SAFET: SISTEMA PARA LA GENERACIÓN DE APLICACIONES DE FIRMA ELECTRÓNICA

V. Bravo Bravo[1], A. Araujo Brett[2], Centro Nacional de Desarrollo e Investigación en Tecnologías Libres, MPPCTI. Venezuela.

Recibido Noviembre 16, 2011 - Aceptado Marzo 4, 2012

http://dx.doi.org/10.18566/puente.v6n1.a05

***Resumen*— SAFET* es un sistema que está conformado por un analizador de lenguaje XML/DTD y un conjunto de aplicaciones, todo ello integrado con tecnologías de firma electrónica y estampillado de tiempo. En este sentido, se propone la vinculación natural del concepto de flujo de trabajo con los procesos de firma electrónica y posteriormente el de estampillado de tiempo, lo que aporta nuevos elementos en la consolidación de un patrón para la construcción de este tipo de aplicaciones. Bajo este enfoque la firma electrónica no se muestra como una sustitución completa de la firma manuscrita, sino como un elemento tecnológico de la era digital con carácter jurídico que es complementario, y que permite realizar operaciones que resultarían difíciles o imposibles de ejecutar de forma segura con elementos no digitales. SAFET genera grafos y reportes en diferentes formatos de archivo. En la última sección se presenta una aplicación con firma electrónica generada utilizando el sistema.**

***Palabras clave*— Firma electrónica, Flujos de Trabajo, XAdES, ICP, lenguaje XML.**

***Abstract*— SAFET is a system that consists of a language parser XML/DTD and applications integrated with electronic signature technology. We propose the natural linkage of the workflow concept with electronic signature concept, which adds new elements in the consolidation of a pattern for develop such applications. Under this approach the electronic signature is not a complete replacement of handwritten signature, it's a technological element of the digital age with legal basis, complementary, and allows operations that would be difficult or impossible to run safely with non-digital terms. SAFET generates graphs and reports using different file formats. The final section presents a software application with electronic signature generated using the system.**

[1] V. Bravo Bravo, Centro Nacional de Desarrollo e Investigación en Tecnologías Libres, MPPCTI. Venezuela; E-mail: vbravo@cenditel.gob.ve

[2] A. Araujo Brett, Centro Nacional de Desarrollo e Investigación en Tecnologías Libres, MPPCTI. Venezuela; E-mail: aaraujo@cenditel.gob.ve

* SAFET (Sistema Automatizado para la Firma y Estampillado Electrónico de Tiempo, http://seguridad.cenditel.gob.ve/safet)

***Keywords*— Electronic signature, workflow, XAdES, PKI, XML.**

I. INTRODUCCIÓN

Cualquier organización tiene definido procesos que agrupan sus actividades diarias, entre las cuales generalmente se incluyen el manejo y procesamiento de datos y el resumen de información para la toma de decisiones. Las actividades cotidianas requieren tiempo y están supeditadas a condiciones específicas para su correcta ejecución. En relación a ello, las tecnologías de la información generalmente sirven como mecanismos para la mejora de procesos.

Por otro lado, la utilización de mecanismos que proporcionen autoría, integridad y ocurrencia en el tiempo de un proceso específico asociado a documentos electrónicos también se considera como un beneficio dentro de las organizaciones. En respuesta a esto, las tecnologías de firma electrónica y estampillado de tiempo proporcionan mecanismos para garantizar el intercambio de información de forma segura [2].

Este trabajo muestra la estructura y funcionalidades de una herramienta que responde a las situaciones planteadas anteriormente. Uno de los propósitos principales, es presentar un sistema computacional que flexibilice la comunicación entre personas y sistemas de información a través del uso de un lenguaje específico para flujos de trabajo, utilizado para la construcción de vistas inteligibles, ordenadas y adaptadas de datos. Es importante señalar que en esta propuesta, interviene la noción de firma electrónica, un elemento de confianza e identificación legal en Internet, aceptada culturalmente en su definición primigenia.

SAFET es una herramienta que tiene funcionalidades relacionadas con flujos de trabajo, firma electrónica, como también capacidades de estampillado de tiempo. Escrita mayoritariamente en

lenguaje C++, actualmente se encuentra liberada la versión 0.1.beta bajo licencia GPL v2.0 disponible en Internet. En los párrafos que siguen se exponen aspectos teóricos y prácticos vinculados con la descripción y arquitectura del sistema. La implementación del concepto de estampillado de tiempo se hereda de la infraestructura propuesta por el esquema XAdES y particularmente, de una de sus implementaciones denominada *digidoc* [13]. El estampillado de tiempo se trata como una cualidad que forma parte del documento firmado electrónicamente, y su certificación es responsabilidad del servidor de validación de certificados en línea OCSP ("Online Certificate Status Protocol" en inglés).

Uno de los retos del trabajo propuesto es lograr la integración del concepto de firma electrónica con otros modelos y tecnologías utilizadas en la automatización de procesos; inicialmente se tienen los flujos de trabajos. Esto implica, como parte del desarrollo de este trabajo la definición de un lenguaje computacional y la correspondiente implementación de un analizador léxico y semántico que permita el modelado y la elaboración de aplicaciones que usen como elemento principal la firma electrónica, lo cual se traduce en la eliminación de fases lentas producto de la eliminación del uso de medios físicos (firma manuscrita) durante un proceso automatizado. En este sentido también hay que decir, que SAFET todavía no dispone de todos los patrones utilizados por el modelo BPM (*Business Process Management*, en inglés), pero incluye patrones compatibles que pueden considerarse como básicos. Por otro lado, se provee de Interfaces Gráficas de Usuario (IGU) y paquetes de librerías de software para la construcción de modelos que permiten la generación expedita de reportes y grafos (un grafo generado por SAFET se aprecia en la fig. 3).

II. BASES TEÓRICAS

Con el objetivo de contextualizar la propuesta se describe brevemente algunos conceptos utilizados en el diseño e implementación del sistema, tales como la idea de flujos de trabajo basado en Redes de Petri, las bases de datos relacionales y firma electrónica y el modelo de confianza basado en Infraestructura de Clave Pública (ICP). Es imprescindible en una primera aproximación: discutir las nociones teóricas que acompañan al proceso de definición del sistema.

A. Flujo de Trabajo basado en Redes de Petri

Las ideas sobre Redes de Petri se iniciaron en los años 60, y se planteaban como una herramienta para resolver problemas en el área de los sistemas distribuidos. Una Red de Petri es un lenguaje para modelado matemático compuesto por elementos tales como transiciones, nodos, arcos dirigidos, y fichas o marcas ubicadas en cada nodo [1]. La configuración de las marcas o fichas pueden variar según eventos, estados o tiempo. Formalmente se puede describir una Red de Petri como una tupla (S,T,F,m_0) constituida por nodos, transiciones, movimientos y una condición inicial respectivamente.

Se han realizado diferentes trabajos que extienden la definición formal de una Red de Petri. Una de estas extensiones es la Red de Petri Coloreada [1] (*Coloured Petri Net* en inglés), que agrega el hecho de poder identificar a cada ficha por un determinado color, convirtiendo las fichas en elementos distinguibles unos de otros. Otra de las extensiones recientes es el propuesto por Hofstede et al. en [3], donde se describe el lenguaje denominado YAWL, que entre otras características incorpora el uso de los operadores *XOR*, *AND*, *OR* de entrada (*JOIN*) y salida (*SPLIT*) para modelar la concurrencia y paralelismo entre eventos o actividades.

B. Base de Datos Relacionales

Las Base de Datos Relacionales (BDR) son un fundamento en la construcción de sistemas de información modernos y profesionales. Este hecho se puede atribuir entre otras causas, a la disponibilidad de un Lenguaje de Consulta estructurado llamado SQL (Structured Query Language en inglés), que prácticamente se ha convertido en un estándar para aplicaciones de gestión de datos [4] y [5]. Por otro lado el desarrollo a través de muchos años de algoritmos para mejorar la eficiencia computacional de los denominados gestores o servidores de base de datos relacional, es otro argumento que apoya este hecho.

Las operaciones básicas o primitivas para el álgebra relacional son las siguientes:

Selección: con SQL se puede utilizar la cláusula WHERE): obtiene un subconjunto (formado por tuplas) de la relación evaluada

Proyección: con SQL se puede utilizar la cláusula SELECT: obtiene un subconjunto de Atributos

Concatenación: con SQL se puede utilizar la cláusula JOIN: teniendo las relaciones **R** y **S**, la

operación de concatenación obtiene una relación formada por tuplas donde los atributos comunes de **R** y **S** coinciden en los valores, las tuplas resultantes incluyan la unión de todos los atributos de **R** y **S**.

Otras operaciones son el Producto Cartesiano, Diferencia, Unión e Intersección (heredadas de la teoría de conjuntos) [6].

Es importante señalar, que generalmente en cada relación se nombra un atributo como clave principal, cuya función es identificar unívocamente a cada tupla y evitar la duplicidad de datos, además de servir como elemento de base para algoritmos de verificación y simplificación de relaciones.

C. Firma Electrónica e Infraestructura de Clave Pública

Se denomina Infraestructura de Clave Pública (ICP) a la combinación de hardware y software, políticas y procedimientos que permiten asegurar la identidad de los participantes en un intercambio de datos usando criptografía de clave pública, y que representa para una organización un tercero de confianza [7]. La ICP se utiliza ampliamente en Internet para mantener la seguridad y confidencialidad de los datos, esto es, se encarga de emitir y gestionar elementos de identificación electrónica conocidos como certificados electrónicos para personas, dispositivos de red, direcciones IP y sitios Web que realizan transacciones digitales.

La firma electrónica es una herramienta que se utiliza como prueba de consentimiento sobre la información que contiene un documento, además de proveer identificación de un autor o persona responsable sobre determinado documento electrónico. Desde esta perspectiva tiene el mismo objetivo que la firma autógrafa. La tecnología de firma electrónica utiliza una ICP, ya que es necesario que tanto los emisores como los receptores de información electrónica dispongan de un tercero de confianza a través de Internet [7].

D. Lenguajes basados en flujo de trabajo

Uno de los elementos principales de esta propuesta es el modelado de los problemas de gestión de información utilizando flujos de trabajo. Por ello se debe considerar el acoplamiento de la firma electrónica con aplicaciones que utilicen conceptos como BPM. Esta tecnología ha servido como solución al modelado rápido de procesos de negocios, donde lenguajes como BPEL [8], extraen elementos como actividades, tareas y condiciones, y a las cuales se les puede agregar los eventos de aceptación o acuerdo relacionados con la firma electrónica. También, dentro de la gama de aplicaciones, se debe contar herramientas que permitan a usuarios y administradores realizar las tareas usuales en el manejo de documentos: la visualización del documento, la visualización de los datos vinculados a los eventos de firma y certificados, así como su segura verificación y validación.

III. TRABAJOS RELACIONADOS

Existen numerosas herramientas que han trabajado los problemas vinculados con el uso de flujos de trabajos y modelado BPM para la solución de necesidades de información de los integrantes de cualquier organización. Por otro lado, existen herramientas que usan el concepto de firma electrónica, que tienen como objetivo aportar soluciones para políticas de ahorro de papel, el incremento de la fluidez burocrática, la adición de vinculación jurídica, entre otros propósitos.

Una de las herramientas que contiene más características en el ámbito de construcción de sistemas de información basados em flujos de trabajo es *Bizagi*. Este herramienta comercial de código fuente propietario, y disponible para sistemas operativos Windows implementa en gran medida el modelo BPM, además dispone de diversas herramientas que presentan una interfaz intuitiva para el modelado de procesos, la construcción de formularios de ingreso y modificación de datos, la elaboración de reportes, y la conexión con sistemas de servicios web (*WebServices*) y de correo electrónico, entre otros.

Uno de los sistemas actuales de firma electrónica más innovadores es *Xyzmo SIGNificant.* Una aplicación comercial de código fuente propietario, que introduce elementos interesantes en el área de ergonomía y adaptación al cambio: no obliga a aprender una nueva técnica de firma sino que ofrece a los usuarios de esta tecnología el uso la firma manuscrita a través de una tableta electrónica o teléfono con interfaz multitoque (*multitouch*) bajo sistemas operativos Android o iOS, esto sin desvincularse del esquema de confianza de Infraestructura de Clave Pública (*PKI*).

En relación con las dos aplicaciones descritas anteriormente, este trabajo muestra una herramienta que agrega la integración de los conceptos de flujos de trabajo y firma electrónica (no los provee separadamente), así como también la disponibilidad de un lenguaje flexible y original basado en XML/DTD. Por último, se puede nombrar como ventaja que el código fuente de SAFET es de tipo código libre (*opensource*), lo que es una cualidad apreciable en los procesos de formación de comunidades de investigadores, programadores y usuarios. En este punto, se puede señalar una desventaja de SAFET con respecto a otras aplicaciones es el hecho de que estas aplicaciones generalmente disponen de interfaces más intuitivas y ergonómicas.

IV. ARQUITECTURA DE SAFET

En esta sección se muestra la estructura y los esquemas de coordinación entre los componentes de SAFET. Se describen sus elementos: el analizador de lenguaje para especificación de flujo de trabajo, la conexión ubicua a bases de datos relacionales vinculadas a las fichas, el motor de cálculo de documentos "libsafet", el visualizador de grafos (implementado utilizando graphviz), el cliente de escritorio "inflow" y finalmente los envoltorios "wrappers" para el lenguaje python "websafet". La figura 1 muestra un diagrama de componentes del sistema.

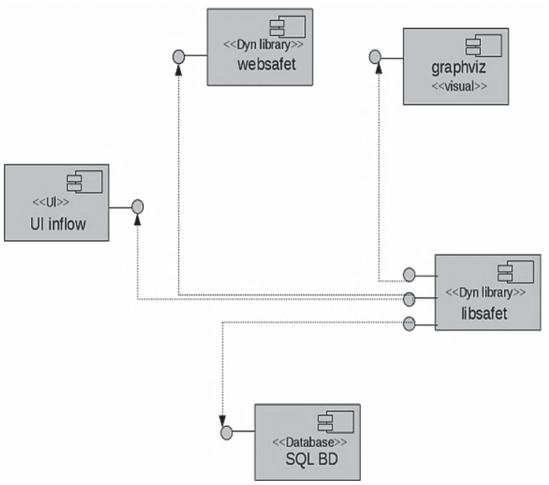

Fig. 1. Componentes de la plataforma SAFET en notación UML

A. Analizador de Lenguaje XML/DTD

Uno de los elementos principales del sistema es el lenguaje de descripción basado en flujos de trabajo, y que forma parte del componente libsafet (Ver Fig. 1). Para su implementación se hace uso de la tecnología XML/DTD [9] [10]. Un archivo en formato DTD determina una sintaxis a seguir para un conjunto de archivos XML que es equivalente a la definición de una gramática o autómata de pila [11]. El código de la Tabla I contiene algunas etiquetas que pueden estar presentes y se utilizan para construir en memoria una estructura de datos que representa una Red de Petri Coloreada, que posteriormente puede ser dibujada por el componente graphviz como un grafo (Ver figura 3). La especificación completa de las etiquetas del lenguaje de descripción de flujo de trabajo puede verse en [15]. En el trozo de documento que se muestra en el código de la Tabla I se tienen las etiquetas: workflow que agrupa todos los demás elementos; la etiqueta task que especifica lugares o actividades; Variable se utiliza para mostrar una lista de documentos relacionado con la tarea o actividad y el atributo query de la etiqueta Port se utiliza para especificar el evento o restricción de salida/entrada entre tareas utilizando sentencias del lenguaje SQL [5]. La palabra SIGN, agregada por el analizador SAFET a las sentencias SQL, puede ser utilizada en las expresiones de consulta para especificar transiciones basadas en el uso de firmas electrónicas. Esta palabra indica que un conjunto de documentos firmados electrónicamente por un determinado Nombre Común (ver [2]) pasan o no pasan a una siguiente tarea. Para la implementación de los mecanismos de firma electrónica se utiliza la biblioteca *digidoc* que sigue la especificación de Firma XML Avanzada *OpenXAdES* (ver [13]). Esta especificación indica la forma en que se realiza y verifica una o varias firmas sobre un documento electrónico validado por una ICP. Por último, la etiqueta *connection* representa la conexión o arcos entre tareas.

TABLA I.
PARTE DE UN DOCUMENTO DE FLUJO DE TRABAJO SAFET.

```
<workflow id="nombre" desc="nombre"> ... (sigue)
<task id="nombre"  title="nombre">
<port  query="SELECT <campos> FROM <tabla>
SIGN <firmante_nombre_comun>"
tokenlink="[claveforanea1:claveforanea2]">
<connection source="nombreTarea"/>  </port>
<variable id="nombre"
tokenlink="[claveforanea1:claveforanea2]"
documentsource="SELECT <campos>,
FROM <tabla> "> ...(sigue)
```

B. Conector con Bases de datos Relacionales

Otro de los elementos que se toman para la definición del sistema es la vinculación con

esquemas de bases de datos relacionales (*SQL DB*). Las fichas de las Redes de Petri, se representan como enlaces a una clave principal dentro de una determinada relación. El atributo *tokenlink* de los elementos *Port* o *Variable* proporciona este enlace. Es importante señalar el uso del lenguaje SQL para especificar consultas a repositorios de datos relacionales dentro de la definición del flujo de trabajo. Los documentos son tuplas que son generadas a partir de consultas a las fuentes de datos, y que tienen una vinculación con la clave principal del elemento *token*. Cada clave diferente representa un color de ficha en la Red de Petri coloreada. En el Código de la Tabla I no se muestra el elemento *pattern*, ya que para utilizarlo se necesita que desde o hacia una tarea existan dos o más transiciones. Los parámetros de conexión al gestor de base de datos así como otros valores necesarios para la ejecución de las funcionalidades del sistema se especifican en un archivo de configuración denominado *safet.conf*. Actualmente SAFET puede conectarse a los gestores de base de datos *PostgreSQL*, *MySQL* y *SQLite*.

C. Motor de cálculo "libsafet"

El motor de cálculo de flujos de trabajo se implementa a través de una librería dinámica, esto es, una pieza de software que incluye todos los objetos y funciones basadas en el esquema propuesto necesarias para la ejecución de consultas al sistema de información (Ver Figura 1). Este motor realiza evaluaciones en serie o de forma paralela de las condiciones de entrada y salida especificadas en un determinado documento de flujo de trabajo a utilizar compatible con la especificación XML/DTD descrita en [15]. Para ello el motor va evaluando secuencialmente las consultas escritas en lenguaje "SQL estándar más clausulas SAFET" (por ejemplo se puede utilizar la clausula *SIGN* agregada a una expresión SQL) especificadas en los atributos *query* de los elementos *Port* (ver Tabla I). El orden de la evaluación se realiza desde la condición inicial hasta la condición final.

En las actividades de cualquier organización es frecuente que muchas situaciones se parezcan y hacen que los objetos de información se comporten de forma similar. Para este caso es aplicable el uso de patrones de comportamiento. SAFET utiliza una lista de patrones básicos que engloban el modelado de una gran cantidad de casos reales. Los patrones actúan en las transiciones y son de dos tipos: salida (SPLIT) y entrada (JOIN). Cada uno tienen tres tipos de comportamiento expresados a través de operadores: Y Lógico (AND), O Lógico (OR) y O Lógico Exclusivo (XOR). Para la utilización de un patrón es necesario que un puerto (Port) tenga conexiones (connection) a dos o más tareas, ya que son operadores binarios (ver [3]). Por ejemplo, si se necesita obtener un listado de todas las solicitudes firmadas por el Director de Presupuesto de una determinada organización o firmadas por el Director Técnico es posible definir una tarea (Task) que utilice un patrón de conjunción (JOIN), y el operador lógico "o" (OR). El objeto Variable de la nueva tarea contendrá la suma de las solicitudes que han sido firmadas por al menos uno de los dos directores.

Por otro lado, el motor de cálculo también incluye un módulo para el procesamiento de formularios que puedan ser utilizados en aplicaciones de escritorio o Web. Es posible definir en estos formularios la ejecución de firmas electrónicas por parte de los usuarios, lo cual se mostrará cuando los objetos de información se encuentren en una determinada tarea. De esta manera las acciones de adición, consulta, modificación y eliminación de datos de los repositorios de información pueden ser definidos utilizando un lenguaje basado en XML/DTD y cuya especificación completa se encuentra en [16].

Debido a que el motor de cálculo se construye como un archivo de tipo librería dinámica (con extensión .dll o .so) que contiene una interfaz exportable en C/C++, es posible elaborar aplicaciones clientes utilizando diferentes lenguajes de programación o plataformas. Esto tomando en cuenta que previamente se deben construir los correspondientes envoltorios por lenguaje o plataforma para el uso de SAFET. En las secciones siguientes IV.D y IV.E se muestran dos ejemplos de ello.

D. Cliente de escritorio "inflow"

Es una IGU desarrollada para el sistema operativo Linux que incluye los módulos "Formulario" para el ingreso, modificación, eliminación de datos y aplicación de firma electrónica, el módulo "Consulta" para visualizar de forma gráfica o textual datos vinculados con firma electrónica, el módulo "Firma/Certificados" que contiene funciones de firma y validación de documentos firmados en el formato de Firma Electrónica Avanzada, y otros módulos para configuración gestión de usuarios. En

la Fig. 2 se muestra la pantalla principal de la aplicación. Esta aplicación utiliza el motor de cálculo (librería dinámica) descrito en los párrafos anteriores y admite diversas configuraciones, las cuales incluyen formularios, consultas, flujos de trabajos, gestión de usuarios y opciones gráficas, y que son distribuibles en archivos compactos (.tar). La acción de carga de una de estas configuraciones convierte al cliente "inflow" en una específica aplicación con firma electrónica, por esta razón se diseña una única aplicación que incluye todos estos módulos. Esto no prohíbe el desarrollo de otras aplicaciones adaptadas a necesidades específicas que puedan utilizar el componente libsafet.

E. Envoltorio para python (websafet)

Es un paquete (librería) disponible para el lenguaje de programación Python, que exporta las clases principales del motor de cálculo SAFET. En este paquete también se incluye componentes para el navegador Mozilla/Firefox versión 3.5 o superior que permite firmar y validar documentos electrónicos de forma remota bajo el formato XAdES.

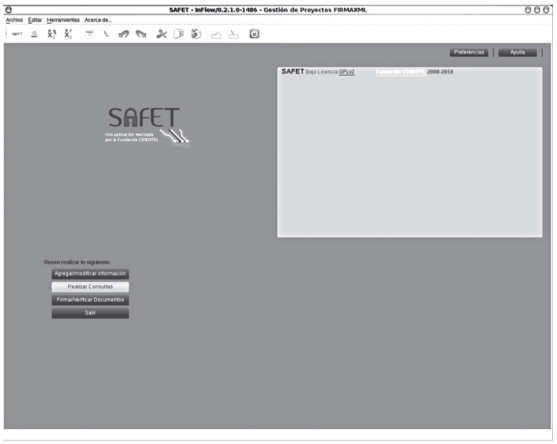

Fig. 2. Pantalla principal de la aplicación "inflow"

Es posible definir una aplicación web con firma electrónica utilizando websafet dado que la librería incluye funciones para generar código HTML/Javascript correspondiente a formularios de ingreso y modificación de datos, reportes y grafos que responderían a una determinada necesidad de información.

V. APLICACIÓN DE PRUEBA

En este apartado se muestra la ejecución de SAFET con un documento de flujo de trabajo de ejemplo, se describe el problema a modelar y se discute sobre los resultados obtenidos.

A. Caso de "Solicitud de viáticos"

Para ilustrar la generación de una aplicación de firma electrónica utilizando SAFET se realizó una configuración para un proceso que gestiona la aprobación o denegación de solicitudes de viáticos y pasajes dentro una institución.

El flujo de actividades para este ejemplo se explica a continuación (ver Fig.3):

1. Un empleado se selecciona para realizar un viaje nacional, en razón de la realización de actividades asignadas.

2. El empleado debe llenar una planilla P con sus datos personales, y los datos del viaje: cédula, nombres, cargo, fecha de ida y vuelta del viaje, ciudad destino y la descripción de actividades a realizar.

3. La planilla P debe ser aprobada (firmada) por el director del departamento correspondiente. El director de departamento debe anexar a P en el cuadro "Observación" los argumentos que respalden la realización del viaje. En este punto se obtiene la Planilla P_{jd}

4. La planilla P_{jd} debe ser aprobada (firmada) por el director de presupuesto. El director puede denegar la petición, en el caso de que exista un problema con los recursos económicos necesarios para el viaje. En el caso de ser aprobada se obtiene la planilla P_{dp}

5. La planilla P_{dp} debe ser aprobada (firmada) por el Presidente de la institución no se aprobaría, si el Presidente decide discrecionalmente no hacerlo. Finalmente si la solicitud ha sido aprobada por el Presidente se obtiene la planilla P_P y seguidamente se realiza una transferencia desde la cuenta bancaria de la institución a la cuenta nómina del empleado, según lo estipulado en P_P.

6. Todas las decisiones y acciones llevadas a cabo en este flujo son objeto de auditoría.

Ya definido el proceso de solicitud, se procede a modelar las actividades utilizando un flujo de trabajo lineal (ver Fig. 3). Los nodos "Director", "Presupuesto" y "Presidente" se vinculan con el uso de la firma electrónica del director de

departamento, director de presupuesto y Presidente de la institución respectivamente. El nodo "Solicitado" indica que la planilla ha sido introducida correctamente por el empleado.

Para el desarrollo de esta aplicación, es requisito inicial que los usuarios participantes cuenten con el hardware apropiado, es decir tarjetas inteligentes que aseguren por lo menos un nivel 2 de autenticación [14]. Para este caso, todos los empleados poseen una tarjeta inteligente certificada que les permite firmar electrónicamente. Cada una de las tarjetas contiene un certificado digital expedido por una ICP que valida su identidad digital.

Seguidamente, es necesario modelar el proceso de solicitud de viáticos a través de un lenguaje XML de definición que evaluará el motor de cálculo [15], para luego desplegarlo, y que los usuarios puedan realizar las acciones correspondientes. El documento firmado que se maneja en este caso es la planilla P que introduce el empleado, y que luego debe ser procesado en orden por el director de departamento, el director de presupuesto, y el Presidente de la institución. En este punto se escribe el documento XML de flujo de trabajo. En este se especifican las tareas y las transiciones que componen el modelo. El código de la Tabla II muestra la definición de la tarea Solicitud.

Para esta tarea (Task) se especifica dos elementos: El puerto (Port) que filtra las fichas hacia la siguiente tarea a través de la una expresión SQL con cláusula SIGN, y la variable, que es un contenedor de las fichas de la tarea.

TABLA II.
ESPECIFICACIÓN EN EL LENGUAJE XML/DTD DE LA TAREA "SOLICITADO"

```
<task id="Solicitado"  title="en_evaluacion" >
          <port side="forward" type="split"   >
    <connection source="Director" query="VariableSolicitado
SIGN Antonio Araujo Brett;Victor Bravo" options=""
tokenlink="[cedula,cedulaviajero:viaticos]">
</connection>
  </port>
<variable id="VariableSolicitado" scope="task"  type="sql"
config="1"   tokenlink="[cedula,cedulaviajero:viaticos]"
documentsource="select
viaticos.id,nombres,apellidos,horasalida,horallegada,proyecto,ciud
ad from personal" rolfield="cedulaviajero"
timestampfield="fechasolicitud">
</variable>
</task>
```

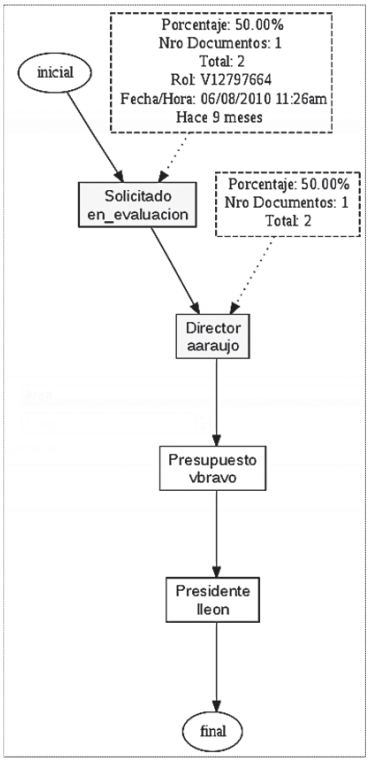

Fig. 3. (Grafo) Diagrama del flujo de trabajo generado por inflow para el caso de gestión de viáticos

Se utiliza la aplicación "inflow" para el uso de firma electrónica con flujos de trabajo (Ver Fig. 2). "inflow" admite múltiples configuraciones (múltiples aplicaciones). En este se especifican los parámetros de conexión a la base de datos, así como otros datos propios de la aplicación a configurar, en este caso, la gestión de solicitudes de viáticos. Con "inflow" los usuarios autorizados (Solicitante, Director, Jefe de Presupuesto o el Presidente) están habilitados según su orden de acceso a la planilla, a firmar y validarla.

También se coloca a disposición un portal Web utilizando websafet donde es posible verificar la planilla, así como también los datos de la firma como fecha, hora, lugar, nombres completos y cargo del firmante entre otros.

Como se muestra en la Fig. 3, es posible obtener el flujo burocrático que ha tenido la planilla P, y sus siguientes transformaciones: "Director "Pjd, Presupuesto Pjp,"Presidente" P_P, permitiendo conocer a los interesados: el tiempo exacto de permanencia en cada estadio, el estado actual de la planilla, la cantidad de fichas (documentos) en cada tarea con su respectivo porcentaje (en relación a todo el proceso) y en general los detalles que podrían

ayudar a obtener un mejor funcionamiento del proceso en cuestión.

SAFET no provee directamente un sistema para la auditoría, pero contiene mecanismos basados en el concepto de firma electrónica que permiten la adaptación a políticas institucionales de resguardo y control de documentos digitales, entre ellos se pueden nombrar la posibilidad de contar con la firma electrónica y verificación para cada uno de los objetos de información que utiliza el sistema, así como también la disponibilidad de un módulo de registro de eventos (log)[12] que permite obtener datos tales como hora, fecha, tipo de acción y detalle de cada uno de los sucesos. Es importante subrayar, que si los certificados de firma electrónica de los usuarios del sistema son expedidos por una autoridad de certificación con validez jurídica, en este caso se hereda el carácter vinculante a la gestión total de documentos, otorgando de esta forma un soporte más fidedigno para los procesos de auditoría.

Bajo el enfoque propuesto para el caso de gestión de viáticos se obtienen algunos beneficios tales como:

Una visualización global (a través de un grafo) del estado de la solicitud por parte de todos los actores: empleados, jefes y Presidente. Usando navegador web o el cliente de escritorio "inflow" es posible ver el estado de la solicitud de viáticos cuando sea necesario (tomando en cuenta las capacidades de los servidores y servicios donde se instalen los sistemas).

La firma electrónica para esta implementación hereda las características de ubicuidad de internet: websafet incluye un complemento para navegadores web que permita la acción de firma de forma remota.

Es posible detectar fases lentas en este proceso, ya que el sistema muestra los lapsos de permanencia de las solicitudes en los diferentes nodos, así como también la cantidad de planillas que contienen cada nodo en un momento determinado (Ver Fig. 3).

En algunos países la legislación otorga vinculación jurídica a los documentos firmados electrónicamente (bajo ciertos criterios y estándares), lo que conlleva ventajas en el caso mostrado, tal como el ahorro considerable de papel debido a que al usar la firma manuscrita se hace necesario la impresión de al menos dos (2) planillas por cada solicitud.

VI. CONCLUSIONES

La construcción de aplicaciones con firma electrónica es una tarea compleja debido que se deben considerar gran cantidad de temas técnicos: ICP, tarjetas inteligentes, lectores, estándares, configuraciones, entre otros y de temas organizacionales: registro de usuarios, procesos de entrega, entre otros. Este trabajo muestra una plataforma que permite la construcción de aplicaciones que usen firma electrónica utilizando como elemento integrador el concepto de flujo de trabajo. La disponibilidad de un analizador propio para un lenguaje declarativo (XML/DTD) brinda gran flexibilidad al momento de modelar un proceso, y el hecho de que este lenguaje esté basado en flujos de trabajo extrae situaciones de los procesos en las organizaciones que pueden ser descritos a través de patrones.

La adición de firmas electrónicas, debidamente homologadas y estandarizadas, a los objetos de información otorga en algunos países un valor jurídico comparable con el que se obtiene con la firma manuscrita, y que además permite subsanar errores o fraudes que actualmente se cometen y que son propios del modelo de firma manuscrita. La visualización en línea de los procesos, el uso de módulo de registros de eventos, y el uso de firma electrónica podría mejorar el nivel de auditoría de los procesos en una organización, ya que se aportan elementos adicionales a la gestión de información usual que podrían ser relevantes en un proceso de auditoría informática.

SAFET contiene una lista de funcionalidades que responden al objetivo de ofrecer una automatización acertada y visualización de procesos o actividades, esto dicho en términos de las diversas y numerosas necesidades de información que hoy en día tiene cualquier organización. Los elementos de SAFET que en conjunto aportan una visión particular para el manejo de información son: un lenguaje original para definición de flujos de trabajo con uso de patrones vinculados a Base de Datos Relacionales; uso del lenguaje SQL para definición de eventos o transiciones en el flujo de trabajo y la disponibilidad de aplicaciones para el uso en el escritorio o a través de la web.

La integración de otros tipos de fuentes de datos como por ejemplo los servicios Web (WebServices), puede habilitar a la plataforma para abordar problemas de integración y consolidación de información.

Por último, para el caso particular del analizador XML, se puede agregar la implementación de nuevos patrones con la finalidad facilitar el modelado de otros procesos más complejos que los tratados en este trabajo.

Estas ideas podrían ser incorporadas al sistema de tal manera que el aporte aumente su significación en la tarea mayor de contribuir con la adopción de la firma electrónica como un elemento de uso extendido, mejorando de esta manera los niveles de seguridad que hoy en día presentan los sistemas de información.

REFERENCIAS

[1] K. Jensen. Coloured Petri Nets. Basic Concepts, analysis methods and practical use. EATCS monographs on Theorical Computer Science.Springer-Verlag, Berlin (1996)

[2] R. Perlman, An overview of PKI trust models. Network, IEEE (1999)

[3] A.H. ter Hofstede, , van~der Aalst, W.M.Yawl: yet another workflow language. Information Systems{30} (2005) 245-275

[4] A. De Miguel, M Piattini, E. Marcos, E. Diseño de Base de Datos relacionales. Alfaomega, Colombia (2000)

[5] S. Sumathi, S. Esakkirajan. Fundamentals Of Relational Database Management. Springer, Berlin, Alemania (2007)

[6] R. Grimaldi. Matemáticas Discretas. 3ra ed. Prentice Hall, Mexico (1998)

[7] N. Duane, , , J.PKI Brink,: Infraestructura de clave pública.McGraw-Hill (2002)

[8] Matjaz Juric and Benny Mathew. Business Process Execution for Web Services BPEL and BPEL4WS. 2 Ed. Packt Publishing.. 2006.

[9] W3C. Extensible markup language. Disponible: http://www.w3c.org/XML/ (2008)

[10] W3C. Guide to the w3c xml specification (xml spec) dtd. 2.1. Disponible: http://www.w3c.org/XML/1998/06/xmlspec-report-v21.htm (1998)

[11] D. Grune, H, Bal, C Jacobs, K Langendoen, Diseño de Compiladores Modernos. Mc Graw Hill. Madrid, España (2007)

[12] Sistema Automatizado de Firma y Estampillado Electrónico de tiempo (SAFET). CENDITEL. Disponible: http://seguridad.cenditel.gob.ve/safet (2009)

[13] Institute E.T.S Openxades. Disponible en: http://www.openxades.org (2009)

[14] Ross Anderson, Security Engineering: A guide to building dependable distributed systems. Wiley, USA (2008).

[15] DTD SAFET Specification. Disponible en: http://seguridad.cenditel.gob.ve/safet/browser/websafet/rc/yawlworkflow.dtd

[16] DTD SAFET Form Specification. Disponible en: http://seguridad.cenditel.gob.ve/safet/browser/websafet/rc/yawlinput.dtd

BIOGRAFÍA

Víctor Bravo Bravo nació en Maracaibo, Venezuela. Es Ingeniero de Sistemas y tiene una maestría en Computación de la Universidad de los Andes (ULA), Venezuela. Ha sido asesor de proyectos de la Industria Petrolera en el área de automatización y control inteligente, ha trabajado como director en importantes proyectos vinculados a procesos de CertificaciónElectrónica masiva. Ha dictado conferencias sobre temas de certificación electrónica en Colombia y Venezuela. Actualmente está adscrito como Investigador en seguridad informática de la Fundación Centro Nacional de Desarrollo e Investigación en Tecnologías Libres (CENDITEL) Nodo Mérida, y ha sido profesor desde el año 2005 de la cátedra de Matemáticas Discretas del Departamento de Computación de la ULA.

Antonio Araujo Brett es merideño, Ingeniero de Sistemas egresado de la Universidad de Los Andes en Mérida, Venezuela. Trabajó como instructor del programa de la Academia de Networking de Cisco Systems en la Fundación Escuela Latinoamericana de Redes (EsLaRed) en Mérida. Formó parte del equipo de instructores de la Academia de Software Libre de Fundacite Mérida. Se desempeñó en el área de Seguridad Informática y desarrollo de software con herramientas libres en Fundacite Mérida desde el 2004 hasta 2007 y en proyectos conjuntos con la Superintendencia de Servicios de Certificación Electrónica (SUSCERTE). Desde finales de 2007 labora como personal de Seguridad de la información en el área de Desarrollo en Tecnologías Libres de la Fundación Centro Nacional de Desarrollo e Investigación en Tecnologías Libres (CENDITEL) en Mérida, Venezuela. Ha participado como ponente en diversos eventos de seguridad de la información y certificación electrónica. Entre sus intereses principales se encuentran investigación y desarrollo en tecnologías de certificación electrónica, dispositivos de hardware criptográfico, seguridad de la información y hardware libre.