INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK

TECHNIKI INFORMACYJNE TEORIA I ZASTOSOWANIA

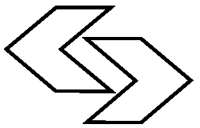
Wybrane problemy
Tom 1(13)

poprzednio

ANALIZA SYSTEMOWA W FINANSACH I ZARZĄDZANIU

Pod redakcją
Jerzego HOŁUBCA

Warszawa 2011



INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK

TECHNIKI INFORMACYJNE TEORIA I ZASTOSOWANIA

Wybrane problemy
Tom 1(13)

poprzednio

ANALIZA SYSTEMOWA W FINANSACH I ZARZĄDZANIU

Pod redakcją
Jerzego HOŁUBCA

Warszawa 2011

Wykaz opiniodawców artykułów zamieszczonych
w niniejszym tomie:

Dr hab. inż. Przemysław GRZEGORZEWSKI, prof. PAN

Prof. dr hab. inż. Jerzy HOŁUBIEC

Dr inż. Tatiana JAWORSKA

Dr hab. inż. Wiesław KRAJEWSKI, prof. PAN

Dr hab. inż. Maciej KRAWCZAK, prof. PAN

Dr hab. Michał MAJSTEREK

Dr hab. inż. Andrzej MYŚLIŃSKI, prof. PAN

Prof. dr hab. inż. Witold PEDRYCZ

Dr hab. inż. Ryszard SMARZEWSKI, prof. KUL

Prof. dr hab. inż. Andrzej STRASZAK

Dr Dominik ŚLĘZAK

Prof. dr hab. inż. Stanisław WALUKIEWICZ

ISBN 9788389475336

KONCEPCJA INTEGRACJI SYSTEMU ORACLE UNIVERSAL CONTENT MANAGEMENT Z PODPISEM ELEKTRONICZNYM

Jacek Olkowski

Studia Doktoranckie IBS PAN

The article describes the concept of integration the Content Management System (CMS) with a digital signature. Content management systems are often associated with WWW management tools. However, WWW management is one of many applications where the CMS could be used. Some of the numbers CMS applications include business solutions that can manage the whole process of document management and the work flow. One of the business implementations of the CMS systems is the Universal Content Management. The solution described in here extends the functionality of the standard CMS to electronic document signing. The electronic signing module was implemented based on presented concept. Only the PDF format is supported by the current version.

Key words*: CMS, UCM, Oracle, Content Management, Podpis elektroniczny, Digital Signature*

1. Wprowadzenie

Systemy zarządzania treścią - bardziej znane, jako CMS-y (ang. *Content Management System*) - najczęściej kojarzą nam się z aplikacjami internetowymi, które umożliwiają mało zaawansowanym użytkownikom tworzenie i administrowanie własnych stron WWW. Nawet popularna otwarta encyklopedia internetowa w polskojęzycznej wersji, pod hasłem CMS zawiera stwierdzenie, że jest to aplikacja internetowa lub zestaw aplikacji, pozwalający na łatwe utworzenie serwisu WWW oraz jego utrzymywanie. Jednak w rzeczywistości systemy klasy CMS nie ograniczają się tylko do zarządzania treścią stron internetowych. Jest to jedna z wielu możliwości ich wykorzystania. Ta sama encyklopedia w wersji anglojęzycznej pod tym samym hasłem zawiera już bardziej ogólną definicję, opisując system zarządzania treścią, jako zbiór procedur służących do zarządzania obiegiem dokumentów.

Większość dużych firm już jakiś czas temu zorientowała się, że aby sprawnie funkcjonować, potrzebuje narzędzia, które zautomatyzuje oraz zorganizuje proces zarządzania dokumentami w ich instytucjach. Dzięki takim rozwiązaniom nie ma już konieczności przenoszenia lub przesyłania dokumentów np. poprzez pocztę elektroniczną. Główną funkcją systemów zarządzania treścią jest tworzenie tzw. repozytorium danych, które odpowiada za udostępnianie i archiwizację wcześniejszych wersji wszystkich zgromadzonych w nich danych. Dzięki temu zespoły pracowników mogą wspólnie edytować ten sam plik z różnych końców świata widząc zmiany wprowadzone do niego przez ich kolegów od razu w momencie ich powstania [3], [4].

Przykładem takiego rozwiązania jest system UCM (Universal Content Management) firmy Oracle, którego rozszerzeniem zajmę się w tym artykule. Innym konkurencyjnym rozwiązaniem o podobnym zastosowaniu może być *Sharepoint* firmy Microsoft.

2. Workflow oraz inne funkcje UCM

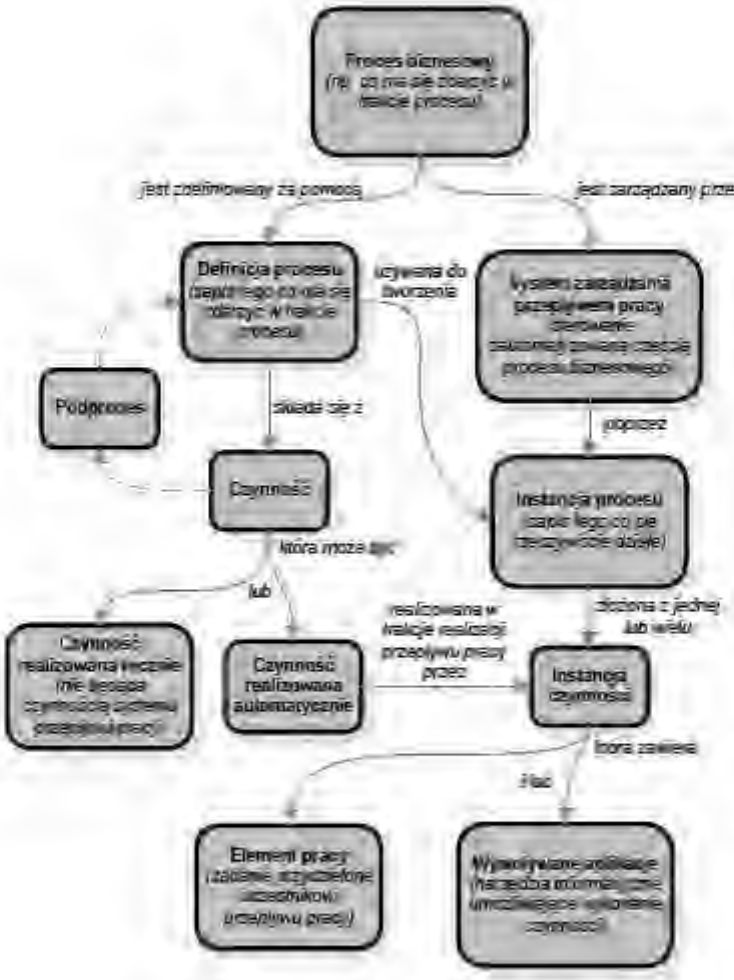
Jedną z istotniejszych cech, które wyróżniają biznesowe systemy zarządzania treścią, jest *Workflow* (ang. *work flow* – przepływ pracy). Na wstępie należałoby odpowiedzieć na pytanie, czym tak naprawdę jest ten *Workflow*? Międzynarodowa koalicja (*WfMC* – The Workflow Management Coalition) zajmująca się tą tematyką, pokusiła się o sformułowanie definicji stwierdzając, że *Workflow* jest to automatyzacja procesów biznesowych lub ich części, podczas których dokumenty są przekazywane między uczestnikami w celu podjęcia kolejnych określonych w procedurach działań [1].

Automatyzacja procesów biznesowych wymaga wcześniejszego ich zdefiniowania oraz opisania wg ustalonej semantyki. Dla bardziej skomplikowanych procesów wymagających udziału systemów zewnętrznych może to być np. BPEL (ang. *Business Process Execution Language* – język do definiowania procesów biznesowych dla usług sieciowych).

Sama identyfikacja procesów oraz ich głównych składowych nie jest łatwym zadaniem. Należy mieć na uwadze, że od jakości wykonania tego zadania zależy sukces wdrożenia całego systemu. Aby zrozumieć znaczenie podstawowych pojęć używanych w tej dziedzinie oraz zależności między nimi *WfMC* wprowadziła model (Rysunek 1.), który może być pomocny podczas opisywania procesów.

Jednak sam proces obiegu dokumentów to nie wszystko. Oracle UCM wspiera niemalże pełen cykl życia informacji (Rysunek 2.) stając się przez to

niezwykle cennym i pomocnym narzędziem w pracy każdego przedsiębiorstwa. Dzięki wersjonowaniu wszystkich przechowywanych dokumentów użytkownik może zawsze wrócić do dowolnego stanu sprzed edycji. Audyt systemu pozwala natomiast na śledzenie wszystkich czynności dokonywanych w dokumencie. Metadane i wyszukiwanie wpływają na łatwość dostępu do wszystkich przechowywanych w systemie danych.



Rysunek 1. Model relacji między pojęciami z obszaru *Workflow* [1]

Rysunek 2. Cykl życia informacji [2]

Można by uznać, że system wspiera wszystkie możliwe operacje na dokumentach w postaci elektronicznej. Jednak od momentu, kiedy w polskim prawie podpis elektroniczny został uznany za równoważny odręcznemu obsługiwany w systemie proces wydaje się być niepełnym.

3. Cel integracji

Duża część dokumentów na pewnym etapie swojego istnienia wymaga akceptacji. Najczęściej akceptacja jest równoważna z podpisaniem jakiegoś dokumentu. Przyjmijmy, że mamy do czynienia z dokumentem, który stanowi jakąś umowę. W pewnym momencie będziemy musieli ją wydrukować a następnie przesłać do podpisu lub uczynić to samemu.

Głównym celem, jaki przyświeca prezentowanej koncepcji jest rozszerzenie systemu UCM o możliwość sygnowania dokumentów podpisem elektronicznym.

3.1. Podpis elektroniczny

Definicja podpisu elektronicznego jest w zasadzie dosyć intuicyjna i mówi nam w zasadzie niewiele: "*dane w postaci elektronicznej, które służą do identyfikacji osoby składającej taki podpis*" [7]. Ciekawostką jest, że pojęcie podpisu elektronicznego jest używane zamiennie z podpisem cyfrowym co jest nie do końca poprawne. Według Polskiej Normy PN-I-02000 [10] podpis cyfrowy jest: "*przekształceniem kryptograficznym danych umożliwiającym odbiorcy danych sprawdzenie autentyczności i integralności danych oraz zapewniające nadawcy ochronę przed sfałszowaniem danych przez odbiorcę*". Tak, więc podpis cyfrowy jest pojęciem zawężonym w stosunku do podpisu elektronicznego. Jednakże podpis cyfrowy jest aktualnie najpopularniejsza formą podpisu elektronicznego.

Aby podpis elektroniczny właściwie realizował zamierzony cel, wyróżnia się trzy podstawowe funkcje jakie powinien spełniać:

- integralność – gwarantuje, iż dokument nie został zmodyfikowany po podpisaniu przez autora,
- niezaprzeczalność – oznacza, że autor dokumentu nie może wyprzeć się swojego podpisu,
- autentyczność – czyli gwarancja tego, kto jest autorem dokumentu.

Dodatkowo, podpis ten powinien spełniać takie same warunki co zwykły podpis, czyli:

- powinien być maksymalnie trudny lub wręcz niemożliwy do sfałszowania,
- powinien być weryfikowalny oraz trwale łączyć się z dokumentem.

Na mocy ustawy o podpisie elektronicznym z dnia 18 września 2001 r. [9], zostało wprowadzone pojęcie bezpiecznego podpisu elektronicznego, który ma taką samą moc prawną, jak tradycyjny podpis wykonany odręcznie. Podpis

ten, potocznie nazywany jest kwalifikowanym[1]. Jednakże, aby podpis mógł zostać tak określonym, musi spełniać dodatkowe kryteria [8], tj.:

- musi zostać dokonany przy użyciu bezpiecznego urządzenia, którym zazwyczaj jest karta kryptograficzna, czyli urządzenie, które zabezpiecza fizycznie i logiczne klucze prywatne właściciela,
- musi być weryfikowany za pomocą kwalifikowanego certyfikatu, wydanego w imieniu Narodowego Centrum Certyfikacji [9].

3.2. Stosowane algorytmy podpisu elektronicznego

Mechanizm działania podpisu elektronicznego jest realizowany przez metody kryptografii asymetrycznej. Podpis jest generowany na podstawie tajnych danych (tj. klucz prywatny), znanych jedynie autorowi podpisu, a jego weryfikacja odbywa się przy pomocy informacji ogólnodostępnych (tj. klucz publiczny). Wszystkie algorytmy podpisów są oparte na jednokierunkowych funkcjach skrótu[2] - tzn. takich funkcjach, które można w łatwy i szybki sposób realizować w jedną stronę, jednak w drugą bardzo trudno. Zadaniem tych funkcji jest stworzenie unikalnego dla każdego dokumentu ciągu znaków (tj. skrótu), który jest na tyle charakterystyczny, że jakakolwiek zmiana w treści dokumentu determinuje zmianę wartość jego skrótu. Zatem do utworzenia podpisu elektronicznego wykorzystywany jest jedynie skrót dokumentu, a nie jego całość. Dzieje się tak dlatego, że skomplikowane funkcje kryptograficzne wymagają bardzo dużych nakładów czasowych, które przy dużych dokumentach byłyby nie do zaakceptowania. Tak więc to jakość wspomnianych wcześniej funkcji skrótu gwarantuje bezpieczeństwo podpisu elektronicznego [5], [6].

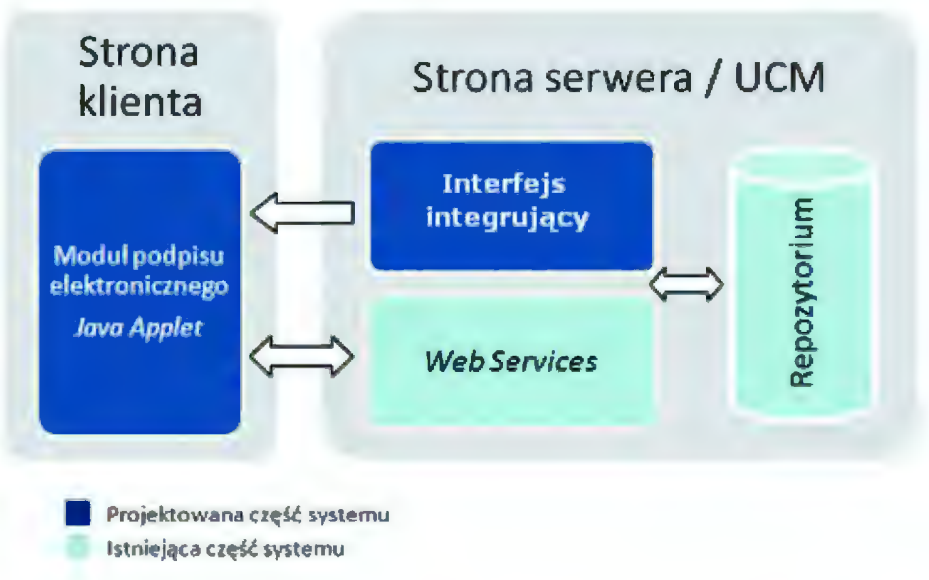
Dwa główne algorytmy podpisu elektronicznego to: RSA (od nazwisk twórców: Ronald Rivest, Adi Shamir, Leonard Adleman) i DSA (ang. *Digital Signature Algorithm*). Mimo iż w 1991r. rząd Stanów Zjednoczonych uznał DSA jako standard podpisu elektronicznego DSS (ang. *Digital Signature Standard*), to jednak najbardziej rozpowszechnionym jest algorytm RSA. Prawdopodobnie wynika to z polityki dużych firm, takich jak np. IBM, Apple czy Microsoft, które już wcześniej zainwestowały wiele pieniędzy w implementacje tego algorytmu.

[1] Podpis kwalifikowany – jest to nazewnictwo potoczne, ponieważ aktualne regulacje prawne w Polsce nie wprowadzają definicji podpisu kwalifikowanego

[2] Funkcja, która przyporządkowuje dowolnie dużej liczbie krótką, zwykle posiadającą stały rozmiar

4. Nowa koncepcja integracji

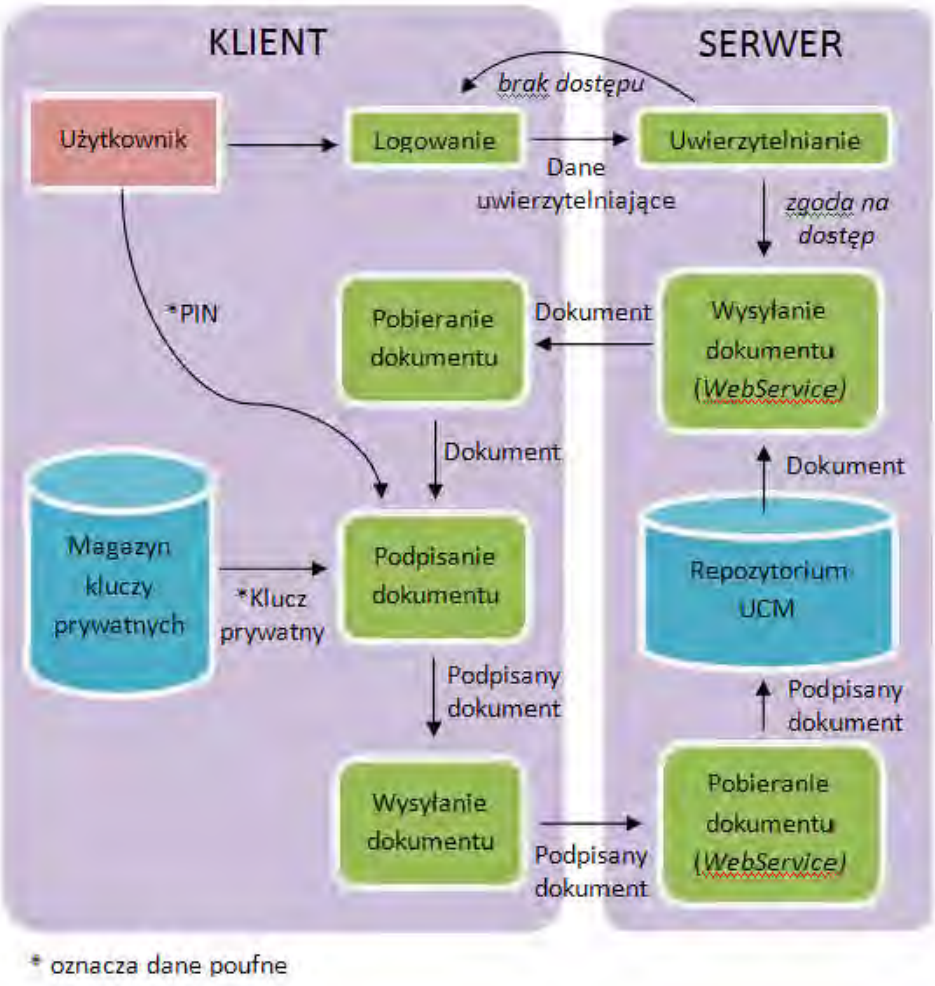
Całość idei rozwiązania sprowadza się do stworzenia takiej funkcjonalności, która będąc w pełni zintegrowana z systemem UCM umożliwi użytkownikom na elektroniczne podpisywanie dokumentów. Pod hasłem pełnej integracji mam na myśli takie rozwiązanie, które będzie dostępne z poziomu istniejącego interfejsu systemu. Jednak zanim przejdę do szczegółów koncepcji to warto wcześniej wspomnieć o wymogach bezpieczeństwa, jakie są istotne w procesie podpisu.



Rysunek 3. Koncepcja integracji

Zakładając, że mamy dostępne narzędzie do podpisu, potrzebujemy jeszcze dwóch elementów tj. dokumentu, który będziemy podpisywać oraz wspomnianego już wcześniej klucza prywatnego, którym podpiszemy dokument. Aby jednak podpis spełniał swoją rolę klucz prywatny powinien być odpowiednio zabezpieczony. Nie powinien być nigdy przesyłany po sieci do innych komputerów, ponieważ ktoś mógłby taki klucz przejąć. W celu zabezpieczenia się przed taką ewentualnością koncepcja zakłada, że podpis elektroniczny będzie generowany na maszynie użytkownika (w architekturze klient – serwer, strona klienta). Dzięki takiemu rozwiązaniu żadne newralgiczne z punktu widzenia bezpieczeństwa dane nie będą przesyłane po sieci.

W ramach ogólnej koncepcji integracji (Rysunek 3.) stworzyliśmy dwa nowe moduły po obu stronach systemu. Po stronie serwera jest to nowy komponent UCM, pełniący rolę interfejsu integrującego. Natomiast po stronie klienta został zrealizowany moduł generowania podpisu elektronicznego, zaimplementowany w technologii Java Applet. Technologia ta pozwoliła na osadzanie na stronie WWW aplikacji napisanych w języku Java.



Rysunek 4. Przepływ danych w systemie

Aplikacje te mimo swojego osadzenia na stronie, wykonywane są na komputerze użytkownika. Taki podział zapewnił bezpieczny przepływ danych niezbędnych do wygenerowania elektronicznego podpisu (Rysunek 4).

Podczas implementacji modułu do podpisu elektronicznego skupiliśmy się przede wszystkim na problemach natury integracyjnej tj., w jaki sposób komunikować się z UCM. Do generowania samego podpisu użyliśmy otwartej biblioteki *iText* [11], która umożliwia podpisywanie dokumentów typu PDF. Interfejs graficzny (Rysunek 5.) został zbudowany w oparciu o standardową bibliotekę Swing.

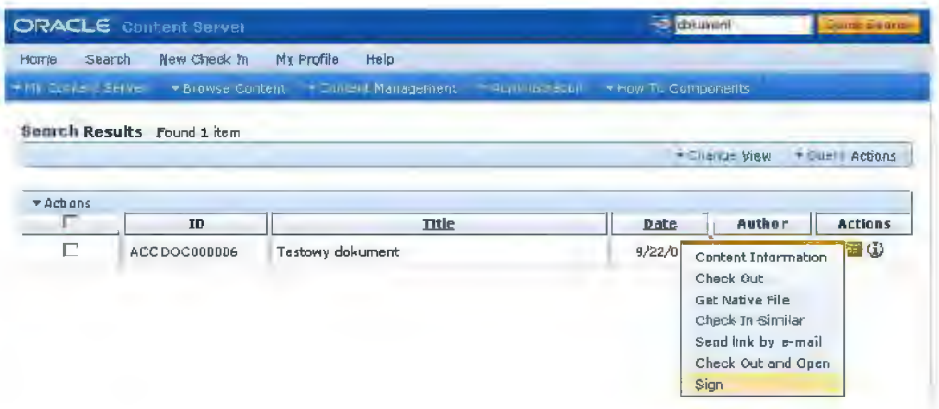
Rysunek 5. Moduł podpisu elektronicznego

4.1. Analiza otrzymanego rozwiązania

Analizując problem komunikacji należy zastanowić się, w jaki sposób implementowany moduł będzie otrzymywał dokumenty do podpisu oraz w jaki sposób będzie je wysyłał z powrotem do repozytorium UCM. Istotne jest również, aby zapewnić spójność rozwiązania z ogólną ideą działania systemów zarządzania treścią, która zakłada, że przed modyfikacją dokumentu, w tym przypadku przed podpisaniem, należy wcześniej oznaczyć dokument, jako edytowany (operacja *checkout*). Po podpisaniu dokumentu nowa wersja dokumentu również musi być odpowiednio zarejestrowana (operacja *checkin*). Wszystkie te założenia zostały zrealizowane dzięki wykorzystaniu wbudowanych w system UCM usług sieciowych (ang. *Web Services*), które udostępniają wszystkie potrzebne nam operacje tj. *checkin* oraz *checkout*.

Kolejna kwestia pozostała do rozważenia to, w jaki sposób udostępnić moduł podpisu w systemie UCM tak, aby był on dostępny z poziomu istniejącego już interfejsu.

Tutaj z rozwiązaniem przyszła możliwość tworzenia nowych komponentów w systemie UCM przewidziana przez producenta systemu. Dzięki niej możliwe było stworzenie nowej opcji „*Sign*" dostępnej w menu kontekstowym dokumentów przechowywanych w UCM (Rysunek 6.).



Rysunek 6. Kontekstowe menu z nową opcją w systemie UCM

Struktura tworzenia nowych komponentów jest ściśle określona i wyspecyfikowana. Przewiduje ona stworzenie plików konfiguracyjnych, które powiedzą systemowi jak komponent się nazywa oraz w jakim kontekście może

być użyty. Ponadto struktura nowego komponentu umożliwia załączenie dodatkowych zasobów, z których nowy komponent może korzystać. W naszym przypadku jest to plik JAR zawierający moduł do podpisu elektronicznego w formie apletu. Sam komponent jest już niczym innym jak stroną internetową w formacie HTML, na której zostanie osadzony wspomniany aplet.

5. Wnioski

Zaproponowane w artykule rozwiązanie zostało zaprojektowane w oparciu o koncepcję, której główną intencją jest maksymalizacja bezpieczeństwa informacji poufnych, oraz minimalizacja ryzyka związanego z wypływem tychże informacji. Na podstawie tej koncepcji został zaimplementowany moduł realizujący podpis elektroniczny dokumentów typu PDF, który może być w łatwy sposób zainstalowany w standardowym systemie UCM. Instalacja tego narzędzia odbywa się przy pomocy wbudowanego w system UCM narzędzia - *Component Wizzard,* które służy do dodawania nowych modułów.

Aktualnie skupiamy się nad dostosowaniu narzędzia do podpisywania innych typów informacji. Niektóre formaty danych takie jak np. pliki graficzne nie przewidują w swojej strukturze miejsca na podpis elektroniczny. Tak więc aby umożliwić podpisywanie takich informacji należy opracować odpowiedni kontener możliwy do podpisania, a jednocześnie gwarantujący integralność zawartych w nim danych.

Literatura

[1] WfMC (1999), *Workflow Management Coalition Terminology & Glossary,* s. 7-24. http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf

[2] Michał Szkopiński :Oracle Polska (2010), *Oracle Universal Content Management*. http://oracle-pl.blogspot.com/2010/07/oracle-universal-content-management.html

[3] Bob Boiko (2005), *Content Management Bible,* Wiley Publishing, cz. 1-2.

[4] Phil Suh, Dave Addey, David Thiemecke, James Ellis (2003), *Content Management Systems*. Apress, rozdz. 1.

[5] Brouce Schneier (2002), *Kryptografia dla praktyków: protokoły, algorytmy i programy źródłowe w języku C*, WNT, Warszawa, cz. 3.

[6] David Hook (2005), *Beginning Cryptography in Java*. Wrox Press, rozdz. 3-4.

[7] PWN (2008), *Encyklopedia PWN A-Z.* Wydawnictwo Naukowe PWN, s.. 786.

[8] Rozporządzenie Rady Ministrów z dnia 7 sierpnia 2002r. (Dz. U. nr 128, poz. 1094).

[9] Ustawa z dnia 18 września 2001 r. o podpisie elektronicznym (Dz. U. 2001 nr 130 poz. 1450).

[10] PKN (2002), *Technika informatyczna - Zabezpieczenia w systemach informatycznych - Terminologia PN-I-02000.*

[11] *iText - Free /Open Source PDF Library for Java and C#* .http://www.itextpdf.com/

ISBN 9788389475336

J. Hołubiec, red. – TECHNIKI INFORMACYJNE-TEORIA I ZASTOSOWANIA, tom 1 (13)