

Dariusz STACHAŃCZYK

POLITECHNIKA ŚLĄSKA, INSTYTUT ELEKTRONIKI

Implementacja systemu katalogowania komponentów wirtualnych umożliwiającego integrację rozproszonych zasobów projektowych

Dr inż. Dariusz STACHAŃCZYK

Ukończył studia na Wydziale Automatyki, Elektroniki i Informatyki Politechniki Śląskiej, pracę doktorską obronił w 2005 r. Obecnie adiunkt w Instytucie Elektroniki Politechniki Śląskiej. Jego zainteresowania naukowe obejmują projektowanie komponentów wirtualnych, wykorzystanie komponentów w projektach układów SoC oraz rozproszone środowiska projektowania.

e-mail: Dariusz.Stachanczyk@polsl.pl

Streszczenie

Możliwość wykorzystania gotowych komponentów wirtualnych (ang. Virtual Component - VC) ma istotny wpływ na przebieg i czas realizacji nowego układu SoC (ang. System on a Chip). Dlatego też, zwłaszcza, gdy w projekcie współpracuje kilka rozproszonych ośrodków projektowych, krytyczne dla projektantów systemu są informacje na temat wszystkich dostępnych w środowisku projektowym komponentów. Głównym celem prowadzonych badań było opracowanie prostego do wdrożenia i wykorzystania systemu katalogowania wirtualnych komponentów, który umożliwiłby integrację rozproszonych zasobów komponentów oraz proste i szybkie wyszukiwanie wymaganych komponentów w dostępnych zasobach. Zaproponowane rozwiązanie wykorzystuje bazę danych XML, w której przechowywane są dokumenty opisujące właściwości dostępnych komponentów. Dostęp do zgromadzonych w katalogu komponentów danych umożliwia projektantowi graficzny interfejs użytkownika.

Słowa kluczowe: komponenty wirtualne, komponenty IP, rozproszone katalogi komponentów wirtualnych.

Implementation of the virtual component cataloguing system enabling integration of distributed design resources

Abstract

The possibility to use several virtual components (VCs) is a crucial factor for a successful design of a new SoC device in the limited by market demands time. Availability of a rich virtual component library can increase the chances for a selection of the all required components thus an easy access to the information about all accessible in a design environment components is very important for the designers. This problem is especially important in case of distributed design environments comprising a number of geographically dispersed design SMEs (Small or Medium Enterprises) that collaborate in the SoC device design. Implementation of a collaborative platform that enables integration of distributed VC resources can significantly improve, a crucial for whole SoC design process, virtual component searching and selection stages. The Virtual Component Management System (VCMS) is a proposal of an inexpensive and easy to implement VC exchange platform that would enable integration of distributed components resources, simple and quick VC searching and facilitate VC selection. To enable virtual component data exchange, as well as to support VC selection, the XML-based description of a virtual component [6] characteristic was used within the VCMS environment. The virtual component XML descriptions are stored in a native XML database that constitutes the VC catalogue, which resources can be accessed via VCMS user interface application. The VCMS GUI allows designer for a VC searching and management, a VC description viewing and a VCMS environment configuration.

Keywords: virtual components, IP components, distributed catalogues of virtual components.

1. Wstęp

Dostępność dużej liczby uprzednio zaprojektowanych i przetestowanych komponentów wirtualnych (VC) może być krytycznym czynnikiem dla procesu projektowania złożonego układu System on a Chip (SoC) warunkującym zarówno czas trwania jak i ostateczny wynik projektu. Niestety, złożoność obecnych systemów elektronicznych powoduje, że większość, zwłaszcza mniejszych ośrodków projektowych nie posiada wystarczających zasobów projektowych, umożliwiających samodzielną realizację tego typu układów. Problem ten może zostać rozwiązany poprzez zaangażowanie w projekcie innych ośrodków projektowych i wykorzystanie ich zasobów komponentów wirtualnych lub skorzystanie z oferty szeregu firm projektujących komponenty wirtualne „na sprzedaż”. O ile z punktu widzenia projektanta potrzebującego komponentu realizującego wybraną funkcjonalność duża liczba dostępnych komponentów jest niewątpliwą zaletą, o tyle w przypadku konieczności sprawdzenia wielu różnych źródeł poszukiwanie niezbędnego komponentu może być operacją czasochłonną i niekoniecznie skuteczną. W sytuacji, gdy istnieje kilka źródeł komponentów wirtualnych projektant zmuszony jest zwykle powtórzyć proces wyszukiwania pożądanego komponentu w każdym z dostępnych zasobów, ponieważ nawet odnalezienie w jednym z dostępnych katalogów komponentu spełniającego zdefiniowane kryteria wyszukiwania, nie wyklucza sytuacji, że w którymkolwiek z pozostałych źródeł komponentów nie znajdzie się inny, podobny komponent lepiej spełniający wymagania specyfikacji realizowanego systemu. Dlatego też, efektywne wykorzystanie posiadanych zasobów komponentów wirtualnych wymaga, aby projektanci układu SoC mieli pełny dostęp do informacji na temat wszystkich dostępnych w środowisku projektowym komponentów, a także mogli szybko i sprawnie przeszukiwać dostępne zasoby pod kątem wymagań specyfikacji realizowanego systemu.

2. Wyszukiwanie komponentów wirtualnych w rozproszonych zasobach

Potencjalnym rozwiązaniem problemu wyszukiwania wirtualnych komponentów wydaje się być wykorzystanie jednej, wspólnej bazy zawierającej informacje o komponentach oferowanych przez różnych dostawców. Przykładem takiego rozwiązania mogą być katalogi komponentów wirtualnych opracowane i udostępnione w portalach internetowych organizacji Design&Reuse (D&R) [1] oraz Virtual Component Exchange (VCX) [2]. Główną wadą powyższych rozwiązań są mało efektywne mechanizmy wyszukiwania komponentów oraz ograniczony opis właściwości komponentu publikowany w postaci strony WWW. Dodatkowo, rozwiązanie takie wymaga, aby zasoby wszystkich ośrodków projektowych zostały skatalogowane w jednej wspólnej bazie danych, co w niektórych przypadkach może być niepożądane lub nawet niemożliwe. Propozycja rozwiązania problemu wyszukiwania wymaganego komponentu w wielu źródłach, które przy zastosowaniu tradycyjnych metod może być operacją wyjątkowo czasochłonną, przedstawiona została w opracowaniu [3]. Celem opisywanego projektu było opracowanie metod i narzędzi, które umożliwiłyby wysłanie uniwersalnego zapytania, zawierającego wymagania względem poszukiwanego komponentu, jednocześnie do kilku różnych katalogów komponentów. Zrealizowane środowisko wykorzystuje centralny rejestr, w którym rejestrowane są dostępne katalogi komponentów, dzięki czemu projektant może wysłać odpowiednie zapytanie do wszystkich, lub ewentualnie tylko kilku

wybranych spośród dostępnych w środowisku, katalogów komponentów wirtualnych. O ile rozwiązanie to rozwiązuje problem jednoczesnego wyszukiwania pożądanego VC w wielu, rozproszonych katalogach, posiada ono poważną wadę, jaką jest istotnie ograniczona funkcjonalność mechanizmów wyszukiwania komponentów. Wyszukiwanie komponentów ograniczone jest tylko do poszukiwania słowa kluczowego w obrębie kilku wybranych atrybutów charakteryzujących komponent, a w wyniku wyszukiwania projektant uzyskuje informacje tylko o podstawowych parametrach wirtualnego komponentu opisanych przez te atrybuty. Zdecydowanie bardziej rozbudowane mechanizmy wyszukiwania komponentów zaimplementowano w środowisku RMS (Reuse Management System) [4]. Środowisko RMS, które podstawowym zadaniem jest wsparcie procesu selekcji komponentów w projektach układów SoC, umożliwia również integrację i wykorzystanie rozproszonych zasoby komponentów wirtualnych. Katalog wirtualnych komponentów systemu RMS (RMS Database) może być rozproszony i integrować bazy komponentów RMS dostępne zarówno w sieci lokalnej jak i Internecie. Charakterystyka poszukiwanego komponentu definiowana jest poprzez określenie, wybranej na podstawie zaimplementowanej w RMS klasyfikacji, kategorii poszukiwanego komponentu, oraz dodatkowo, poprzez specyfikację atrybutów charakterystycznych dla wybranej klasy komponentów.

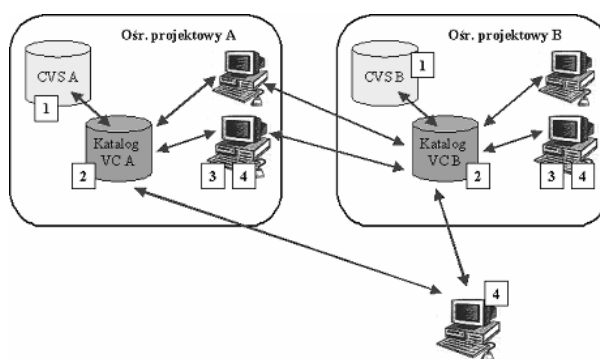
Poza technicznymi możliwościami przeszukiwania różnych, zwykle rozproszonych zasobów komponentów ważnym aspektem wyszukiwania komponentów jest problem oceny i porównania właściwości różnych komponentów i wyboru tego, który najlepiej spełnia wymagania specyfikacji realizowanego systemu. Rozwiązanie powyższego problemu wymagałoby zastosowania odpowiedniej, zunifikowanej metody opisu właściwości wirtualnych komponentów dla wszystkich dostępnych źródeł, co zdecydowanie ułatwiłoby analizę i porównanie parametrów komponentów pochodzących od różnych dostawców. Niestety, udostępniany zwykle opis zawiera w większości przypadków tylko podstawowe informacje dotyczące wybranych atrybutów komponentu, powodując że w celu zapoznania się ze specyfikacją komponentu konieczne jest pobranie i analiza dodatkowej dokumentacji.

Podsumowując - pomimo faktu istnienia szeregu rozwiązań i narzędzi dedykowanych celom katalogowania i wyszukiwania wirtualnych komponentów możliwości ich wykorzystania do integracji rozproszonych zasobów komponentów wirtualnych są w większości przypadków istotnie ograniczone. Ograniczenia te zostały uwzględnione w trakcie przygotowywania specyfikacji i realizacji środowiska integracji rozproszonych zasobów komponentów VCMS (ang. Virtual Component Management System) [5].

3. Środowisko integracji rozproszonych zasobów projektowych

Głównym celem w trakcie realizacji VCMS [5] było opracowanie taniego, prostego do wdrożenia i wykorzystania środowiska katalogowania i wymiany wirtualnych komponentów, które umożliwiałoby integrację rozproszonych zasobów komponentów wirtualnych oraz proste i szybkie wyszukiwanie komponentów na podstawie kryteriów obejmujących szereg atrybutów komponentu. W zrealizowanym środowisku wykorzystano bazującą na XML metodę opisu komponentów wirtualnych [6], co pozwoliło na implementację mechanizmów usprawniających wymianę danych na temat wirtualnych komponentów oraz ich wyszukiwanie w rozproszonych katalogach. Architektura oraz główne elementy środowiska integracji rozproszonych zasobów projektowych VCMS, obejmującego dwa przykładowe ośrodki projektowe A i B przedstawiona została schematycznie na Rys. 1. Podstawowym elementem VCMS jest katalog VC (element 2 na Rys. 1), którego zadaniem jest przechowywanie informacji na temat dostępnych w środowisku komponentów. Chociaż katalog komponentów

wykorzystywany jest tylko do przechowywania zdefiniowanych za pomocą XML opisów właściwości komponentów, dzięki zawartym w opisie danym na temat lokalizacji plików wirtualnego komponentu katalog umożliwia uzyskanie dostępu do właściwych plików projektowych, np. zapisanych w repozytorium CVS (element 1 na rys. 1). Najważniejszym z punktu widzenia projektanta elementem środowiska jest interfejs użytkownika (elementy 3 i 4 na rys. 1), który pozwala na konfigurację środowiska (np. rejestrację dostępnych katalogów), realizację operacji związanych z zarządzaniem (dodawanie, edycja, kasowanie danych VC), a także wyszukiwaniem komponentów w dostępnych katalogach.



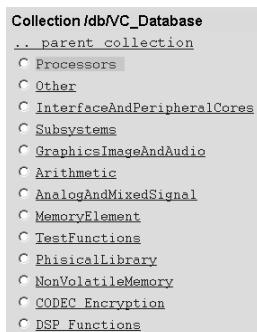
Rys. 1. Architektura i główne elementy środowiska VCMS
Fig. 1. Architecture and main elements of a VCMS environment

4. Katalog komponentów wirtualnych środowiska VCMS

Ze względu na fakt wykorzystania XML jako języka opisu komponentów wirtualnych, katalog komponentów zrealizowany został w oparciu o dedykowaną dla potrzeb przechowywania dokumentów XML bazę danych *eXist* [7]. Do zalet wynikających z zastosowania bazy dokumentów XML zaliczyć należy między innymi możliwość efektywnego przechowywania i przetwarzania całych dokumentów XML, które w łatwy i szybki sposób mogą być przeszukiwane i pobierane. Poza oferowaną funkcjonalnością ważną zaletą bazy *eXist* jest również to, że jest ona narzędziem całkowicie darmowym, udostępnionym na zasadach określonych przez licencję GNU. Publiczna dostępność bazy *eXist* umożliwia implementację własnego katalogu komponentów, a tym samym współdzielenie własnych zasobów w rozproszonym środowisku, nawet najmniejszym ośrodkom projektowym. Komunikacja pomiędzy bazą a interfejsem użytkownika bazuje na protokole zdalnego wywoływania procedur - XML-RPC, którego głównymi zaletami są wysoka elastyczność i mała złożoność pozwalające na prostą implementację mechanizmów obsługi bazy dokumentów XML w różnych językach programowania, między innymi w języku Java wykorzystanym do realizacji aplikacji użytkownika środowiska.

Przechowywane w bazie *eXist* dokumenty XML są zorganizowane w postaci hierarchicznej struktury tzw. kolekcji, opracowanej na podstawie klasyfikacji funkcjonalnej wirtualnych komponentów. Katalog komponentów wirtualnych VCMS został zrealizowany w postaci kolekcji o nazwie *VC_Database*, której struktura (tylko główne kolekcje), przedstawiona została na rys. 2. W porównaniu do rozwiązania wykorzystującego pojedynczą kolekcję do przechowywania plików wszystkich komponentów wirtualnych, wykorzystanie hierarchicznej struktury kolekcji pozwala na lepszą organizację danych wewnątrz katalogu oraz usprawnienie i skrócenie czasu operacji wyszukiwania komponentów poprzez ograniczenie obszaru wyszukiwania tylko do plików XML zapisanych w obrębie wybranej kolekcji (klasy

funkcjonalnej) komponentów. Możliwość poszukiwania wymaganego komponentu tylko w obrębie wybranej klasy funkcjonalnej pozwala na wyeliminowanie z listy otrzymanych wyników tych komponentów, które np. spełniałyby przypadkowo kryteria wyszukiwania pomimo przynależności do zupełnie odrębnej klasy funkcjonalnej.



Rys. 2. Kolekcja komponentów wirtualnych *VC_Database* – główne kategorie VC
Fig. 2. The *VC_database* collection in the eXist native XML database

Wyszukiwanie danych w bazie *eXist* realizowane jest za pomocą języka zapytań XQuery [8], umożliwiającego zdefiniowanie obszarów dokumentu, nazw elementów czy atrybutów, które powinny zostać przeanalizowane pod kątem zgodności z kryterium opisanym w treści zapytania. Projektant za pomocą graficznego interfejsu definiuje kryteria wyszukiwania w stosunku do różnych atrybutów VC na podstawie których tworzone jest odpowiednie zapytanie XQuery, wysyłane następnie do wszystkich zarejestrowanych w środowisku katalogów komponentów. W wyniku wyszukiwania projektant otrzymuje listę komponentów spełniających podane kryteria. Niezależnie od katalogu, z którego pochodzi komponent dla każdego wyniku możliwe jest wyświetlenie zuniifikowanego, szczegółowego opisu właściwości, co pozwala na dokładne porównanie poszczególnych komponentów.

5. Podsumowanie

Wsparcie operacji wyszukiwania komponentów w rozproszonych katalogach było głównym celem w trakcie implementacji systemu VCMS. W celu weryfikacji użyteczności VCMS zrealizowano eksperymentalne środowisko składające się z trzech rozproszonych katalogów komponentów i przeprowadzono szereg prób obejmujących wyszukiwanie przykładowych komponentów w dostępnych zasobach. Nakład operacji wymaganych do zakończenia procesu selekcji pojedynczego komponentu w środowisku VCMS porównany został z wynikiem uzyskanym w trakcie wyszukiwania analogicznego komponentu w środowisku wykorzystującym trzy różne metody katalogowania komponentów takie jak: portal WWW, repozytorium CVS czy lokalny system plików. Przedstawione w Tabeli 1 porównanie różnych metod dostępu i przeszukiwania zasobów wirtualnych komponentów wskazuje, że największe korzyści z wdrożenia VCMS mogą być uzyskane w przypadku środowisk, które nie wykorzystują żadnych zaawansowanych metod katalogowania komponentów. W takich przypadkach konieczność bezpośredniej komunikacji i wymiany informacji z dostawcą komponentu zwykle istotnie komplikuje proces wyszukiwania i selekcji komponentu. Zaletą VCMS jest również możliwość definiowania złożonych, wieloargumentowych kryteriów wyszukiwania, co pozwala zredukować liczbę otrzymywanych wyników w stosunku do tradycyjnych rozwiązań wykorzystujących proste kryteria wyszukiwania. Dodatkowo, rozbudowany opis XML komponentu, wykorzystany w VCMS pozwala na ograniczenie liczby przypadków, w których konieczne

jest uzyskanie i analiza dodatkowej dokumentacji, co wyraźnie wydłuża czas selekcji komponentów.

Tab. 1. Porównanie metod wyszukiwania i selekcji komponentów wirtualnych
Tab. 1. The comparison of the VC searching and selection methods

	Tradycyjne metody katalogowania VC			VCMS
	Portal WWW	CVS	System plików	Zintegrowane zasoby
Dostęp do danych VC	<ul style="list-style-type: none"> wymagana rejestracja dostęp poprzez przeglądarkę WWW 	<ul style="list-style-type: none"> wymagana aplikacja klienta CVS wymagane dane logowania 	<ul style="list-style-type: none"> wymagany kontakt z dostawcą VC 	<ul style="list-style-type: none"> wymagane dane logowania prosta, jednorazowa konfiguracja zuniifikowany dostęp do wszystkich zasobów
Mechanizmy wyszukiwania VC	proste, mało efektywne	brak	brak, wymagany kontakt z dostawcą VC	złożone, wieloargumentowe kryteria
Czas analizy wyników wyszukiwania	zależny od liczby wyników	zależny od liczby wyników	zależny od liczby wyników dostarczonych przez dostawcę VC	zależny od liczby wyników, ograniczonych przez kryteria wyszukiwania
Czas analizy dodatkowej dokumentacji	zależny od liczby wyników	zależny od liczby wyników	zależny od liczby wyników dostarczonych przez dostawcę VC	dzięki szczegółowemu opisowi VC ograniczony do minimum

6. Literatura

- [1] Design and Reuse, <http://www.design-reuse.com>
- [2] Virtual Component Exchange, <http://www.thevcx.com>
- [3] Haverinen A., Price G., Venkatramani K., Yunk M., A Standard Internet Ready VC Exchange System, Proceedings of Design, Automation and Test in Europe - DATE 2000 User Forum, Paryż, Francja, 2000
- [4] Seepold R., Reuse of IP and Virtual Components, Design, Automation and Test in Europe - DATE 1999, Monachium, Niemcy, 9-12.03.1999
- [5] Dariusz Stachańczyk, Platform for the integration of the distributed virtual component resources, Proceedings of the Workshop on Challenges in Collaborative Engineering, (CCE06), Praga, Czechy, 19-20.04.2006
- [6] Edward Hryniewicz, Dariusz Stachańczyk, Wykorzystanie języka XML do opisu właściwości wirtualnych komputerów, Pomiar Automatyka Kontrola - Reprogramowalne Układy Cyfrowe RUC'06, nr 7, 2006, str. 43-45
- [7] eXist – Open Source Native XML Database, <http://exist.sourceforge.net>
- [8] World Wide Web Consortium, XQuery - XML Query Language, <http://www.w3.org/TR/xquery>