ROZPROSZONE SYSTEMY KOMPUTEROWE LOKALIZACJI MOBILNYCH ELEMENTÓW

J. Długopolski, J. Szymaszek, A. Uszok, K. Zieliński

Katedra Informatyki AGH, Al. Mickiewicza 30, 30-059 Kraków

1. Wprowadzenie

Zagadnienia pozyskiwania informacji o położeniu użytkowników systemu i wykorzystaniu jej w działaniu systemów komputerowych są w centrum uwagi nie tylko takich kowencjonalnych aplikacji jak np. telefonia komórkowa GSM czy GPS, lecz także eksperymentalnych systemów jak:

- Active Badge System [] - jest to środowisko do lokalizacji ludzi i urządzeń przy użyciu krótko zasięgowej sieci transmisji sygnałów w podczerwieni. Zbierane infromacje lokalizacyje służą do personalizacji i zapewnienia mobilności środowiska pracy,

- Piconet [] - sieć radiowa krótko zasięgowa pozwalająca na stworzenie inteligentego środowiska wyposażonego w zdolność lokalizcji użytkowników, identfikacji ich upranień oraz informacji o ich stanie z jednej stony a z drugiej przekazywania użytkownikowi informacji o otoczeniu.

- ORL Ultrasonic Location System [] - system do lokalizacji położenia i śledzenia ruchu w przestrzeni trójwymiarowej z dokładnością do kilku centymetrów. Obszar zastosowań tego systemu obejmuje między innymi aktywne środowisko biurowe (*Active Office*).

Podane przykłady systemów należą do szybko rozwiającej się systemów komputerowych z wbudowanymi mechanizmami lokalizacji i mobilności (*context and location-aware computers*).

Niezależnie od obszaru zastosowań techniki i doładności realizacji lokalizacji systemy te posiadają wiele cech wspólnych pozwalających na budowę ram programowych. Ramy reprezentują podstawą funkcjonalność, którą można uszczegó'lowić w celu uzyskania finalnej aplikcji.

Celem niniejszego artykułu jest przedstawienie wybranych systemów lokalizacji komputerowej, obszarów ich zastosowań oraz ogólnej architektury oprogramowania skonstruowanej w Instytucie Informatyki AGH dla tej klasy systemów. Struktura artykułu jest następująca. W punkcie 2 omówiono przykładowe systemy lokalizacji ze szczególnym uwzględnieniem systemu Active Badge [] Punkt 3 zawiera opis ogólny architektury oprogramowania rozważanej klasy systemów.

Przykład takiej architektury, której implementację zrealizowano w Instytucie Informatyki AGH przedstawia punkt 4. Wybrane szczegółowe zagadnienia implementacyjne są przedmiotem punktu 5. Całość referatu kończą wnioski.

2. Przegląd wybranych systemów lokalizacji

System Active Badge System (ABS)jest produktem opartym na projekcie badawczym z Olivetti Research Laboratory w Cambridge w Anglii. Głównym celem projektu było stworzenie interaktywnego systemu do lokalizacji i monitorowania osób. System jest połączeniem środowiska sprzętowego i programowego. Komponentami sprzętowymi są odbiorniki i nadajniki w formie sensorów i osobistych identyfikatorów, które komunikują się ze sobą za pomocą sygnalów promieniowania podczerwonego oraz komputer zarządzający całym systemem. Sensory są statycznie rozmieszczone w róznych miejscach budynku, gdzie dokonują detekcji obecności poszczeg lnych identyfikator w. Komponenty programowe są w stanie dostarczać informacje dotyczące miejsc przebywania osób na rózne sposoby.

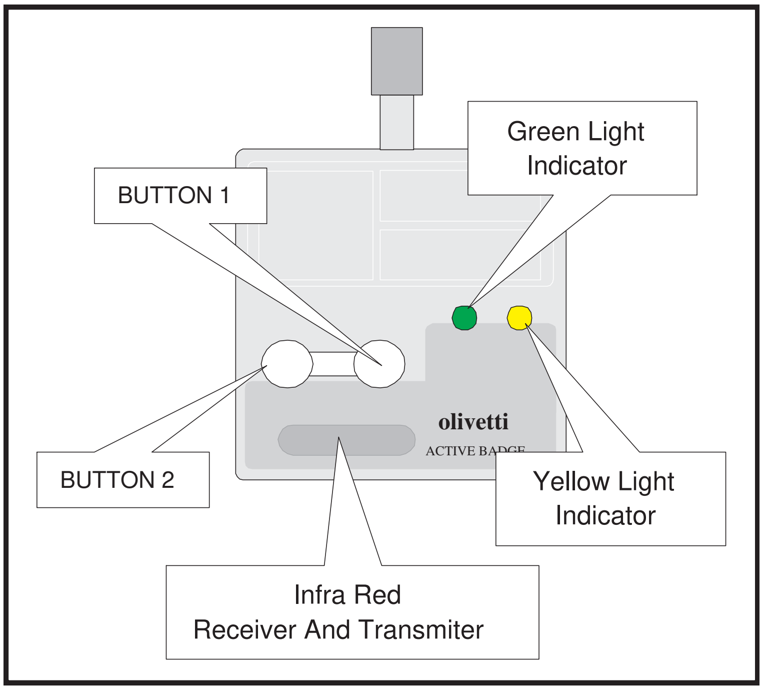
Osobiste identyfikatory jak sama nazwa wskazuje są urządzeniami identyfikującymi osoby je noszące. Każdy identyfikator składa się m.in. z nadajnika promieniowania podczerwonego, wskaźników informujacych o stanie urządzenia oraz dwóch przycisków (patrz rysunek 1).

Głównym zadaniem identyfikatora jest emitowanie sygnałów promieniowania podczerwonego, zawierających informacje o numerze identyfikacyjnym i inne dane. Generalnie identyfikator generuje dwa rodzaje sygnałów: sygnały pasywne regularnie co dziesięć sekund oraz sygnały aktywne w momencie wciśnięcia przez użytkownika jednego z dwóch przycisków. Sygnały te zawierają w sobie informacje mówiącą, który z przycisków został wciśnięty.

Sensory są urządzeniami, których głównym zadaniem jest odbiór sygnałów promieniowania podczerwonego emitowanego przez identyfikatory i dostarczenie zawartych w nich informacji do centralnego komputera zarządzającego systemem przy pomocy łącza szeregowego RS422. Na rysunku 2 został przedstawiony zewnętrzny wygląd takiego sensora.

Składa się on m.in. z odbiornika i nadajnika sygnałów promieniowania podczerwonego, dwóch wskaźników LED oraz przycisku służącego do restartowania urządzenia. Sensory są ustawiane w różnych miejscach, a informacja o ich położeniu znajduje się w specjalnej bazie danych. Dzięki temu możliwa jest lokalizacja dowolnej osoby noszącej identyfikator ABS.

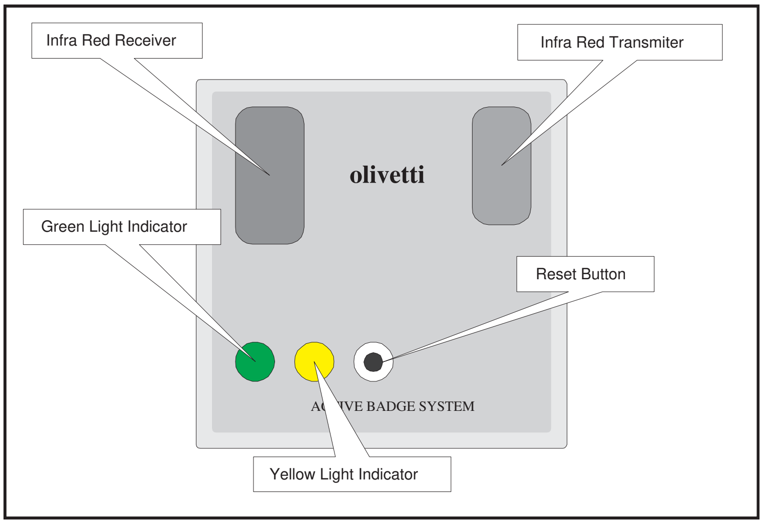
Ultrasoniczny system lokalizacji. System GPS umożliwia globalną lokalizację z dokładnością do kilku metrów w terenie otwartym. Jednakże staje się



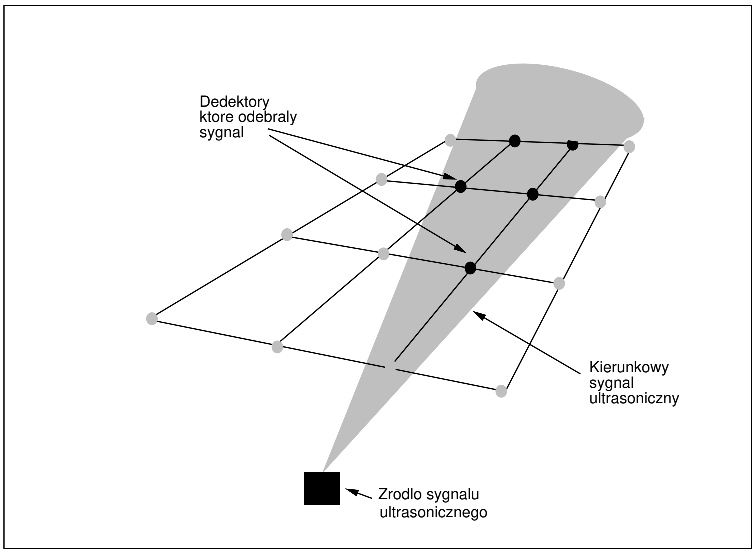
Rysunek 1: Wyglad zewnętrzny Aktywnego Identyfikatora A

on znacznie mniej efektywny wewnętrz budynków. Odbicia sygnału radiowego powodują bowiem, ze system przestaje być precyzyjny. W celu lokalizacji ludzi i urządzeń w pomieszczeniach zamkniętych z doładnością do kilku centymetrów opracowano w ORL system ultrasoniczny. Zasada działania tego systemu została zilustrowana na rysunku 3.

W pomieszczeniu na suficie rozmieszcza się siatkę dedektorów sygnału ultrasonicznego. Dedektory te połączone są za pomocą interfejsu szeregowego pomiędzy sobą oraz z komputerem sterującym. Miniaturowe nadajniki kierunkowego sygnału ultrasonicznego rozmieszcza się na przedmiotach, które mają być lokalizowane. Posiadają one formę plakietek przy czym każda plakietka posiada kilka takich nadajników, co pozwala na dołkadnieją jej lokalizacje w przestrzeni. Z okresem 200ms komputer sterujący wysyła sygnał *reset* do sieci dedektorów jednocześnie wysyłając drogą radiową pakiet zawierający adres plakietki, która ma wytransmitować sygnał ultrasoniczny. Pierszwy z dedektorów, który wykryje sygnał zgłasza ten fakt to komputera sterującego. Podstawą obliczeń lokalizacyjnych jest czas po którym następuje dedekcja sygnału i położenie dedektora, który pierwszy wykrył sygnał. Uwzględniając opoźnienia występujące w systenie oraz znając szybkości rozchodzenia się sygnału ultrasonicznego na podstawie



Rysunek 2: Wyglad zewnętrzny Sensora ABS



Rysunek 3: Zasada działania systemu ultrasonicznego

prostych obliczeń można ustalić położenie plakietki w przestrzeni. Praktyczna realizacja systemu potwierdziła jego dużą dokładość.

Przykłady zastosowań systemu obejmują:

- Wizualizacje ruchów przedmiotów w przestrzeni trójwymiarowej. Kojarząc kilka nadajników ultrasonicznych z urządzeniem można sędzić jego położenie. Wykorzystuje się w tym celu trójwymiarową wizualizacje komputerowaą, np. w VRML.

- Inteligentne środowisko. Komputer znając położenie przedmiotów w pomieszczeniu oraz aktualną pozycję użytkownika może informować go, np. glosowo o przedmiotach przed którymi aktualnie się znajduje. Stwarza to możliwość np. zbudowania inteligentego środowiska dla ludzi niewidomych.

- Sytem ultrasoniczny może być także używany w powiązniu z systemem Active Badge w celu dokładniejszej lokalizacji, usprawniając działanie takich aplikacji jak np. *Teleporting System.*

Piconet Projekt Piconet dotyczy krótko zasięgowej sieci radiowej, umożliwiającej komunikacje na odległość zaledwie kilku metrów (5m) z niezbyt dużą szybkością transmisji (około 9600b/s). Dzięki temu komponenty sieci są bardzo proste tanie i mają niski pobór mocy, co sprawia, że mogą być one instalowne praktyczne wszędzie lub moszone w formie plakietek.

Piconet należy do grupy sieci radiowych typu *ad hoc*, objętych standardem IEEE 802.11, jednakże w jej konstrukcji wprowadzono szereg uproszczeń związanych z dążeniem do obniżenia wielkości i ceny komponentów systemu.

Uniwersalność komunikacji w sieci Piconet przejawia się w schemacie adresacji, który obejmuje dobrze znane i czasowe adresy grupowe. Pierwszy rodzaj adresów wykorzystywany jest dla predefiniowanych usług, które są realizawane przez każdy wezeł sieci Piconet. Przykładem tego typu usługi jest ładawanie systemu.

Czasowe adresy grupowe są wyznaczane na drodze statystycznej poprzez losowanie adresu w przestrzeni 30bitowej i służą do komunikacji w grupie utworzonej ad hoc w celu realizacji okreslnej funkcji.

Węzel sieci Piconet jest wyposażony w jądro, którego funkcje realizują zarządzanie jego działaniem, oraz program ładujący. Program ładujący jak sama nazwa wskazuje ładuje program aplikacyjny, który wykonuje się następnie w środowisku jądra. Aplikacja stanowi kod, który umożliwia dostowanie się węzła sieci Piconet do realizacji kreślonego zadania.

Węzły sieci co 30s rozgłaszają swój identyfikator, poinformując w ten sposób inne węzły w najbliższym otoczeniu o ich istnieniu oraz ewentualne rozpoczęcie dialogu. W ten sposób każdy węzeł sieci Piconet przedstawia się, a fakt rozpoczęcia dialogu zależy od informacji jaką posiada on o węźle, którego identyfikator właśnie otrzymał. Węzły sieci Piconet mogą być na stałe związane z pewnymi urządzeniami lub mobile np. noszone przez użytkowników systemu.

Przedstawiona funckcjonalność sieci Piconet pozwala budowę szeregu bardzo ciekawych aplikacji jak np.:

1. Personalizacja środowiska pracy na drodze automatycznej konfiguracji urządzeń zgodnie z preferencjami użytkownika. Pojawienie się użytkownika zaopatrzonego w plakietkę z węzłem sieci Piconet powoduje jego rozpoznanie przez wszystkie urządzenia w danym otoczeniu, posiadające także węzły sieci Piconet. Mogą się one przedstawić węzłowi użytkownika, podając swoje identyfikatory a następnie uzyskać od niego polecanie odpowiedniej konfiguracji, czy też wykonania określonych sekwencji operacji,

2. Autoruzacja i autentykacja użytkownika systemu. Jest to bardzo ważna funkcja znajdująca zastosowanie w wielu systemach. Schemat jej realizacji nie odbiega zasadniczo od scenariusza przedstawionego powyżej.

3. Sieć Piconet może współpracować z innymi systemami lokalizacji. Przykładowo węzł sieci Piconet może być połączony z monitorem w celu wyświetlania informacji zgodnie z preferencjami użytkownika znajdującego sie w pobliżu. Informacje te mogą zwierać dane z systemu GPS lub Active Badge. "Inteligentne" zachowanie się monitora uzyskuje się dzięki automatycznemu rozpoznaniu użytkowika i środowiska oraz z konfigurowaniu atrybutów urządzeń.

3. Architektura oprogramowania systemów lokalizacji

Prawie wszystkie systemy lokalizacyjne posiadają podobną strukturę w sensie sposobu przetwarzania informacji. Model obliczeniowy zawyczaj formułowany jest w kategoriach lokalizowanych obiektów i lokalizacji. Lokalizowany obiekt reprezentuje osobę lub sprzęt, którego położenie ma być śledzone. Pojęcie lokalizacji jest bardziej złożone i może być formułowane na różnym poziomie abstrackji. W szczególnym przypadku mogą to być współrzędne geometryczne, geograficzne, pomieszczenie, np. typu pokój, korytarz itp. Dla każdego wyróżnionego poziomu abstrakcji należy zdefiniować odwzorowanie występujących pojęć w elementy fizyczne systemu, np. pokój jest o określona grupa dedektorów. Odzworowanie to izoluje wyższe warstwy oprogramowania od sprzętu.

Lokalizowane obiekty oraz lokalizacje posiadają szereg określających ich atrybutów przechowywanych zawyczaj w odpowiedniej bazie danych. Informacja ta zmieniana jest stosunkowo rzadko, przez administratora systemu, i dlatego określana jest mianem informacji statycznej. Zawiera ona wszystkie istotne informacje o lokalizowanym obiekcie i lokalizaji, nadając tym abstrakcyjnym obiektom określoną semantykę.

Informacja o lokalizacji zmienia się na bieżąco i musi być zawsze odniesiona jest to określonej przestrzeni lokalizacji i lokalizowanych obiektów. Lokacja stanowi wyróżnioną cześć przestrzeni, przy czym zbiór lokacji wyróżnionych na danym poziomie abstracji reprezentuje rozłączne obszary przestrzeni trójwymiarowej. Definicja loklizacji związana jest bezpośrednio z dokładnością lokalizacji obiektów np. z dokładnością do pokoju, piętra lub budynku. Większość systemów lokalizacji zakłada, że wyróżniona przestrzeń lokalizacji jest statyczna i niezmienia się w trakcie działania systemu.

Każdy z rozważanych rodzajów infromacji, tj. statyczna i dynamiczna, wnosi odmienny sposób jej przetwarzania. Informacja statyczna jest zawyczaj wczytywana przez aplikację w momencie jej uruchamiania. W trakcie pracy systemu zmiany informacji statycznej powinny być notyfikowe do poszczególnych aplikacji i zmieniane automatycznie. Wymaga to zastosowania schematu obserwator-obserwowany [] i odpowiedniego mechanizmu uakualniania kopii informacji w aplikacji.

Przetwarzanie informacji dynamicznej dostarczanej na bieżaco przez system lokalizacji można podzielić na następujące fazy:

Akwizycja - polega a odbiorze wiadomości generowanych przez system lokalizacji o aktualnym położeniu obiektu. Informacja ta zwykle jest formułowana w kategoriach fizycznych kompnentów systemu, np. dedektor x odebrał sygnał od plakietki y.

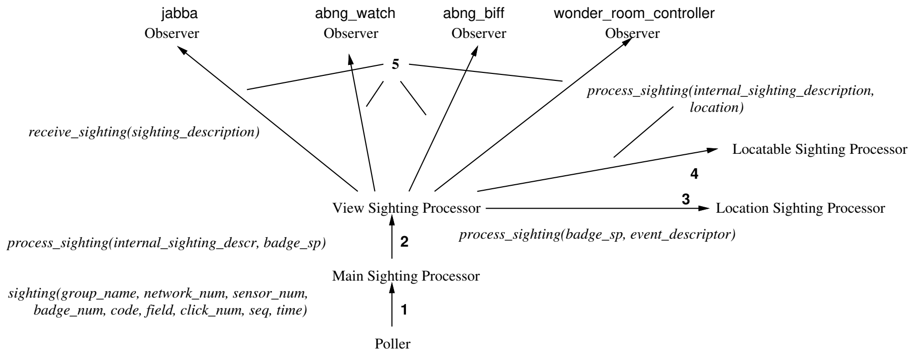
Przetwarzanie - przetwrzanie wiadomości lokalizacyjnych polega na zakwalifikowaniu ich do odpowiednich lokalizacji, odfitrowaniu powtarzających się informacji, oraz dokonaniu zmian przypisania obiektów do lokalizacji zgodnie z aktualnym stanem. Dla każdego typu - poziomu abstracji - lokalizacji proces przetwarzania może być realizowany równolegle przez różne logiczne lub fizyczne procesory.

Prezentacja - związana jest przekazaniem i prezentacją informacji o lokalizacji obiektów do konkretnych aplikacji. Bardzo często zadaniem aplikacji jest wizualizacja przestrzeni lokalizacji w postaci graficznej, np. mapy planarnej bądź trójwymiarowej przestrzeni wygenerowanej w VRML oraz zaznaczenie w tej przestrzeniu aktualnego położenia obiektów, badź ich trajektorii.

Przestawiony ogólny model funkcjonalny systemu lokalizacji zostanie dalej uszczegółowiony dla przypadku systemu Active Badge.

4. Budowa oprogramowania systemu ABng

Proces przetwarzania zdarzeń w systemie ABng opiera się na koncepcji *View*. Każde *View* składa się z dwóch zbiorów elementów typu: **lokacji** i **obiektów lokalizowalnych** Dane *View* może zawierać lokacje tylko jednego typu, np. tylko pokoje. W systemie może istnieć wielu różnych obiektów typu *View*, zawierających różne zbiory lokacji i obiektów lokalizowalnych. Lokacje i obiekty lokalizowalne danego *View* powiązane są ze sobą poprzez relację *sighting*. Taka relacja zachodzi pomiędzy tymi obiektami jeśli dany obiekt lokalizowalny był ostatnio widziany w danej lokacji. Pojedyńcza lokacja może być oczywiście w relacji *sighting* z wieloma obiektami lokalizowalnymi. Jądro systemu ABng składajcące się zespołu współdziałjących ze sobą procesorów sighting na różnych poziomach abstrakcji odpowiedzialne jest za przyjmowanie, przedtwarzanie i odświerzanie informacji o lokalizacji (Rys. 4).



Rysunek 4: Ścieżka przetwarzania sightings

Na dole hierarchii procesorów znajduje się *Main Sighting Processor* (MSP), który otrzymuje komunikaty od wszystkich badge'y aktywnych w systemie. Powyżej niego znajdują się moduły *View Sighting Processor* (VSP) związane z poszczególnymi *View*. Ostatni krok przetwarzania to *Location sighting Processors* (LPS) zwiazane z lokacjami danego *View* i *Locatable Sighting Processors* (LEPS) związane z obiektami lokalizowalnymi. Kolejne kroki przetwarzania informacji wysłanej przez badge'a to:

1. MSP przkazuje informację od danego badge'a, uzyskaną od *Poller*'a, do wszystkich VSP, których *View* zawiera obiekt lokalizowalny i lokację związane z badge'em i sensor'em, których identyfikatory są zawarte w odebranej informacji.

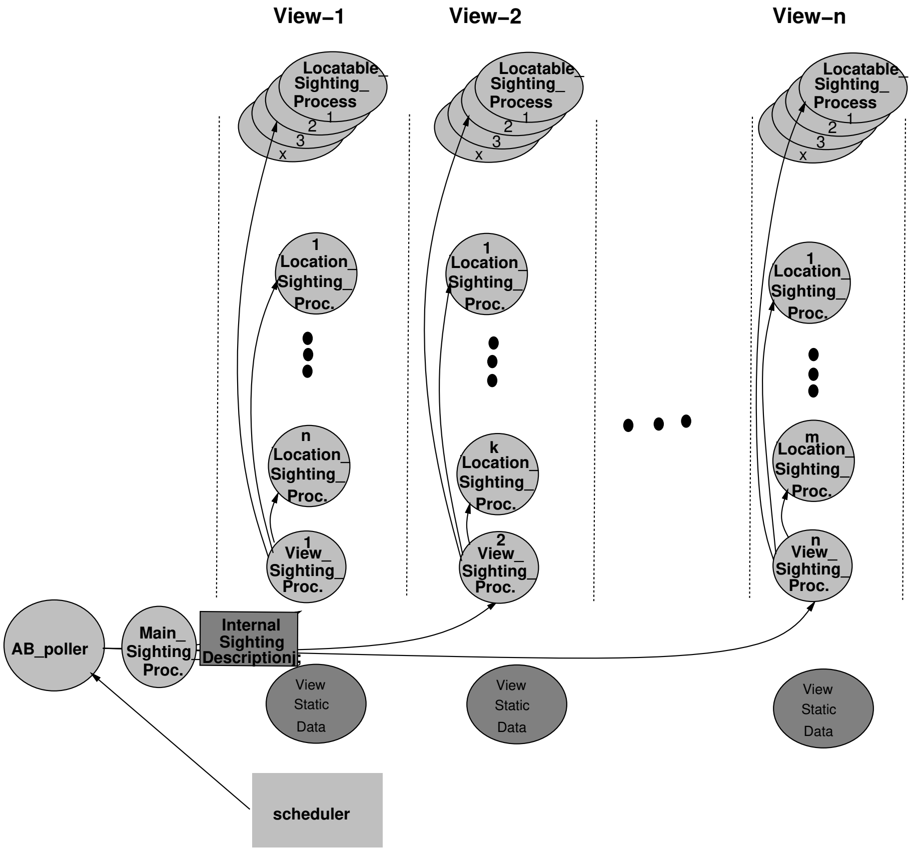
2. Każdy VSP znajduje swóje obiekt lokacji i obiekt lokalizowalny związane z odebraną od MPS wiadomością i przekazuje ją odpowiewdnim LSP (jendemu lub dwóm w zależności od tego czy obiekt lokalizowalny zmienił sowją lokację) i LESP do przetwożenia.

3. Uaktywnione LSP związane z poprzednią lokacją danego obiektu lokali-

zowalnego usówa swoją relację sighting, natomiast LSP związane z nową lokacją tworzy taką relację z obiektem lokalizowanym.

4. Uaktywnione LESP odświerzają dotyczącą go relację *sighting*.

5. Po zakończeniu kroków **4** i **5** VSP rozpoczyna notyfikację wszystkich obserwatorów zarejstrowanych w nim w lokacjach i obiekcie lokalizowalnym. których dotyczyło przetwarzane *sightinh*.

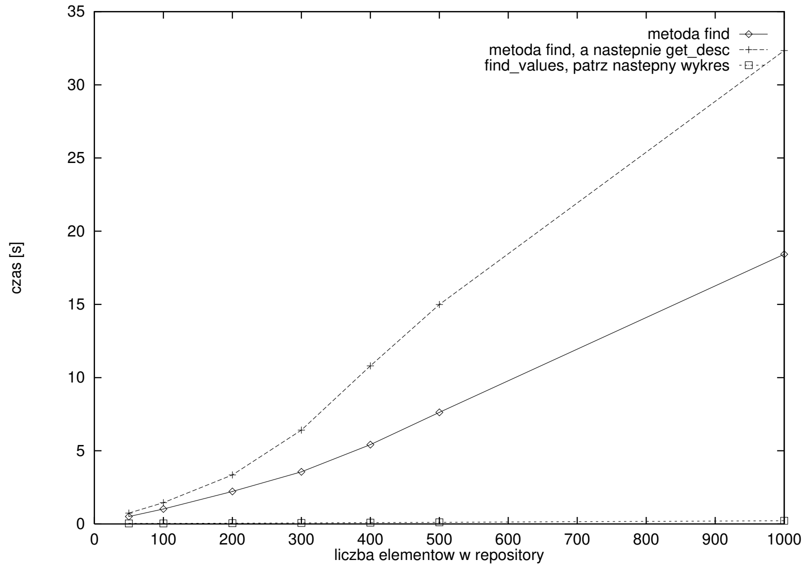


Rysunek 5: Wewnętrzna architektura jądra systemu ABng

5. Wybrane aspekty implementacyjne systemu

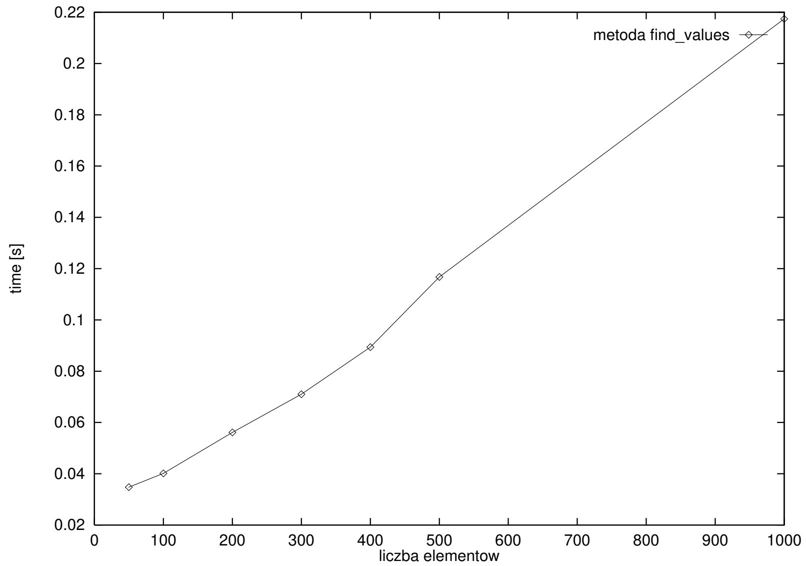
System ABng zaimplementowany zgodnie z przedstawionymi tutaj założeniami był przedmiotem testów mających na celu ocenę jego efektywności i skalowalności. System został zaimplementowany w środowisku Orbix 2.2MT, podczas gdy programy kilentów używane w testowaniu zostały napisane w języku Java w środowisku OrbixWeb. Zaprezentowane w tej sekcji rezultaty testów zostały uzyskane w środowisku składającym się z 15 stacji roboczych Sun Ultra i serwera Sun Enterprise 3000. Komputery połączone były ze sobą przy pomocy dwóch switch'y ethernet 10Mb. Komponenty systemu ABng działały na serwerze. Programy kielentów zaś na osobnych stacjach roboczych, tak więc wszystkie wywołania CORBA przekazywane były przez sieć. Wszystkie testy przeprowadzono 100 razy, a następnie obliczono średnią ich wyników.

Kosztów różnych sposobów dostępu do elementów repository W systemie zrealizowano dwa różne sposoby dostępu do elementów repository. Jeden z nich (metoda *find*) umożliwia pobieranie referencji elementów, a następnie wykonywanie na tych referencjach dowolnych operacji, np. pobrania opisu danego elementu. Drugi sposób (metoda *find_value*) pozwala na pobranie od razu opisu elementów bez uzyskiwania ich referencji. Czasy wykonania tych metod na jednym z repositori systemu przy zmieniającej sie liczbie elementów przedstawiono na Rys.6 i 7.



Rysunek 6: Porównanie efektywności różnych metod dostępu do elementów repository

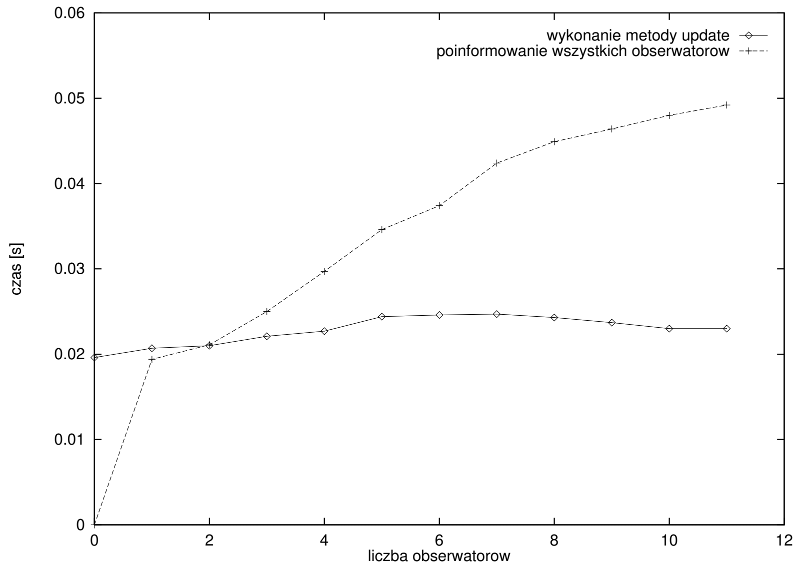
Uzyskane rezultaty pokazują, że dostę do obiektów repository poprzez ich referecncje jest o kilka rzędów bardziej kosztowny niż dostęp poprzez wartości. Dlatego dostęp poprzez referencje, pomimo swoich wielu ważnych zalet, powinnie być użuwany ostrożnie i cz[esto w połączeniu z dostępem przez wartość.



Rysunek 7: Porównanie efektywności różnych metod dostępu do elementów repository, powiększenie

Ocena czas notyfikacji i skalowalności systemu Metody używane do informacji obserwatorów o zaistniałych zmianach w stanie repositorii, czyli w szczególności w zmianie lokalizacji jakiegoś elementu, są asynchronicznymi operacjami CORBA typu *oneway*. Czas jaki potrzebny jest do poinformowania wszystkich obserwatorów o zaistniałej zmianie ma bardzo istotny wpływ na efektywne działanie systemu i zależy on oczywiście od ilości zarejestrowanych obserwatorów. Na wykresie Rys. 8 zaprezentowano tę zależnosc wyznaczoną eksperymentalnie będącą wynikiem przeprowadzonych na systemie ABng testów.

Uzyskane rezultaty pokazują, że czas wykonania metody *update* na elmencie repository prawie nie zależy od ilości zarejestrowanych obserwatorów. Jednakże czas poinformawania ich wszystkich o zaistaniałej zmianie zwieksza się o ok. 3 [ms] z każdym dodanym obserwatorem. Należy zauważyć tutaj, że proces informowania obserwatorów jest wyzwalany przez metodę *update*, ale później jest wykonywany w stosunku do niej w trybie asynchronicznym. Uzyskany wynik oznacz, że w jednej sekundzie można wykonać jedynie ok. 330 kolejnych operacji notyfikujących (oczywiście ta liczba zależy od użytego hardware'u). Dla użytego



Rysunek 8: Wplyw wzrastającej liczby obserwatorów na czas notyfikacji

sprzetu można więc napisać nierówność:

`Obs * Evn <= 330`

gdzie (`Obs`) jest liczbą obserwatorów, a (`Evn`) średnią liczbą zdarzeń na sekundę.

W przypadku systemy ABng zdarzeniami, kóre tworzą intensyny strumień zdarzeń są zmiany lokalizacji. Każdy badge generuje co 10 [s] nowy *sighting* informujący o jego aktualnym położeniu. Jednakże statystycznie mniej niż 4 % tych zdarzeń niesie istotną informacje: zmianę lokacji lub naciśnięcie przez użytkownika, któregoś z klawiszy badge'a. Tylko takie zdarzenia muszą być raportowane obserwatorą systemu. Stosując wyprowadzoną formułę do tych danych można oczekiwać, że system ABng zaimplementowany w architekturze CORBA będzie się skalował do około 500 badge'y i ponad 100 działających programów obserwujących zmiany w lokalizacji. Dodatkowo, koncepcja *View* zastosowana w systemie ABng umożliwia istotne ograniczenie ilości notyfikacji poprzez filtrację.

Aby uzyskać skalowalność systemu ponad te liczby badge'y i program'ów konieczne jest zastosowanie:

- protokołu typu *multicast*'owego,
- koncepcji drzewa rozgłoszeniowego,
- lub obu tych elementów.

W przypadku stosowania protokołu *multicast*'owego, na przykład poprzez użycie implementacji Event Service firmy Iona, liczba obserwatorów `Obs` w formule zostaje zredukowana do 1 i liczba notifikowanych zdarzeń może osiągnąć 330 na suekundę.

6. Wnioski