

A planning algorithm in a class of discrete event systems

Guelvis Mata ^a, Bladimir Ruiz ^a, Claudia Camacho ^b, Arnaldo Méndez ^c, Sergio Muñoz ^d & Hicher Zambrano ^e.

^a Facultad de Ingeniería, Universidad Nacional de Chimborazo, Riobamba, Ecuador. & Departamento de Matemáticas, Facultad de Ciencias, Universidad de Los Andes, Mérida, Venezuela. gemata@unach.edu.ec, lbrui@unach.edu.ec

^b Universidad Politécnica Territorial de Mérida, "Kléber Ramírez", Venezuela. claucamacho11@gmail.com

^c Dpto. de Física y Matemáticas, Facultad de Ingeniería, Universidad Nacional Experimental Francisco de Miranda, Venezuela. mendez.arnaldo@ula.ve

^d Universidade do Estado do Rio de Janeiro, Resende, Brasil. sergio.munoz@uerj.br

^e Facultad de Ingeniería, Universidad de Los Andes, Mérida, Venezuela. zhichert@gmail.com

Received: November 25th, 2017. Received in revised form: May 24th, 2018. Accepted: June 8th, 2018.

Abstract

In this article, we will propose an approach to generate a planning based on the concepts of k-interaction and consensus, showing for this an algorithm that couples the plant with these concepts. In this sense, the resulting models will be reduced and, consequently the computational complexity.

Keywords: discrete event systems; planning; k-interactions and consensus.

Un algoritmo de planificación en una clase de sistemas de eventos discretos

Resumen

En este artículo, nosotros proponemos un enfoque para generar una planificación fundamentada en los conceptos de k-interacción y consenso, mostrando para ello un algoritmo que acopla la planta con dichos conceptos. En este sentido, se reducirán los modelos resultantes y en consecuencia la complejidad computacional.

Palabras clave: sistemas de eventos discretos; planificación; k-interacciones y consenso.

1. Introducción

Los sistemas distribuidos, de transporte, logísticos, redes de comunicación, entre otros, son Sistemas de Eventos Discretos (SED), y en general las disciplinas teóricas que soportan su estudio incluyen la teoría de sistemas, la investigación de operaciones, la ciencia de la computación y la ingeniería industrial (ver [2,5,11-14,16]).

Usualmente, un SED es referido como un conjunto discreto de eventos, junto con un mecanismo que especifica las ordenes posibles en que estos pueden ocurrir. Así, la no linealidad inherente de los SED clasifica a esta clase de sistemas como complejos (ver [1]). Ellos consisten de un conjunto de procesos independientes asincrónicos (componentes) que operan concurrentemente e interactúan sobre un medio ambiente compartido (ver [9]). La

planificación inapropiada de la ocurrencia de eventos puede conducir a estos sistemas a estancamientos (bloqueo del sistema), a sobreflujos de capacidad, etc, degradando la ejecución deseable para el logro de objetivos. Así, uno siempre quiere que los SED sean funcionalmente correctos en términos de propiedades deseables. En consecuencia, el análisis formal es un aspecto primordial para la verificación de dichas propiedades. Más aun, el análisis automático (enfoque algorítmico) nos permite construir herramientas de software para el estudio adecuado de la dinámica del sistema.

Actualmente, (ver [3,4,6,8,17]) la generación de una planificación en los SED esta direccionada desde:

1. La comparación de metodologías para la toma de decisiones;
2. La aplicación de hojas de cálculo;
3. La modelación de redes de Petri, para señalar las rutas del

How to cite: Mata, G., Bladimir Ruiz, G.M., Camacho, C., Méndez, A., Muñoz, S. and Zambrano, H., Un algoritmo de planificación en una clase de sistemas de eventos discretos. DYNA, 85(206), pp. 283-293, September, 2018.

sistema.

Cualquiera de estos enfoques es potencialmente útil para ordenar el flujo de eventos resultantes. Nosotros presentamos en este artículo un algoritmo para la construcción de una planificación basada en las nociones propias de k-interacción y consenso, entre las componentes asincrónicas de un SED. Este incluye:

- Consenso en cada paso: expresado desde criterios de ordenamientos racionales para la toma de decisiones;
- Solamente las componentes de cambio en el estado actual: determinadas por las k-interacciones físicamente posibles;
- Flexibilidad y opción de reconfiguración del sistema.

Formalmente, dado un SED \mathcal{A} con comportamiento dinámico \mathcal{L} , y una planificación \mathcal{L}_p capturada por un autómata \mathcal{H} , $\mathcal{A} \parallel \mathcal{H}$ es el SED sujeto a la planificación, donde \parallel es la composición paralela de las componentes asincrónicas de \mathcal{A} , condicionadas por \mathcal{H} bajo consenso.

La organización de este artículo comienza con una sección de Preliminares, que incluye las definiciones básicas de la teoría de lenguajes y autómatas, las cuales constituyen fundamentalmente el piso de nuestro modelo; a saber, la composición paralela. Posteriormente, presentamos una metodología de modelación en el contexto de manufactura, que finaliza en una expresión genuina que llamaremos k-interacción. Ella junto con el concepto de consenso, apertura la base teórica que permitirá la generación de un algoritmo para la planificación de un SED. Luego, presentaremos una aplicación en sistemas de producción, para finalmente establecer algunas conclusiones y trabajos futuros.

2. Preliminares

Sea Σ un conjunto cualquiera y consideremos el conjunto Σ^* de todas las combinaciones finitas de elementos de Σ , es decir, el conjunto conformado por n-uplas $s = (\alpha_1, \alpha_2, \dots, \alpha_n)$, con $\alpha_i \in \Sigma$, para todo $1 \leq i \leq n$. El entero n será llamado la longitud de s y es denotado por $|s|$.

Convenimos reescribir a s como $\alpha_1\alpha_2 \dots \alpha_n$, si $n > 0$, al tomar a α en lugar de la 1-upla (α) . Sobre este conjunto definimos una operación binaria como sigue: si $\omega = \sigma_1\sigma_2 \dots \sigma_m$, con $\sigma_j \in \Sigma$, para todo $j > 0$; es otro elemento de Σ^* , entonces el producto $s\omega = \alpha_1\alpha_2 \dots \alpha_n\sigma_1\sigma_2 \dots \sigma_m$ es también un elemento de Σ^* . Esta operación binaria, nombrada concatenación le da a Σ^* estructura de monoide con unidad θ ; la 0-upla. Cualquier subconjunto \mathcal{L} de Σ^* es llamado un lenguaje sobre Σ ; en consecuencia, la unión, intersección, diferencia y complemento de lenguajes son lenguajes sobre Σ . Más aún, si \mathcal{L}_1 y \mathcal{L}_2 son lenguajes sobre un alfabeto Σ entonces

$$\mathcal{L}_1\mathcal{L}_2 = \{s \in \Sigma^* : s = s_1s_2, \quad s_1 \in \mathcal{L}_1, s_2 \in \mathcal{L}_2\} \quad (1)$$

$$\overline{\mathcal{L}_1} = \{s \in \Sigma^* : \exists t \in \Sigma^*, st \in \mathcal{L}_1\} \quad (2)$$

son lenguajes sobre Σ . Sea $s, t \in \Sigma^*$ será llamado un segmento de s si $s = utv$, para algunos $u, v \in \Sigma^*$. Si $u = \theta$, entonces t será llamado un prefijo de s . Por lo tanto, $\overline{\mathcal{L}_1}$ es el conjunto de todos los prefijos de todas las palabras en \mathcal{L}_1 .

Claramente, por definición, $\mathcal{L}_1 \subseteq \overline{\mathcal{L}_1}$. Cuando $\mathcal{L}_1 = \overline{\mathcal{L}_1}$ entonces \mathcal{L}_1 será llamado prefijo-cerrado o simplemente cerrado (ver [7,10,15]).

Otro tipo de operación que se realiza con frecuencia entre lenguajes es la proyección natural, o simplemente la proyección. Formalmente, sean $\Sigma_i, i = 1, 2, \dots, n$; alfabetos y pongamos $\bigcup_{i=1}^n \Sigma_i = \Sigma$. Las proyecciones naturales, $\pi_i: (\bigcup_{i=1}^n \Sigma_i)^* \rightarrow \Sigma_i^*, i = 1, 2, \dots, n$; son dadas por

$$\begin{aligned} \pi_i(\theta) &= \theta_i \\ \pi_i(\alpha) &= \begin{cases} \alpha, & \text{si } \alpha \in \Sigma_i \\ \theta_i, & \text{si } \alpha \notin \Sigma_i \end{cases} \\ \pi_i(s\alpha) &= \pi_i(s)\pi_i(\alpha), \end{aligned} \quad (3)$$

$$\text{para cualquier } s \in \left(\bigcup_{i=1}^n \Sigma_i\right)^* \text{ y } \alpha \in \bigcup_{i=1}^n \Sigma_i$$

Sea Σ un alfabeto. Un Generador Finito Determinístico (GFD) \mathcal{G} sobre Σ (Σ -generador finito), es un séxtuple $(Q, \Sigma, \delta, \mathcal{E}, i, T)$, donde Q es un conjunto finito de estados, i y $T \subseteq Q$ llamados respectivamente estado inicial y conjunto de estados finales, δ es un subconjunto de $Q \times \Sigma \times Q$ de eventos y \mathcal{E} es un subconjunto de $Q \times \Sigma$ de eventos activos (o habilitados) sobre los elementos de Q (ver [7]). Un evento (q, σ, p) será denotado $q \xrightarrow{\sigma} p$. Así, obtenemos un grafo dirigido ponderado: los estados son los vértices y los eventos determinan los arcos, con marca en Σ . El conjunto de todas las trazas $c: i \rightarrow q, q \in Q$ es denotado por $\mathcal{L}(\mathcal{G})$, y será llamado lenguaje generado por \mathcal{G} . Una traza $c: i \rightarrow t$ en \mathcal{G} , con $t \in T$, será llamada una tarea. El conjunto de las etiquetas de tareas en \mathcal{G} forman un lenguaje $\mathcal{L}_m(\mathcal{G})$ de Σ^* llamado el comportamiento de \mathcal{G} ; es decir,

$$\mathcal{L}_m(\mathcal{G}) = \left\{ s \in \Sigma^* : s \text{ es una etiqueta de la traza } \right. \\ \left. c: i \rightarrow t \text{ en } \mathcal{G}, t \in T \right\}. \quad (4)$$

Un lenguaje \mathcal{L} de Σ^* es llamado regular si existe un Σ -generador finito \mathcal{G} tal que $\mathcal{L} = \mathcal{L}_m(\mathcal{G})$. Ahora dado un Σ -generador finito $\mathcal{G} = (Q, \Sigma, \delta, \mathcal{E}, i, T)$ un estado $q \in Q$ será llamado accesible si existe una traza $c: i \rightarrow q$. Si todos los estados de \mathcal{G} son accesibles, entonces diremos que \mathcal{G} es accesible. Se llama parte accesible de \mathcal{G} al Σ -generador finito $\mathcal{G}^a = (Q^a, \Sigma, \delta^a, \mathcal{E}^a, i, T^a)$, donde Q^a es el conjunto de estados accesibles de \mathcal{G} , $T^a = T \cap Q^a$, $\delta^a = \delta / Q^a$, $\mathcal{E}^a \subset \delta^a$. Note que $\mathcal{L}_m(\mathcal{G}) = \mathcal{L}_m(\mathcal{G}^a)$. Por otro lado, δ determina la función parcial $\delta: Q \times \Sigma \rightarrow Q$. Algebraicamente podemos extender δ a una función parcial $\widehat{\delta}: Q \times \Sigma^* \rightarrow Q$ satisfaciendo las siguientes condiciones

$$\begin{aligned} \widehat{\delta}(q, \theta) &= q, \quad \forall q \in Q, \\ \widehat{\delta}(q, st) &= \delta(\widehat{\delta}(q, s), t), \quad \forall q \in Q, s \in \Sigma^*, t \in \Sigma. \end{aligned} \quad (5)$$

Luego, $\hat{\delta}$ así definida, es efectivamente una extensión de δ . Por lo tanto, en lo que sigue, no se hará distinción entre $\hat{\delta}$ y δ . Sean $\mathcal{G}_i = (Q_i, \Sigma_i, \delta_i, \mathcal{E}_i, i_i, T_i)$ $i = 1, 2, \dots, n$; n Σ -generadores finitos determinístico. Se llama Composición Paralela (o síncrona) de los \mathcal{G}_i ; al GFD accesible $| \prod_{i=1}^n \mathcal{G}_i = \mathcal{G}^a$ donde $\mathcal{G} = (\times_{i=1}^n Q_i, \cup_{i=1}^n \Sigma_i, \delta, \mathcal{E}, (i_1, i_2, \dots, i_n), \times_{i=1}^n T_i)$

con $\delta((q_1, q_2, \dots, q_n), \alpha) = (P_1, P_2, \dots, P_n)$, $P_i = \delta_i(q_i, \alpha)$, y $P_j = q_j$, si para $i, j = 1, 2, \dots, n, \alpha \in \cap_i \mathcal{E}_i(q_i) \setminus \cup_{j:j \neq i} \Sigma_j$;

e indefinida en otro caso, y $\mathcal{E}(q_1, q_2, \dots, q_n) = \prod_i \mathcal{E}_i(q_i) \setminus \cup_{j:j \neq i} \Sigma_j$; $i, j = 1, 2, \dots, n$. Se puede chequear sin dificultad que si los \mathcal{L}_i , son lenguajes regulares de los \mathcal{G}_i sobre Σ_i , tales que $\mathcal{L}_m(\mathcal{G}_i) = \mathcal{L}_{mi}$, $i = 1, 2, \dots, n$, entonces se satisface

$$(i) \quad \mathcal{L}(| \prod_{i=1}^n \mathcal{G}_i) = \prod_{i=1}^n \pi_i^{-1}(\mathcal{L}(\mathcal{G}_i)). \quad (6)$$

$$(ii) \quad \mathcal{L}_m(| \prod_{i=1}^n \mathcal{G}_i) = \prod_{i=1}^n \pi_i^{-1}(\mathcal{L}_m(\mathcal{G}_i)). \quad (7)$$

3. Metodología y construcción del modelo en manufactura

Un sistema de manufactura es un par de conjuntos: conjunto de actividades y conjunto de recursos; que interactúan para obtener un producto. Este incluye un plan de proceso de producción (programa) que especifica las actividades y los tipos de recursos, así como también las relaciones de precedencia entre actividades. Sea \mathcal{G} un sistema de manufactura. La metodología incluida en este artículo para llevar a cabo la modelación será establecida mediante la composición paralela, organizada desde las interpretaciones de condiciones locales de estados (o estatus) y eventos tal como sigue:

1. Identificar las actividades y recursos necesarios para la producción de un artículo (o producto).
2. Ordenar las actividades por las relaciones de precedencia tal como lo establece el programa.
3. Para cada actividad, crear un generador componente que represente el estatus oci:=<<ocioso>> o eje:=<<ejecución>>.
4. etiquetar un evento α :=<<comienzo de ejecución de la actividad>> creando un arco desde oci hasta eje, y un evento β :=<<completación de actividad>> para incluir un arco desde eje hasta oci. En general, el evento de completación para una actividad será el evento de comienzo para la nueva actividad (ver Fig. 1).

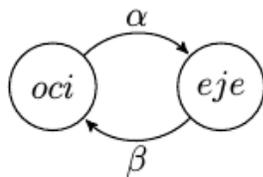


Figura 1. Un generador componente para una actividad. Fuente: Los autores.

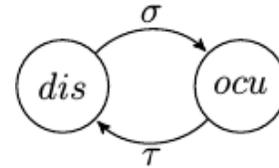


Figura 2. Un generador componente para un recurso. Fuente: Los autores.

5. Para cada actividad, crear un generador componente para cada recurso necesario para el comienzo de la actividad dis:=<<disponible>> y ocu:=<<ocupado>> etiquetar un evento σ :=<<comienzo de utilización del recurso>> el cual será igual al comienzo de la actividad; creando un arco desde dis hasta ocu. Finalmente, un evento τ :=<<finalización de utilización del recurso>> el cual será igual al de finalización de dicha actividad; para incluir un arco desde ocu hasta dis. En general, el evento de completación para una actividad con recurso de entrada corresponde a la disponibilidad de dicho recurso comienzo para la nueva actividad (ver Fig. 2)
6. Establecer el estado inicial; para cada actividad y cada recurso los estatus ocioso y disponible determinan respectivamente el estado inicial.

El modelo para el sistema de manufactura \mathcal{G} será establecido por $| \prod_{i=1}^n \mathcal{G}_i$, con $\mathcal{L} = \prod_{i=1}^n \pi_i^{-1}(\mathcal{L}_i)$, donde los \mathcal{G}_i determinan las dinámicas de las n componentes constituyentes \mathcal{G}_i , $i = 1, 2, \dots, n$; del sistema. La construcción del modelo para el sistema de manufactura $\mathcal{G} = (Q, \Sigma, \delta, \mathcal{E}, q_0, Q_m)$ será establecido desde la interpretación, organización y conjugación de los generadores componentes. Para ello,

Definición 1. Sea \mathcal{G} el generador que representa un sistema global. Un evento $\alpha \in \cup_{i=1}^n \Sigma_i$ será llamado una k -interacción en \mathcal{G} si cada vez que ocurre α éste produce cambios en k -componentes, $k \leq n$, siempre que $\alpha \in \prod_i \mathcal{E}_i(q_i) \setminus \cup_{j:j \neq i} \Sigma_j$, $i, j = 1, 2, \dots, k$.

Convendremos en lo que sigue que una k -interacción en \mathcal{G} es simplemente un evento de orden k en \mathcal{G} y al conjunto constituido por las k -interacciones lo denotamos Σ_k , $0 < k \leq n$. Luego, $\Sigma = \cup_{k=1}^n \Sigma_k$ es una partición de Σ ($\Sigma_i \cap \Sigma_k = \emptyset$, $i \neq j$). Una k -interacción no es más que una expresión simbólica de la ocurrencia simultanea de eventos de ciertos generadores componentes. Estos eventos representan acciones distintas (con etiquetas distintas) pero podríamos darles una interpretación que nos permita conjugarlas mediante un mismo evento. Otro hecho aún más significativo para la construcción del modelo, ocurre cuando una actividad en el sistema requiere de uno o más recursos; donde el comienzo y la completación de dicha actividad tiene una relación directa con el comienzo y la finalización del uso de esos recursos. Igualmente, estos eventos tienen representaciones distintas en el sistema, pero la interpretación de este hecho por el mismo evento nos permitirá simplificar la construcción del modelo. Por lo tanto, diremos que las componentes \mathcal{G}_i , $i = 1, 2, \dots, n$; del sistema de manufactura \mathcal{G} estarán determinadas bajo simetría de los eventos que representan los comienzos y finalizaciones de las actividades, así como de la adquisición y liberación de los recursos. Finalmente, en la

composición paralela de los \mathcal{G}_i , un evento común solo ocurre si ocurre simultáneamente en los \mathcal{G}_i relacionados a través de dicho evento. Por lo tanto, los \mathcal{G}_i están sincronizados sobre los eventos comunes. Los eventos no comunes no están sujetos a esta restricción y pueden ocurrir siempre que sea posible. Por lo tanto, $\mathcal{G} = \bigcap_{i=1}^n \mathcal{G}_i$.

4. Condiciones y planificación por consenso

Sea $\mathcal{G} = (Q, \Sigma, \delta, \mathcal{E}, q_0, Q_m)$ (sistema de manufactura) la composición síncrona de n componentes (subsistemas) $\mathcal{G}_i = (Q_i, \Sigma_i, \delta_i, \mathcal{E}_i, q_{0_i}, Q_{m_i}), i = 1, \dots, n$. Sabemos que, para cualesquiera \mathcal{L}_i y $\mathcal{L}_{m_i}, i = 1, \dots, n$; existe un generador finito determinístico \mathcal{G}_i tal que $\mathcal{L}_i = \mathcal{L}(\mathcal{G}_i)$ y $\mathcal{L}_{m_i} = \mathcal{L}_m(\mathcal{G}_i), i = 1, 2, \dots, n$. Sea \mathcal{C} una condición (expresada como un lenguaje) para \mathcal{G} y para cada i , con $i = 1, 2, \dots, n$; sea $\mathcal{C}_i = \pi_i(\mathcal{C})$ la proyección i -ésima de \mathcal{C} . La afectación que puede producir \mathcal{C} sobre \mathcal{G} puede ser proyectada a las dinámicas de las componentes. Más aún, existen condiciones no únicas que producen el efecto global impuesto por las condiciones locales. Por lo tanto, conocer $\mathcal{L} \cap \mathcal{G}$ (lenguaje global ante una condición) es conocer los $\mathcal{L}_i \cap \mathcal{C}_i, i = 1, \dots, n$; es decir, conocer la afectación \mathcal{C} sobre el sistema \mathcal{G} es conocer las afectaciones \mathcal{C}_i sobre las componentes \mathcal{G}_i . Note que la condición \mathcal{C} no tiene sentido lógico, en relación al acoplamiento con \mathcal{G} , si ésta no está reflejada en la dinámica del sistema. En este sentido, para la construcción de la metodología por consenso podemos suponer que la planificación \mathcal{L}_p es un sublenguaje de $\mathcal{L}(\mathcal{G})$.

Definición 2. Sean \mathcal{G} un sistema global y \mathcal{C} una condición sobre \mathcal{G} . Para cualesquiera $q = \delta(q_0, s)$ y $\alpha \in \mathcal{E}(q)$ una k -interacción, sean j_1, j_2, \dots, j_k ; los índices de las componentes de cambio por la ocurrencia de α en q . Diremos que α es obtenida por consenso si para todo $j_l, l = 1, \dots, k$ se tiene que $\pi_{j_l}(s)\alpha \in \pi_{j_l}(\mathcal{C} \cap \mathcal{L})$.

De acuerdo a la definición 2 establecemos el conjunto de todos eventos permitidos en q obtenidos por consenso, el cual denotaremos por $\mathcal{J}(q)$. La construcción para la planificación en los sistemas de manufactura; $\mathcal{G} = (Q, \Sigma, \delta, \mathcal{E}, i, Q_m)$ establecida por el producto síncrono, está fundamentada en el consenso como sigue: Para cada estado del sistema de manufactura; $q = \delta(i, s)$, donde $s \in (\bigcup_{i=1}^n \Sigma_i)^*$ e $i = (i_1, i_2, \dots, i_n)$ es el estado inicial, consideramos el conjunto de todos los eventos habilitados en q ; $\mathcal{E}(q)$: para cada k -interacción $\alpha \in \mathcal{E}(q)$, donde j_1, j_2, \dots, j_k son los índices de las componentes de cambio por la ocurrencia en q , α deberá satisfacer: $\pi_{j_l}(s)\alpha \in \pi_{j_l}(\mathcal{C} \cap \mathcal{L}), \forall j_l, l = 1, \dots, k$, para ser obtenida por consenso. Comenzamos el procedimiento desde el estado inicial i , y consideremos el subconjunto de eventos obtenidos por consenso en i ; $\mathcal{J}(i)$. La ocurrencia de un evento α en $\mathcal{J}(i)$ determinará un nuevo estado q , y por ende se genera un arco dirigido desde i al nuevo estado q etiquetado por el evento α ; $i \xrightarrow{\alpha} q$. Ahora, esto ocurrirá eventualmente para todo evento α en $\mathcal{J}(i)$. Y en el estado actual q se repetirá el proceso antes mencionado con respecto a $\mathcal{J}(q)$, generando

de esta manera bajo consenso el árbol que determina el generador que reconoce la planificación desde un punto de vista constructivo. El procedimiento repetido sucesivamente generará un conjunto de estados bajo consenso y puede ser representado gráficamente. Finalmente, el proceso resultante será un reconocedor de la planificación y será denotado por $\mathcal{P}(\mathcal{G})$. Formalmente,

Definición 3. Sea $\mathcal{P}(\mathcal{G})$ como antes. El lenguaje generado por $\mathcal{P}(\mathcal{G})$ será llamado lenguaje de planificación y es dada por

- $\theta \in \mathcal{L}(\mathcal{P}(\mathcal{G}))$;
- para cada $s \in (\bigcup_{i=1}^n \Sigma_i)^*$, y para cada $\alpha \in \bigcup_{i=1}^n \Sigma_i$ se tiene que: $s \in \mathcal{L}(\mathcal{P}(\mathcal{G}))$, $s\alpha \in \mathcal{L}(\mathcal{G}) = \mathcal{L}$, α es obtenido por consenso, si y solo si, $s\alpha \in \mathcal{L}(\mathcal{P}(\mathcal{G}))$.

El lenguaje de planificación marcado por $\mathcal{P}(\mathcal{G})$ es dado por $\mathcal{L}_m(\mathcal{P}(\mathcal{G})) = \mathcal{L}(\mathcal{P}(\mathcal{G})) \cap \mathcal{L}_m(\mathcal{G})$.

Para facilitar la escritura, denotaremos a los lenguajes de planificación y planificación marcada por \mathcal{L}_p y \mathcal{L}_{p_m} respectivamente. El conjunto de todos los estados generados por consenso desde el estado inicial i está dado por $Q_p = \{q \in Q : \exists s \in \mathcal{L}_p, \delta(i, s) = q\}$, y el generador determinístico accesible $\mathcal{P}(\mathcal{G}) = (Q_p, \Sigma, \delta, \mathcal{J}, i, Q_{p_m})$ es el reconocedor de \mathcal{L}_p . La selección de eventos comunes en la composición paralela, lo cual determina el acoplamiento entre las componentes, junto con la reducción en cuanto a la navegación de estados sometidas a las condiciones impuestas localmente permite lidiar eficientemente con la complejidad computacional resultante del modelo completo.

5. Algoritmo para la planificación

Para la construcción de $\mathcal{P}(\mathcal{G})$ mediante el algoritmo son necesarias las siguientes especificaciones (ver Fig. 3):

- Los elementos del generador composición paralela \mathcal{G} : el estado inicial $(q_{0_1}, \dots, q_{0_n})$, el conjunto de estados finales $\times_{i=1}^n Q_{i_m}$, el conjunto $\times_{i=1}^n Q_i$ de estados, el conjunto de eventos activos \mathcal{E} y la función de transición δ ;
- el conjunto de las k -interacciones; $\Sigma^k, k \in \{1, 2, \dots, n\}$, donde n representa el número de componentes constituyendo a \mathcal{G} ;
- \mathcal{L} el lenguaje generado por el generador composición paralela; y
- \mathcal{C} expresando el condicionamiento global sobre la dinámica del sistema.

Proposición 1. Sea n el cardinal del conjunto Q_p . La planificación $\mathcal{P}(\mathcal{G})$ se puede llevar a cabo a la sumo en $n - 1$ pasos.

Demostración. Supongamos que para $q_i \in Q_p; i = 0, 1, \dots, n - 1$ ($q_j \neq q_i; i \neq j$); se tiene que $\mathcal{J}(q_i) = \alpha_{i+1}$, tal que $\delta(q_i, \alpha_{i+1}) = q_{i+1}; i = 0, 1, \dots, n - 2$. Ahora, si $\mathcal{J}(q_{n-1}) = \emptyset$, entonces el nivel del nodo que representa a q_{n-1} está dado por

$$l(q_{n-1}) = \sum_{i=0}^{n-1} |\mathcal{J}(q_i)| \leq \sum_{i=0}^{n-1} 1 = n. \quad (8)$$

Algoritmo 1 Para la construcción de $\mathcal{P}(\mathcal{G})$

Entrada: $\mathcal{G} = \left(\bigtimes_{i=1}^n Q_i, \bigcup_{i=1}^n \Sigma_i, \delta, \mathcal{E}_i, (q_{01}, \dots, q_{0n}), \bigtimes_{i=1}^n Q_{im} \right), \Sigma^k, \mathcal{L}, \mathcal{C}$

Salida: $\mathcal{P}(\mathcal{G}), \mathcal{L}_p$

- 1: **inicio** : Alcanzados $\leftarrow \{(q_{01}, \dots, q_{0n})\}$, $\mathcal{J}((q_1, \dots, q_n)) \leftarrow \emptyset$, $\mathcal{E} \leftarrow \emptyset$, (q_{01}, \dots, q_{0n}) no etiquetado, $s \leftarrow \theta$, $\mathcal{L}_p \leftarrow \emptyset$
- 2: **mientras** $(q_1, \dots, q_n) \in$ Alcanzados \wedge (q_1, \dots, q_n) no etiquetado **hacer**
- 3: **para** $\alpha \in \mathcal{E}((q_1, \dots, q_n))$ **hacer**
- 4: **para** $k \in \{1, 2, \dots, n\}$ **hacer**
- 5: **si** $\alpha \in \Sigma^k$ **entonces**
- 6: $l = k$, interrumpir
- 7: **fin si**
- 8: **para** $i \in \{j_1, \dots, j_j\}$ **hacer**
- 9: **si** $\pi_i(s)\alpha \in \mathcal{C}_i \cap \mathcal{L}_i$ **entonces**
- 10: Imprimir: La componente i -ésima no viola la condición \mathcal{C}
- 11: **en otro caso**
- 12: Imprimir: La componente i -ésima viola la condición \mathcal{C}
- 13: **fin si**
- 14: **fin para**
- 15: $\mathcal{J}((q_1, \dots, q_n)) \leftarrow \alpha$, $s \leftarrow s \cdot \alpha$, $\mathcal{L}_p \leftarrow \mathcal{L}_p \cup \{s\}$
- 16: Imprimir: α es obtenida por consenso en (q_1, \dots, q_n)
- 17: **si** $\delta((q_1, \dots, q_n), \alpha) = (q'_1, \dots, q'_n) \notin$ Alcanzados **entonces**
- 18: Alcanzados $\cup \{(q_1, \dots, q_n) \wedge (q'_1, \dots, q'_n)\}$ no etiquetado
- 19: **fin si**
- 20: **fin para**
- 21: (q_1, \dots, q_n) etiquetado
- 22: **fin para**
- 23: **fin mientras**
- 24: **devolver** $\mathcal{P}(\mathcal{G}), \mathcal{L}_p$

Figura 3. Algoritmo para la construcción de $\mathcal{P}(\mathcal{G})$.

Fuente: Los autores.

Si $\mathcal{J}(q_{n-1}) \neq \emptyset$, entonces $\delta(q_{n-1}, \alpha_n) = q_n$. Esto implica que necesariamente $q_n = q_i$, para algún i ; $i = 0, 1, \dots, n-1$. Sea $l: Q_p \rightarrow \mathbb{N}$, $\mathbb{N} = \{0, 1, 2, \dots\}$, una función de nivel: para cada $q_j \in Q_p$, $l(q_j) \in \mathbb{N}$ es el nivel del estado q_j definida como $l(q_j) = \sum_{i=0}^{n-1} |\mathcal{J}(q_i)|$; entonces, $l(q_n) \leq l(q_{n-1}) = n-1$. Finalmente, $l(q_{n-1}) \leq n-1$.

Nosotros construimos al conjunto $\mathcal{J}(q)$ cuyos elementos son las k -interacciones obtenidas por consenso, $\mathcal{J}(q) \subseteq \mathcal{E}(q)$, para obtener posteriormente, para cada $\alpha \in \mathcal{J}(q)$ una nueva bolsa $\mathcal{J}(\delta(q, \alpha))$. Este proceso continua considerando la nueva bolsa $\mathcal{J}(\delta(q, \alpha)) \subseteq \mathcal{E}(\delta(q, \alpha))$, obteniendo de esta manera el lenguaje de planificación \mathcal{L}_p , en algún sentido, por sucesiones de bolsas de la forma $\mathcal{J}_0(q_0), \mathcal{J}_1(\delta(q_0, \alpha_1)), \dots, \mathcal{J}_n(\delta(\dots \delta(\delta(q_0, \alpha_1), \alpha_2), \dots), \alpha_n)) \dots$. Finalmente, $\mathcal{P}(\mathcal{G})$ es una realización de \mathcal{L}_p construida desde las afectaciones locales \mathcal{C}_i , $i = 1, \dots, n$; sobre las componentes \mathcal{G}_i , $i = 1, \dots, n$. La justificación por la cual una condición \mathcal{C} sobre \mathcal{G} es dada, es por la necesidad de establecer una planificación sobre el sistema. Por ello estableceremos el acoplamiento con \mathcal{G} .

Teorema 1. Dado una condición \mathcal{C} sobre \mathcal{G} . Sea para cada j , $j = 1, \dots, n$, $\mathcal{C}_j = \pi_j(\mathcal{C})$ la condición local para \mathcal{G}_j . Entonces, $\mathcal{C} \cap \mathcal{L} = \mathcal{L}_p$ (más restrictivamente, $\mathcal{C} \cap \mathcal{L}_p = \mathcal{L}_{p_m}$).

Demostración.

$s \in \mathcal{L}_p \Leftrightarrow$ admite una descomposición de la forma

$$\begin{aligned} & \alpha_1 \alpha_2 \dots \alpha_n \text{ con } \alpha_i \in \mathcal{J}_{i-1}, i = 1, \dots, n \\ & \Leftrightarrow \omega \alpha_n, \text{ con } \omega = \alpha_1 \alpha_2 \dots \alpha_{n-1} \wedge \alpha_n \in \mathcal{J}_{n-1} \\ & \Leftrightarrow \omega \alpha_n, \text{ con } \omega \in \mathcal{J}_0 \mathcal{J}_1 \dots \mathcal{J}_{n-2} \wedge \alpha_n \in \mathcal{J}_{n-1} \\ & \Leftrightarrow \pi_j(\omega) \alpha_n \in \mathcal{C}_j \cap \mathcal{L}_j, \text{ para todo } j \\ & \Leftrightarrow \pi_j(\omega) \pi_j(\alpha_n) \in \pi_j(\mathcal{C} \cap \mathcal{L}), \text{ para todo } j \\ & \Leftrightarrow \pi_j(\omega \alpha_n) \in \pi_j(\mathcal{C} \cap \mathcal{L}), \text{ para todo } j \\ & \Leftrightarrow \prod_{j=1}^n \pi_j^{-1}(\pi_j(s)) \in \prod_{j=1}^n \pi_j^{-1}(\pi_j(\mathcal{C} \cap \mathcal{L})) \\ & \Leftrightarrow s \in \mathcal{C} \cap \mathcal{L} \end{aligned} \quad (9)$$

Análogamente, se puede probar que $\mathcal{C} \cap \mathcal{L}_p = \mathcal{L}_{p_m}$. ■

El resultado anterior; teorema 1 expresa que la planificación global sobre el sistema \mathcal{G} puede ser obtenida desde las planificaciones locales de las componentes \mathcal{G}_i , mostrando de esta manera el análisis global desde un punto de vista local.

6. Aplicación

6.1. Un sistema de producción por lote

Consideremos un sistema de producción de pinturas constituido solo por tres líneas de producción, ya que la gama de productos elaborados por la empresa podría ser muy amplia. Para cada línea de producción las unidades que la conforman establecen los subsistemas. Un sistema de producción de pinturas es un sistema de producción por lote, porque la finalización de producción de una pintura específica dará comienzo a la fabricación de otra pintura quizás con características distintas. La materia prima (recursos) utilizada para la elaboración de las pinturas son: los pigmentos, los vehículos y los solventes, a los que se suman los aditivos. Para la construcción del modelo asumiremos la existencia de estos recursos. Para cada subproceso (o subsistema) en el proceso de fabricación de pinturas cada actividad es llevada a cabo por una unidad de producción. Las actividades son descritas a continuación:

1. La dispersión es la fase inicial y de mayor duración en el proceso, en la cual se agita la carga del vehículo, el agregado de solventes y aditivos. Luego, se agregan sucesivamente los diferentes tipos de pigmento, empezando con los de más difícil mojado. En esta fase se logra obtener una pasta llamada pasta de molienda.
2. En una segunda fase; molienda, se consigue obtener una mayor finura de la pasta de molienda por medio de los llamados molinos.
3. En una última etapa dilución o mezcla se ajusta la consistencia de la pintura con el agregado de solventes y/o resinas compatibles con la base de molienda y generalmente se incluyen aditivos que ayudan en la estabilización.

A continuación, describiremos las líneas de producción del sistema de producción de pinturas; pintura blanca, pintura de color y pintura texturizada (o masilla), en la que se pueden distinguir algunas de las etapas discutidas anteriormente (ver Fig. 4):

- a) PB := subsistema de producción de base incolora. Éste

está constituido por dos unidades de producción distinguidas por sus operaciones para la elaboración de este producto:

- i) PB_1 := unidad de dispersión de la base concentrada incolora;
- ii) PB_2 := unidad de mezclado de terminación de base incolora.

Luego de estas etapas, se obtiene la base incolora, la cual puede continuar a envasado, a completar el proceso de fabricación de pintura texturizada o a completar el proceso de fabricación de pintura de color.

- b) MA := subsistema de producción de pintura texturizada. Para este producto se distinguen las siguientes operaciones llevadas a cabo por las correspondientes unidades de producción:

- i) MA_1 := unidad de dispersión de base de alta viscosidad;
- ii) MA_2 := unidad de mezclado del concentrado con la base blanca.

Una vez obtenido el concentrado producto de la mezcla con la base blanca, la culminación se logra con el envasado del producto.

- c) PC := subsistema de producción de pintura de color. Éste se caracteriza por las siguientes actividades:

- i) PC_1 := unidad de dispersión del pigmento para formar una pasta de color;
- ii) PC_2 := unidad de molienda de la pasta de color para formar empaste;
- iii) PC_3 := unidad de mezclado del empaste con resinas y solventes formando un concentrado de color;
- iv) PC_4 := unidad de mezclado del concentrado de color con la base incolora.

La pintura de color es el resultado de la mezcla del concentrado de color con la base incolora. La culminación de este proceso se logra con el envasado de la pintura, luego de haber pasado previamente por control de calidad.

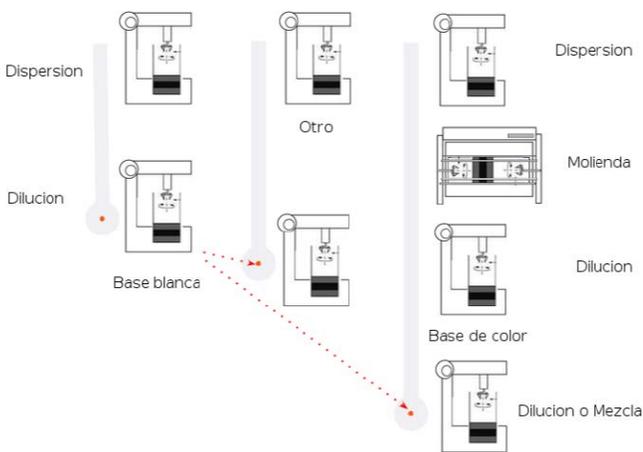


Figura 4. Líneas de producción y etapas en la elaboración de las pinturas. Fuente: Los autores

En la producción de la pintura de color se utiliza la técnica tirada de color, que consiste de un proceso (reinicializable) que se lleva a cabo para disminuir gradualmente el valor del color (o diferencia luminosa) hasta alcanzar el color más oscuro. Esto reduce al mínimo la limpieza y la nueva configuración de la maquinaria entre cada lote de pintura de color. El blanco (pintura opaca, tinta no transparente) es el único color que no es utilizado en esta técnica, ya que, una pequeña cantidad de pigmento blanco puede afectar negativamente a los colores medios. Con la idea de facilitar los cálculos en la construcción de la metodología de modelación para esta clase de sistemas de manufactura solo consideraremos los casos expuestos anteriormente. Excluiremos, por ejemplo, los subprocesos de filtrado y envasado en los procesos de fabricación de pintura texturizada (o masilla) y pintura de color. Además, consideraremos que un producto obtenido en cada unidad de producción es un recurso para la siguiente unidad en cada subproceso. Más aún, la base incolora en PB es un recurso necesario para continuar el envasado de la misma o completar los procesos de fabricación en MA y PC .

6.2. Interpretaciones en el sistema de producción

Para el subsistema de producción de pintura blanca PB , para la actividad dispersión (llevada a cabo por la unidad PB_1) representando los estatus $q_{0_1} \ll \text{ocioso} \gg$ y $q_{1_1} \ll \text{ejecución} \gg$ los eventos σ_1 y σ_3 etiquetan el comienzo y la completación respectivamente (ver Fig. 9), y para la actividad de mezclado o dilución (llevada a cabo por la unidad PB_2) representando los estatus $q_{0_2} \ll \text{ocioso} \gg$ y $q_{1_2} \ll \text{ejecución} \gg$ los eventos σ_3 y σ_5 etiquetan el comienzo y su completación respectivamente de dicha actividad (ver Fig. 5).

La composición síncrona de los $\mathcal{G}_i = (Q_i, \Sigma_i, \delta_i, \varepsilon_i, q_{0_i}, Q_{m_i})$, $i = 1, 2$; representa al subsistema PB , donde cada uno de sus elementos pueden distinguirse en la Fig. 6.

Su comportamiento es dado por

$$\mathcal{L}_{m_1} = (\sigma_1 \sigma_3 \sigma_5 + \sigma_1 \sigma_3 \sigma_1 (\sigma_5 \sigma_3 \sigma_1)^* \sigma_5 \sigma_3 \sigma_5)^*$$

Para el subsistema de producción de masilla MA , para la actividad dispersión (llevada a cabo por la unidad MA_1)

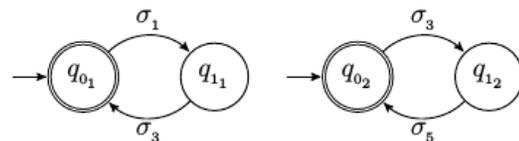


Figura 5. Unidades de producción PB_1 y PB_2 . Fuente: Los autores.

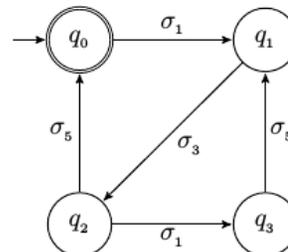


Figura 6. Generador composición representando a PB . Fuente: Los autores.

representando los estatus $q_{0_3} = \ll \text{ocioso} \gg$ y $q_{1_3} = \ll \text{ejecución} \gg$ los eventos σ_{17} y σ_{19} etiquetan el comienzo y la completación de dicha actividad (ver Fig. 7). Así mismo, para la actividad mezclado (llevada a cabo por la unidad MA_2) representando los estatus $q_{0_4} = \ll \text{ocioso} \gg$ y $q_{1_4} = \ll \text{ejecución} \gg$ los eventos σ_{19} y σ_{21} etiquetan el comienzo y la completación respectivamente (ver Fig. 7).

Las $G_i = (Q_i, \Sigma_i, \delta_i, \mathcal{E}_i, q_{0_i}, Q_{m_i})$ $i = 3, 4$; representan las dos unidades de producción del subsistema MA y la composición síncrona de estas dos unidades se muestra en la Fig. 8.

Su comportamiento es dado por

$$\mathcal{L}_{m_2} = (\sigma_{17}\sigma_{19}\sigma_{21} + \sigma_{17}\sigma_{19}\sigma_{17}(\sigma_{21}\sigma_{19}\sigma_{17})^*\sigma_{21}\sigma_{19}\sigma_{21})^*$$

Para el subsistema de producción de pintura de color PC , para la actividad dispersión (llevada a cabo por la unidad PC_1) representando los estatus $q_{0_5} = \ll \text{ocioso} \gg$ y $q_{1_5} = \ll \text{ejecución} \gg$ los eventos σ_7 y σ_9 etiquetan el comienzo y la completación (ver Fig. 9) y para la actividad molienda (llevada a cabo por la unidad PC_2) representando los estatus $q_{0_6} = \ll \text{ocioso} \gg$ y $q_{1_6} = \ll \text{ejecución} \gg$ los eventos σ_9 y σ_{11} etiquetan el comienzo y su completación respectivamente (ver Fig. 9).

Para el mezclado del empaste con resinas y solventes (concentrado de color llevado a cabo por la unidad PC_3) representando los estatus $q_{0_7} = \ll \text{ocioso} \gg$ y $q_{1_7} = \ll \text{ejecución} \gg$ los eventos σ_{11} y σ_{13} etiquetan el comienzo y la completación (ver Fig. 9) y el mezclado del concentrado de color con la base incolora (llevada a cabo por la unidad PC_4) representando los estatus $q_{0_8} = \ll \text{ocioso} \gg$ y $q_{1_8} = \ll \text{ejecución} \gg$ los eventos σ_{13} y σ_{15} etiquetan el comienzo y la completación respectivamente (ver Fig. 9).

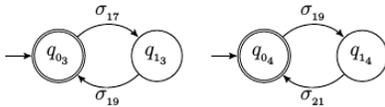


Figura 7. Unidades de producción MA_1 y MA_2 . Fuente: Los autores.

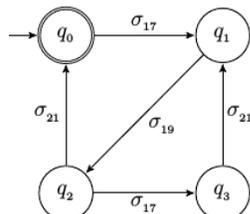


Figura 8. Generador composición representando a MA . Fuente: Los autores.

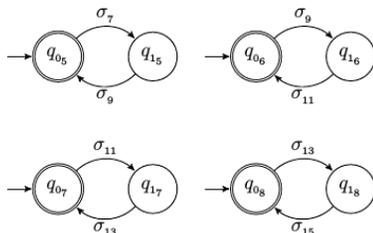


Figura 9. Unidades de producción PC_i , $i = 1, \dots, 4$. Fuente: Los autores.

Igualmente, la composición síncrona de los $G_i = (Q_i, \Sigma_i, \delta_i, \mathcal{E}_i, q_{0_i}, Q_{m_i})$, $i = 5, \dots, 8$; representando las unidades de producción del subsistema PC , se ilustra en la Fig. 10.

Su comportamiento es dado por

$$\mathcal{L}_{m_3} = (\sigma_7\sigma_9\sigma_{11}\sigma_{13}\sigma_{15} + \sigma_7\sigma_9(\sigma_7\sigma_{11} + \sigma_{11}\sigma_7)\sigma_{13} + \sigma_{11}\sigma_{13}\sigma_7) \dots + \dots$$

Para el envasado de la pintura blanca (denotamos esta actividad por EN representado los estatus $q_{0_9} = \ll \text{ocioso} \gg$ y $q_{1_9} = \ll \text{ejecución} \gg$ los eventos σ_{23} y σ_{25} etiquetan el comienzo y la completación de dicha actividad respectivamente (ver Fig. 11).

Su comportamiento es $\mathcal{L}_{m_4} = (\sigma_{23}\sigma_{25})^*$. Para los subsistemas de producción EN, MA y PC requerimos la disponibilidad de la base blanca, para este recurso representado $q_{0_{10i}} \ll \text{ocioso} \gg$ y $q_{1_{10i}} \ll \text{ejecución} \gg$ $i = \{1, 2, 3\}$: el evento σ_5 etiqueta el comienzo de disponibilidad del recurso (agregación de la base blanca por PB), mientras que, los eventos σ_{13} , σ_{19} y σ_{23} etiquetarán la finalización de disponibilidad del mismo (uso de la base blanca por los subsistemas de producción EN, MA y PC respectivamente). El generador composición paralela representando el uso de la base blanca por parte de EN, MA y PC , es denotado por Re y está dado a continuación en la Fig. 12.

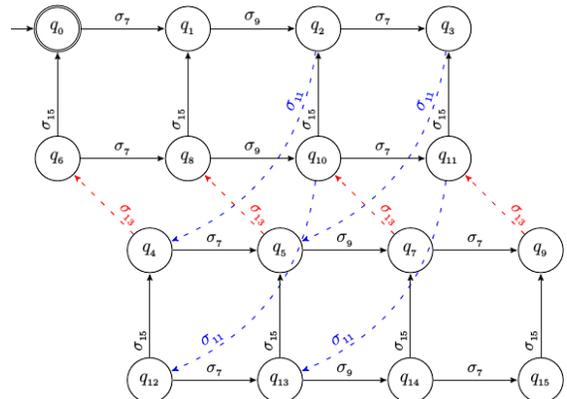


Figura 10. Generador composición representando a MA . Fuente: Los autores.

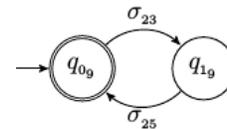


Figura 11. Generadores representando a EN . Fuente: Los autores.

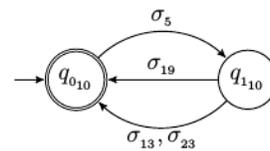


Figura 12. Generador composición representando Re . Fuente: Los autores.

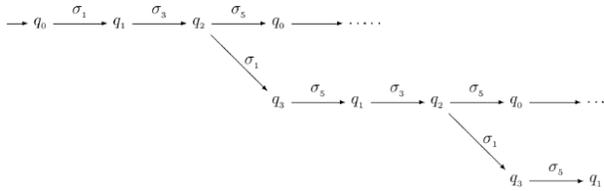


Figura 13. Muestra parcial de la construcción de $\mathcal{P}(\mathcal{G}_1)$.
Fuente: Los autores.

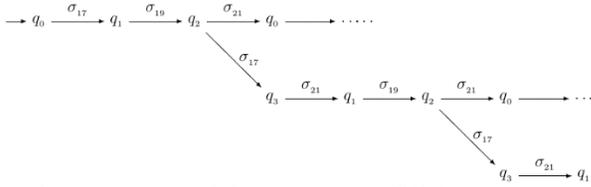


Figura 14. Muestra parcial de la construcción de $\mathcal{P}(\mathcal{G}_2)$.
Fuente: Los autores.

6.4. Planificaciones locales

Para cada estado del sistema de producción de pintura; $q = \delta(q_0, s)$, donde $s \in \Sigma^*$ y $q_0 = (q_{0_1}, q_{0_2}, \dots, q_{0_5})$ es el estado inicial, consideramos el conjunto de todos los eventos habilitados en q ; $\mathcal{E}(q)$: para cada k -interacción $\sigma \in \mathcal{E}(q)$, donde j_1, j_2, \dots, j_5 son los índices de las componentes de cambio por la ocurrencia en q , σ es consensuada sí $\pi_{j_i}(s) \in \mathcal{K}_i, \forall j_i, i = 1, \dots, 5$. Para el subsistema de producción PB , las actividades de dispersión y dilución son llevadas a cabo siempre que $\pi_1(s)\sigma \in \mathcal{K}_1, \sigma \in \Sigma_1$. El procedimiento repetido sucesivamente generará un conjunto de estatus Q_{p_1} obtenidos por consenso y su dinámica puede ser representada gráficamente (ver Fig. 13).

Donde $\mathcal{L}_{p_1} = (\sigma_1\sigma_3\sigma_5)^5 + \sigma_1\sigma_3\sigma_1(\sigma_5\sigma_3\sigma_1)^3\sigma_5\sigma_3\sigma_5$; es lenguaje de planificación local. Para el subsistema de producción MA , las actividades de dispersión y mezclado con la base blanca se ejecutarán siempre que $\pi_2(s)\sigma \in \mathcal{K}_2, \sigma \in \Sigma_2$. El procedimiento repetido sucesivamente generará un conjunto de estatus Q_{p_2} obtenidos por consenso y su dinámica puede ser representada gráficamente (ver Fig. 14).

Donde $\mathcal{L}_{p_2} = (\sigma_{17}\sigma_{19}\sigma_{21})^2 + \sigma_{17}\sigma_{19}\sigma_{17}(\sigma_{21}\sigma_{19}\sigma_{17})^0\sigma_{21}\sigma_{19}\sigma_{21}$; es el lenguaje de planificación local. Para el subsistema de producción PC , las actividades de dispersión, molienda, dilución y mezclado con la base blanca se efectuaran siempre que $\pi_3(s)\sigma \in \mathcal{K}_3, \sigma \in \Sigma_3$. Repetido sucesivamente el procedimiento se generará un conjunto de estatus Q_{p_3} obtenidos por consenso y su dinámica puede ser representada gráficamente (ver Fig. 15).

Donde \mathcal{L}_{p_3} lenguaje de planificación local es $\mathcal{L}_{p_3} = (\sigma_7\sigma_9\sigma_{11}\sigma_{13}\sigma_{15})^2 + \sigma_7\sigma_9\sigma_7\sigma_{11}\sigma_{13}\sigma_9(\sigma_{11}\sigma_{15}\sigma_{13}\sigma_7\sigma_9)^0\sigma_{11}\sigma_{15}\sigma_{13}\sigma_{15} + \dots$
Para el subsistema de producción EN , la actividad de envasado se cumple si $\pi_4(s)\sigma \in \mathcal{K}_4, \sigma \in \Sigma_4$.

El procedimiento repetido sucesivamente generará un conjunto de estatus Q_{p_4} obtenidos por consenso y su dinámica puede ser representada gráficamente (ver Fig. 16).

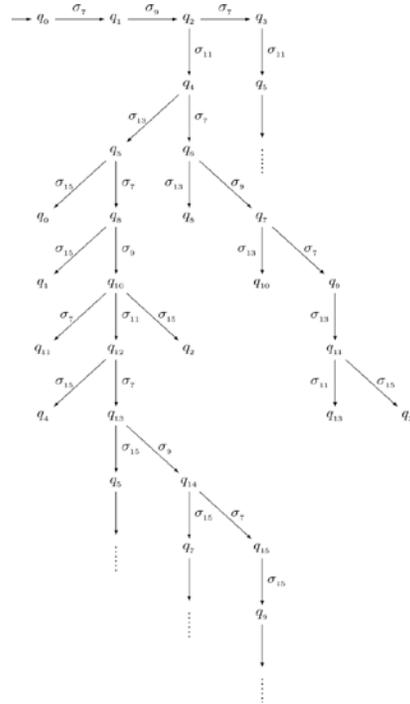


Figura 15. Muestra parcial de la construcción de $\mathcal{P}(\mathcal{G}_3)$.
Fuente: Los autores.

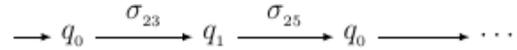


Figura 16. Muestra parcial de la construcción de $\mathcal{P}(\mathcal{G}_4)$.
Fuente: Los autores.

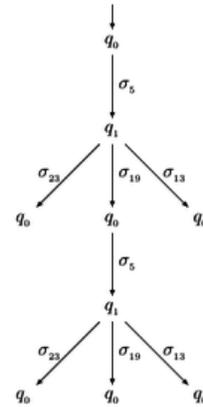


Figura 17. Muestra parcial de la construcción de $\mathcal{P}(\mathcal{G}_5)$.
Fuente: Los autores.

Donde $\mathcal{L}_{p_4} = (\sigma_{23}\sigma_{25})^1$ es lenguaje de planificación local. Para el sistema de producción, la distribución de la base blanca proporcionada por PB a los subprocesos EN, MA , y PC se efectuaran siempre que $\pi_5(s)\sigma \in \mathcal{K}_5, \sigma \in \Sigma_5$. Mientras, este requerimiento se cumpla se generará un conjunto de estatus Q_{p_5} obtenidos por consenso y su dinámica puede ser representada gráficamente (ver Fig. 17).

Donde $\mathcal{L}_{p_5} = (\sigma_5\sigma_{13})^2 + (\sigma_5\sigma_{19})^2(\sigma_5\sigma_{23})^1$; es el lenguaje de planificación local.

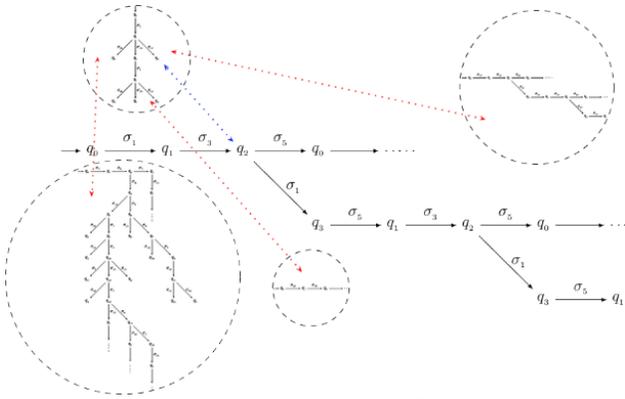


Figura 18. Muestra parcial de la construcción de $\mathcal{P}(G)$.
Fuente: Los autores.

6.5. Planificación por consenso

Finalmente, para el sistema de producción de pintura la sincronización de los subsistemas (PB, EN, MA y PC) es establecida desde la agregación de la base blanca por parte de PB y el posterior uso de la misma por parte de EN, MA y PC (ver Fig. 18).

Así, la planificación $\mathcal{P}(G)$ es construida desde la sincronización de las planificaciones locales $\mathcal{P}(G_i) = (Q_i, \Sigma_i, \delta_i, J_i, q_{0_i}, Q_{p_{im}})$, con $i = 1, \dots, 5$; sobre PB, MA, PC, EN y Re respectivamente y está dada por $\mathcal{P}(G) = (Q_{p_1} \times \dots \times Q_{p_5}, \Sigma, \delta, J, (q_{0_1}, \dots, q_{0_5}), Q_{p_{1m}} \times \dots \times Q_{p_{5m}})$ donde

- $Q_{p_1} \times \dots \times Q_{p_5}$ es el conjunto de estados;
- Σ es el alfabeto y representa a $\Sigma_1 \cup \dots \cup \Sigma_5$;
- $(q_{0_1}, \dots, q_{0_5})$ es el estado inicial;
- $Q_{p_{1m}} \times \dots \times Q_{p_{5m}}$ es el conjunto de estados finales;
- $\delta((q_1, \dots, q_5), \alpha) = (P_1, \dots, P_5)$, $P_i = \delta_i(q_i, \alpha)$, y $P_j = q_j$, si para $i, j = 1, \dots, 5$; $\alpha \in \bigcap_i J_i(q_i) \setminus \bigcup_{j:j \neq i} \Sigma_j$,

e indefinida en otro caso;

- $J(q_1, \dots, q_5) = \bigcap_i J_i(q_i) \setminus \bigcup_{j:j \neq i} \Sigma_j$; $i, j = 1, \dots, 5$;
- el cual tiene como lenguaje (de planificación) $\mathcal{L}_p = \mathcal{L}_{pm} = \bigcap_{i=1}^5 \pi_i^{-1}(\mathcal{L}_{pm_i})$.

7. Conclusión

La descomposición de una planificación en planificaciones locales, direcciona el análisis global desde un punto de vista local, lo cual permite bajo modularidad, unificación de criterios y juicios, flexibilidad, capacidad de gestión, viabilidad e implementación reducida; en consecuencia, un proceso ágil desde el punto de vista de automatización. En efecto, las descomposiciones son

$$\mathcal{L}_p = \bigcap_{i=1}^n \pi_i^{-1}(\mathcal{L}_{p_i}) \quad \text{y} \quad \mathcal{L}_{pm} = \bigcap_{i=1}^n \pi_i^{-1}(\mathcal{L}_{pm_i}) \quad (12)$$

expresadas por lenguajes cerrados y \mathcal{L}_{pm} -cerrados respectivamente; de donde, se avanza modularmente a la reducción e implementación óptima. A propósito, enfatizamos que la planificación en SED siempre será una intención que posiblemente no pueda ser llevada a cabo completamente; por ejemplo, cuando estamos en teoría de control, debido a los eventos no controlables. Por otro lado, el modelo construido para la clase de sistemas de manufactura con componentes cooperantes (unidades de producción) puede ser considerado para el estudio de las k-interacciones cuando las unidades del sistema toman opciones cooperativas. Más precisamente, si el número de opciones cooperativas crece, entonces seguramente se generarán nuevas interacciones indeseables entre estas, produciendo problemas de funcionalidad en el sistema; pero nuevamente será establecido el acoplamiento entre las componentes por consenso (reconfiguración del sistema), con lo cual podríamos abordar propiedades de seguridad y no bloqueo desde un punto de vista local.

En relación a los trabajos futuros se puede considerar el desarrollo de programas de software para ejecutar cálculos referentes al algoritmo de planificación. También, comparar la planificación por consenso con diseños establecidos por Redes de Peri, representadas por bases de Gröbner, por razones de explosión de estados. Finalmente, se pueden extender nuestros argumentos teóricos de planificación, como una herramienta a un nivel de abstracción temporal para el análisis de SED.

Bibliografía

- [1] Aldaniyazov, K.N., Main factors for the improvement of a complex system of strategic production cost management. Revista Espacios, 39(11), pp. 29, 2018.
- [2] Al-Jaar, R.Y. and Desrochers, A.A., Petri nets in automation and manufacturing. Advances in Automation and Robotics, 2, pp. 153-225, 1990.
- [3] Andrade, H.deS. and Loureiro, G., Proposta de um modelo de planejamento estratégico baseado em engenharia de sistemas. Revista Espacios, 39(13), pp. 10, 2018.
- [4] Arias-Bolzmann, L., Agurto, W.F., Chávez, Á.M., Pantoja, R. and Pinto, A., Decisions in hierarchical production planning: goals, heuristics and bias. Revista Espacios, 30(14), pp. 3, 2018.
- [5] Caspi, P., Model of discrete event systems in computer science, Proceedings of the European Control Conference (ECC 91), Grenoble, France, 1991.
- [6] Dante, O., Coromings, A. and Lusa, A., Estado del arte sobre planificación agregada de la producción. Tesis Doctoral, Instituto de la Organización y Sistemas Industriales, Universidad Politécnica de Cataluña, Barcelona, 2007.
- [7] Eilenberg, S., Automata, languages and machines. Academic Press, New York, 1974.
- [8] Gradišar, D. and Mušič, G., Production-process modelling based on production-management data: A Petri-net approach. International Journal of Computer Integrated Manufacturing, 20(8), pp. 794-810, 2007. DOI: 10.1080/09511920601103064.
- [9] Guo-zhu, J.I.A. and Cheng, Yan., Reengineering method of production system based on theory of constraint and system dynamics. Computer Integrated Manufacturing Systems, 4, 2007.
- [10] Hopcroft, J.E., Motwani, R. and Ullman, J.D., Automata theory, languages, and computation. Addison Wesley, Press Reading, Massachusetts, 1979.
- [11] Lapulus, E., Fang, S., Rang, C. and Van-Gerwen, R., Manufacturing Integration, Computers in Industry, 1995.
- [12] Marsan, M.A., Balbo, G., Bobbio, A., Chiola, G., Conte, G. and Cumani, A., The effect of execution policies on the semantics and

- analysis of stochastic Petri nets. IEEE Transactions on Software Engineering, 15(7), pp. 832-846, 1989. DOI: 10.1109/32.29483
- [13] Mata, G., Lugo, A. and Rojas, G., Aplicación de bases de Gröbner en el problema de alcanzabilidad de estados de sistemas de eventos discretos modelados por redes de Petri. Lecturas Matemáticas, 37(1), pp. 5-23, 2016.
- [14] Murata, T., Petri Nets: Properties, analysis and applications, Proceedings of the IEEE, 77(4), 1989. DOI: 10.1109/5.24143
- [15] Peterson, J., Petri Net Theory and the Modeling of Systems. Prentice Hall, PTR Upper Saddle River, NJ, USA, 1981.
- [16] Wonham, W., Design Software: TCT [Online]. Systems Control Group, Dept. of ECE, Univ. of Toronto, 2015. Available at: <http://www.control.toronto.edu/DES>.
- [17] Yan, H.S., Zhang, X.D. and Jiang, M., Hierarchical production planning with demand constraints. Computers & Industrial Engineering, 46(3), pp. 533-551, 2004. DOI: 10.1016/j.cie.2004.01.012

G.E. Mata-Díaz, received a Dr. degree in mathematical sciences: Doctor Applied Science, Universidad de Los Andes, ULA, Mérida-Venezuela in 2016. From 1997, he is currently a full professor in the department of mathematics at Universidad de Los Andes, Venezuela. Has experience in mathematics, with emphasis on dynamic systems of discrete events (analysis and control).

ORCID: 0000-0001-7147-1422

L.B. Ruiz-Leal, received a PhD degree in mathematical sciences: Doutor em Ciências, from the National Institute of Pure and Applied Mathematics: Instituto Nacional de Matemática Pura e Aplicada IMPA Rio de Janeiro-Brasil in 2005. From 2005, he is currently a full professor in the department of mathematics Universidad de Los Andes, Venezuela. Has experience in Mathematics, with emphasis on ergodic theory and hyperbolic dynamical systems. His research interests include: transitive and robustly transitive invariant sets, infinite ergodic theory and symbolic dynamics.

ORCID: 0000-0002-7737-3847

C.C. Camacho-Camacho, received a MSc. degree in alternative pedagogy in the teaching of mathematics from Universidad Politécnica Territorial de Mérida "Kleber Ramírez", Venezuela in 2018. Has experience in mathematics, with emphasis on teaching systems and didactic.

ORCID: 0000-0003-3630-2812

A.J. Méndez-Urrieta, received a Dr. degree in mathematical sciences: Doctor Applied Science, Universidad de Los Andes, ULA, Mérida-Venezuela in 2017. Has experience in mathematics, with emphasis on Control engineering and automation.

ORCID: 0000-0002-6406-6805

S. Muñoz, received a PhD degree in mathematical sciences: Doutor em Ciências, from the National Institute of Pure and Applied Mathematics: Instituto Nacional de Matemática Pura e Aplicada IMPA Rio de Janeiro-Brasil in 2005. From 2017 he is currently a full professor in the department of mathematics, physics and computers (Dpto. de Matemática, Física e Computação DMFC) at the university: Universidade do Estado do Rio de Janeiro UERJ Resende, Rio de Janeiro, Brasil. Has experience in Mathematics, with emphasis on ergodic theory and dynamical systems. His research interests include: transitive and robustly transitive invariant sets, infinite ergodic theory and symbolic dynamics.

ORCID: 0000-0002-6952-6102

H.B. Zambrano-García, received the Lcdo. Mathematics in 1994, from the Universidad de los Andes. Mérida, Venezuela. Has experience in mathematics, with emphasis on programmer in systems, and in control and automation. His research interests include: analysis of complex systems and control of hybrid systems.

ORCID: 0000-0002-6059-5136



UNIVERSIDAD NACIONAL DE COLOMBIA

SEDE MEDELLÍN
FACULTAD DE MINAS

Área Curricular de Ingeniería
de Sistemas e Informática

Oferta de Posgrados

Especialización en Sistemas
Especialización en Mercados de Energía
Maestría en Ingeniería - Ingeniería de Sistemas
Doctorado en Ingeniería- Sistemas e Informática

Mayor información:

E-mail: acsei_med@unal.edu.co
Teléfono: (57-4) 425 5365