en distintas posiciones o estructuras anidadas, se vuelve necesario implementar un mecanismo de exploración que permita recorrer y analizar adecuadamente estas configuraciones.

La implementación de este recorrido se ve facilitada por las capacidades que ofrece la librería GoJS, la cual permite realizar iteraciones diferenciadas según el tipo y dirección del enlace. Esto brinda la posibilidad de avanzar a través del grafo respetando las reglas de evolución del GRAFCET, identificando correctamente las ramificaciones, convergencias y demás elementos propios del modelo secuencial. También, haciendo uso de funciones recursivas, es posible crear un algoritmo compacto que abarca todas las estructuras posibles. Gracias a estas herramientas, el sistema logra mantener una ejecución precisa y eficiente, incluso en configuraciones complejas.

III. RESULTADOS

A. Desarrollo del Formato GRAFCETJSON

El formato GRAFCETJSON fue desarrollado como una solución estructurada para representar diagramas GRAFCET mediante notación JSON, extendiendo significativamente las capacidades de las herramientas existentes. La implementación se basa en una arquitectura de metadata que permite modelar tanto GRAFCETs funcionales como tecnológicos a través de estructuras de datos optimizadas.

La Tabla I presenta una comparación detallada entre GRAFCETJSON y las alternativas existentes, evidenciando las mejoras significativas implementadas. Las principales ventajas

incluyen capacidades de simulación nativas, soporte completo para operaciones lógicas complejas, y optimización para sistemas industriales de gran escala.

TABLE I
COMPARACIÓN DE CARACTERÍSTICAS ENTRE HERRAMIENTAS GRAFCET

Característica	GoJS Base	CADe SIMU	GRAFCET JSON
Simulación nativa	No	Sí	Sí
Operaciones lógicas	No	Completas	Completas
Temporizadores	No	Sí	Sí
Contadores	No	No	Sí
Acciones múltiples	No	Sí	Sí
Acciones condicionadas	No	Sí	Sí
Optimización grandes	N/A	Deficiente	Excelente
Cuadrícula diseño	No	Sí	Sí
Aplicación industrial	Limitada	Educativa	Profesional

B. Estructura del Formato JSON

La Tabla II ilustra los componentes principales de la estructura GRAFCETJSON mediante un ejemplo práctico de 4 etapas que incluye contadores, temporizadores y acciones condicionadas. La arquitectura se fundamenta en tres secciones principales: metadata del modelo, definición de nodos (etapas) y enlaces (transiciones).

TABLE II
ESTRUCTURA PRINCIPAL DEL FORMATO GRAFCETJSON

Componente	Descripción	Ejemplo
grafcetLevel	Nivel del modelo	"technologic"
systemVariables	Variables del sistema	{ "variable": "S1", "type": "input", "value": "0" }
nodeDataArray	Etapas y acciones	{ "step": "1", "actions": [...], "text": "Iniciar contador" }
linkDataArray	Transiciones y condiciones	{ "condition": [{ "text": "START", "type": "RE" }] }
condition	Expresiones booleanas	[{ "text": "S2", "type": "" }]
actions	Acciones por etapa	{ "variable": "T1", "timing": "delay", "timernum": "2000" }

C. Validación y Rendimiento

La validación del sistema se realizó mediante el diseño y simulación de sistemas industriales representativos, incluyendo líneas embotelladora, sistemas de autolavado, procesos de horneado industrial y sistemas de embotellado. Los resultados demuestran que GRAFCETJSON mantiene fluidez operacional en modelos complejos de hasta 30 etapas sin degradación del rendimiento, superando significativamente las limitaciones de CADesimu en sistemas de gran escala.

Las pruebas comparativas evidencian que mientras CADesimu experimenta degradación progresiva del rendimiento y eventual bloqueo en GRAFCET complejos, GRAFCETJSON mantiene responsividad constante tanto en las operaciones de diseño como de simulación. Esta mejora se atribuye a la

optimización del motor de eventos discretos y la estructura eficiente de representación JSON.

D. Motor de Simulación

La implementación del motor de simulación basado en eventos discretos permite la evaluación simultánea de múltiples GRAFCET, soportando estructuras concurrentes y condiciones complejas. El sistema integra algoritmos de parsing para expresiones booleanas, manejo de temporizadores con precisión de milisegundos, y operaciones de contador con persistencia de estado.

La Fig. 2 muestra la arquitectura del sistema GRAFCETJSON Viewer, desarrollado sobre el framework Electron con licencia académica de Northwoods Software para GoJS. La herramienta proporciona compatibilidad multiplataforma y capacidades de serialización/deserialización automática de modelos.

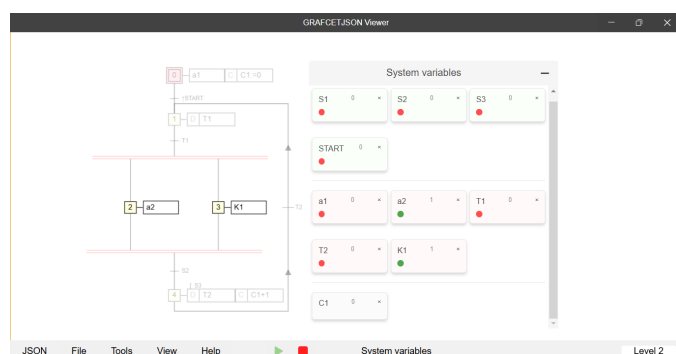


Fig. 2. Arquitectura del sistema GRAFCETJSON Viewer.

Los resultados obtenidos demuestran que el formato GRAFCETJSON constituye una solución robusta para la representación y simulación de sistemas GRAFCET industriales, superando las limitaciones técnicas de las herramientas existentes y proporcionando una base sólida para aplicaciones profesionales de automatización industrial.

IV. CONCLUSIONES

Este trabajo ha presentado GRAFCETJSON, un formato estructurado en JSON que permite la representación digital completa de diagramas GRAFCET conforme a la norma IEC 60848. La solución desarrollada extiende significativamente las capacidades de las herramientas existentes mediante la implementación de una arquitectura optimizada que soporta tanto GRAFCETs funcionales como tecnológicos.

Las principales contribuciones incluyen el desarrollo de un formato abierto y estandarizado que facilita la interoperabilidad entre sistemas, la implementación de un motor de simulación basado en eventos discretos que mantiene rendimiento constante en sistemas complejos, y la creación de una herramienta gráfica profesional que supera las limitaciones de soluciones educativas como CADe-SIMU.

Los resultados de validación demuestran que GRAFCETJSON constituye una solución robusta para aplicaciones industriales, proporcionando una base formal y extensible para el

diseño, edición y simulación de sistemas de control secuencial. La herramienta desarrollada representa un avance significativo hacia la modernización de las metodologías de automatización industrial.

ACKNOWLEDGMENT

Los autores agradecen a Northwoods Software por licenciar la librería GoJS para su uso académico, lo cual hizo posible el desarrollo de esta investigación.

REFERENCES

- [1] IEC (International Electrotechnical Commission), "IEC 60848:2013 Specification language GRAFCET for sequential function charts," 2013. [Online]. Available: <https://webstore.iec.ch/publication/6023>
- [2] N. P. Mahalik, *Fieldbus Technology: Industrial Network Standards for Real-Time Distributed Control*. Berlin: Springer-Verlag, 2013.
- [3] R. David and H. Alla, *Petri Nets and Grafcet: Tools for modelling discrete event systems*. Prentice Hall, 1992.
- [4] R. Mroß, A. Schnakenbeck, M. Völker, A. Fay, y S. Kowalewski, «Unambiguous Interpretation of IEC 60848 GRAFCET based on a Literature Review», 2022 IEEE 27th International Conference On Emerging Technologies And Factory Automation (ETFA), pp. 1-8, sep. 2023, doi: 10.1109/etfa54631.2023.10275504.