

Eligibilitate

Titlul solutiei	VM		
Nume Echipa	ImaGenesis		
Membrii echipa	Nume	Adresa e-mail	IC Username
	Pais Vasile	pvf2005@gmail.com	V Pais
	Pais Lucia	lucia.pais@gmail.com	Lucia Pais
	Ureche Vlad	vlad.ureche@gmail.com	Vlad Ureche
Mentor	Titlu	Nume	E-mail
	Prof. dr. ing.	Nicolae Tapus	ntapus@cs.pub.ro
University	Politehnica University of Bucharest, Faculty of Automatic Control and Computer Science		

Abstract

Avand in vedere dezvoltarea continua a aparaturii electronice, aparitia de noi gadget-uri si noi componente, este necesara posibilitatea simularii acestora pentru a intelege modalitatea lor de functionare, pentru a testa software scris pentru acestea sau pentru a crea si studia sisteme mai complexe obtinute prin imbinarea mai multor astfel de module.

Pentru a rezolva aceasta problema, VM vine atat in ajutorul studentilor cat si al producatorilor de componente, simuland echipamente si oferind diverse facilitati ca de exemplu *partajarea masinii virtuale* in module componente (ceea ce permite folosirea diverselor tehnici de colaborare in designul unui proiect), *remote participation* (acces – controlul echipamentelor de la distanta prin intermediul unui web service, data acces – preluarea datelor de la distanta prin web service si comunicatie cu masina virtuala prin intermediul unui server de VNC), *autentificare* etc.

Inovatia proiectului consta in faptul ca simulatorul de masini virtuale este gazduit pe un web server si in acelasi timp poate fi administrat prin web, prin intermediul unui web service. Aplicatia permite accesul mai multor persoane la acelasi device iar pentru a vizualiza rezultatele simularii, starea interna a sistemului simulat sau pentru a interactiona cu masina virtuala se foloseste protocolul standard VNC. Din punct de vedere al programului, VM se distinge prin arhitectura modulara, fiecare componenta din lumea reala reprezentand un modul in cadrul aplicatiei fapt ce confera o mare flexibilitatea si posibilitatea colaborarii intre mai multe persoane pentru crearea unui sistem.

Scenarii de utilizare

Aplicatia este menita pentru a fi folosita in special in scopuri educationale, avand in vedere existenta multor impedimente in folosirea dispozitivelor reale. De exemplu, in Romania multe dintre componentele electronice nu sunt disponibile sau sunt foarte scumpe pentru a si le putea permite un student (de mentionat ar fi AT32AP7000 – ATMEL 32-bit RISC microcontroller – care conform Froogle costa 53\$ dar in Romania depaseste 100\$). De asemenea in cazul procurarii de componente electronice de obicei este inclusa o anumita cantitate minima, fapt ce sporeste costurile. Si mai mult, unele componente sunt disponibile doar online, deci comanda si timpul de livrare cresc timpul total necesar realizarii unui proiect hardware.

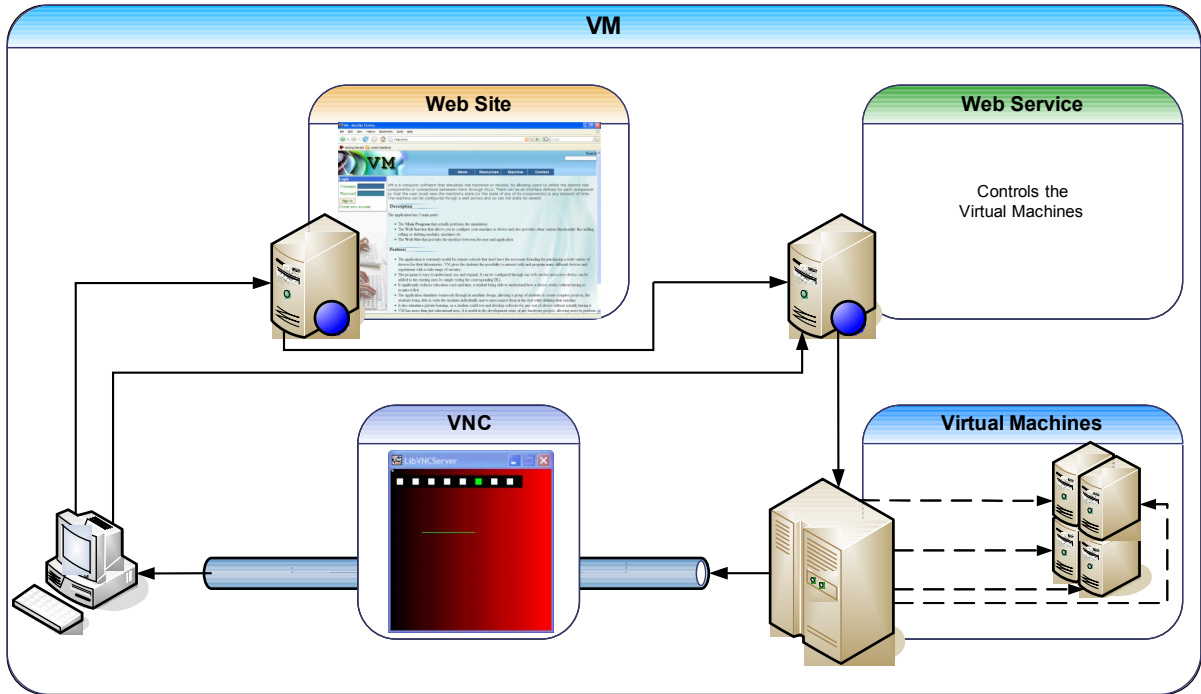
VM isi propune sa rezolve aceste probleme prin intermediul simularii. Astfel se micsoareaza considerabil costurile si timpul pierdut pentru asteptarea livrarii unui component. VM ajuta studentii/grupurile de studenti ce dezvolt proiecte hardware deoarece orice componenta poate fi simulata cu ajutorul acestuia si poate comunica apoi cu componente reale existente in sistemul respectiv.

In cadrul unei facultati, daca se doreste sa se ofere tuturor studentilor acces la un anumit device, el poate fi simulat cu ajutorul VM-ului, fiind apoi disponibil prin intermediul web service-ului. Este permis accesul mai multor persoane simultan la device, iar interactiunea cu utilizatorul este inlesnita prin intermediul interfeței intuitive si a serverului de VNC care permite vizualizarea grafica a rezultatelor simularii. Tot prin VNC se poate face si inspectarea starii interne a masinii virtuale la orice moment de timp. Folosirea VNC-ului are multe avantaje, unul dintre ele fiind chiar posibilitatea accesarii lui de pe un telefon mobil dotat cu Microsoft Windows Mobile.

Acest mod de lucru, bazat total sau partial pe simulare si accesibil de oriunde, oricand prin intermediul Internetului, poate fi folosit ca o metoda eficienta in cadrul invatamantului la distanta (tip e-learning), nefiind necesara deplasarea persoanei respective, procurarea componentelor etc, rezultatele simularilor fiind in concordanta cu cele reale, iar cunostintele teoretice si practice necesare dezvoltarii unui sistem real putand fi acumulate prin aceasta metoda.

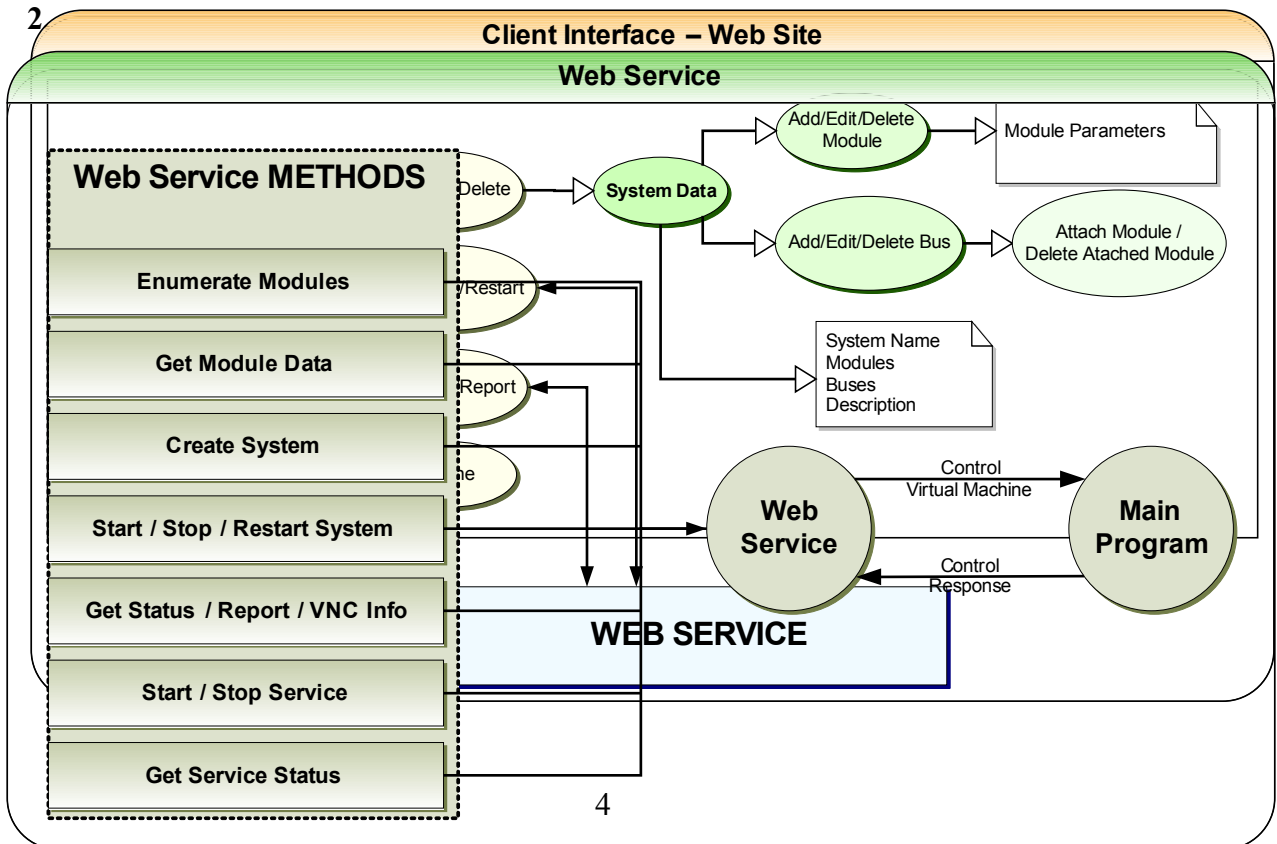
VM nu este limitat numai la folosirea academica el putand fi utilizat si de firme ce dezvolt hardware, permitand testari multiple inainte de implementarea fizica a componentei respective.

Arhitectura

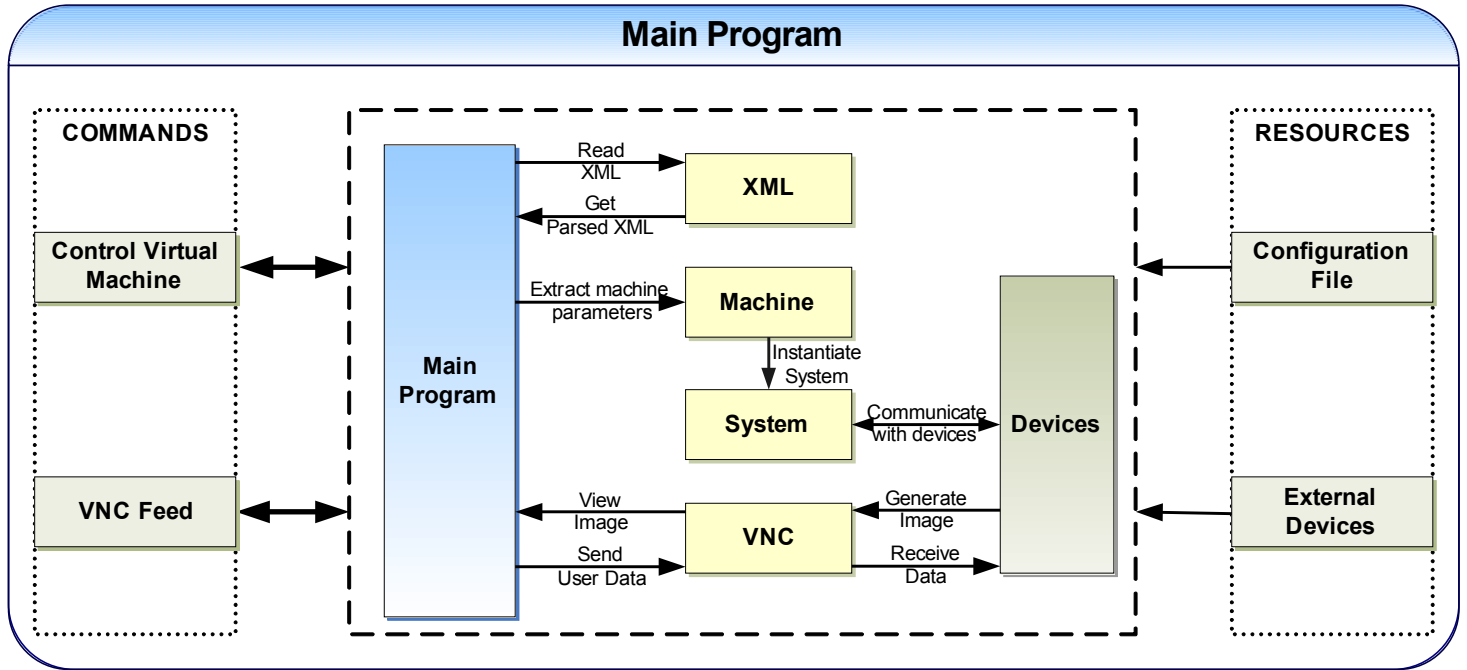


Aplicatia are patru componente majore: Interfata cu clientul ce se realizeaza prin intermediul unui site web, Web Service-ul pentru controlul masinii virtuale, Programul principal ce realizeaza simularea efectiv si VNC-ul ce permite afisarea grafica a rezultatelor simularilor.

- 1.
- 2.



3.



Din punct de vedere al functionalitatii, in cadrul programului principal exista modulul Main Program , el fiind cel care incarca restul modulelor. La pornire acesta citeste fisierul de configurare XML, localizeaza modulele necesare, instantiaza o masina virtuala pe care o initializeaza, initializeaza fiecare modul in parte, iar apoi porneste modulul System. Modulul System genereaza impulsurile de ceas pentru sincronizarea componentelor. In spate exista un thread care parcurge toate componentele si permite acestora sa modifice outputul afisat prin VNC.

Aceste module sunt de baza, ele existand in orice masina virtuala creata, indiferent de tipul sau. Blocul devices contine modulele custom, implementate de utilizator sau deja existente si un singur modul de baza si anume BUS. BUS este un modul pentru conexiuni si permite comunicarea intre componente, simuland astfel o magistrala reala. Exista posibilitatea impartirii unui BUS in mai multe BUS-uri – split, sau a conectarii mai multor BUS-uri intr-unul singur – merge.

Modulele custom pot fi de tip CPU, memory, driver/latch, button/switch, LED-uri, LCD Display etc. Ele sunt stocate in biblioteci dinamice (DLL) si sunt instantiate in momentul rularii conform setarilor din fisierul de configuratie.

Trebuie tinut cont de faptul ca nu sunt simulate si caracteristicile electrice, VM nefiind destinat unei simulari electrice care sa foloseasca impedante, curenti sau alti parametri ai semnalului. De asemenea nu sunt luati in calcul parametri fizici de genul dimensiunilor componentelor si al modului lor de incadrare pe placuta finala.

Tehnologii folosite

Programul principal este scris in limbajul C++, iar pentru compilare am folosit Microsoft Visual C++ si Platform SDK, pentru integrarea cu sistemul de operare Windows si o mai buna folosire a resurselor sale. Pentru parsarea fisierului de configurare XML am folosit parserul Expat, iar pentru interactiunea clientului cu masina virtuala – protocolul standard VNC ce face posibila accesarea masinii inclusiv de pe un telefon mobil dotat cu Microst Windows Mobile.

Web Service-ul este implementat in C# folosind Microsoft Visual Web Developer. Pentru gestionarea fisierelor de configuratii ale masinilor virtuale definite am folosit un parser XML de tip DOM implemetat nativ in C#

Pentru implementarea clientului ce realizeaza integrarea cu web service-ul am folosit php, iar pentru managementul utilizatorilor, baze de date. Pentru realizarea unei interfete placute a fost folosit AJAX, metoda ce permite transferul asincron de date intre server si browser, rezultand astfel o interfata atractiva cu un timp de raspuns imbunatatit. Pentru a completa interfata grafica au fost folosite diverse elemente Java Script, exploatand avantajul de a fi singura tehnica cunoscuta pe marea majoritate a browserelor.

Versiune curenta si proof of concept

Programul principal este complet, simularea fiind functionala. Permite implementarea oricarei masini fiind implementate dll-urile de baza.

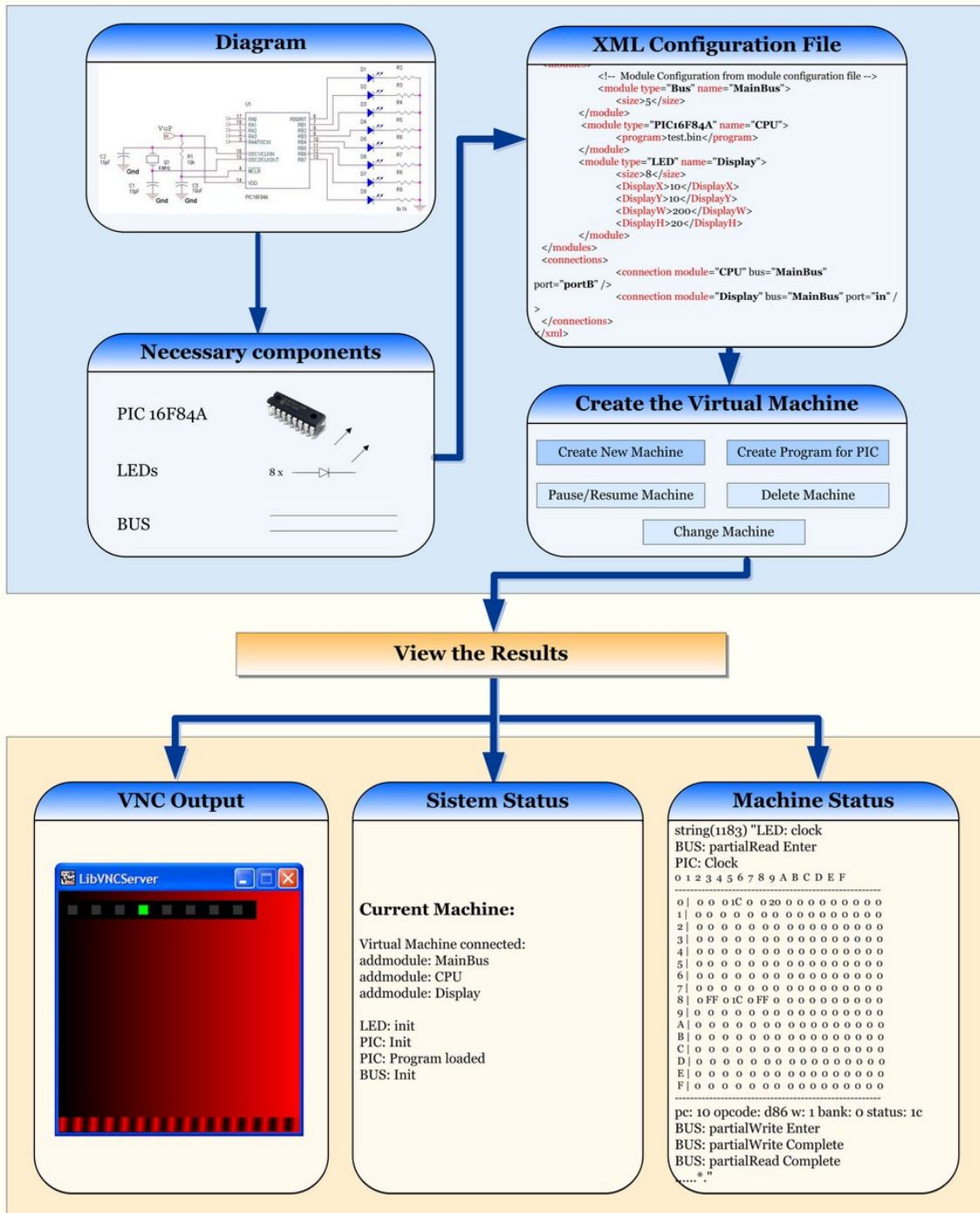
Exista un exemplu deja implementat si anume PIC-ul 16F84A. Acesta a fost ales deoarece are un set redus de instructiuni fiind astfel usor de implementat si ideal pentru exemplificare.

Pentru realizarea sa am implementat 2 noi module PIC si LEDs (pentru output), conectate prin modulul de baza BUS, conexiunea la nivel logic fiind realizata de modulul de baza System. In acelasi timp modulul System genereaza semnal de ceas si asigura data flow-ul intre module.

Pentru testarea PIC-ului am realizat un program de test in assembler care rulat trimite o valoare pe Port A, valoare ce poate fi observata cu ajutorul VNC-ului pe led-uri.

Web Service-ul si Client-ul sunt de asemenea implementate si permit definirea de utilizatori cu diverse drepturi, definirea de noi masini din modulele existente si managementul acestora, crearea de module noi, rularea sistemelor definite de utilizator etc.

Virtual Machine Example



Planuri de dezvoltare ulterioara

Avem in vedere implementarea de noi module a caror interconectare sa permita definirea unor sisteme mult mai complexe, dar arhitectura modulara a aplicatiei are ca principal avantaj faptul ca si utilizatorii pot implementa module pentru componentele pe care doresc sa le simuleze, dand o mare flexibilitate programului.

Dorim sa implementam noi facilitati pentru managementul si in special partajarea masinilor virtuale definite fapt ce va permite o mai buna cooperare in cadrul unei echipe in care fiecare membru va putea implementa anumite module, iar in final prin compunerea lor rezultand un sistem functional.

De asemenea, avem in vedere implementarea de noi functionalitati ale clientului pentru a imbunatati interactiunea cu utilizatorul.