DynamicUserInterfaceswithJava

FrankKargl
frank.kargl@informatik.uni-ulm.de

TorstenIllmann
torsten.illmann@informatik.uni-ulm.de

MichaelWeber
weber@informatik.uni-ulm.de

StefanRibhegge
stefan.ribhegge@informatik.uni-ulm.de

DistributedSystemsDepartment
UniversityofUlm
Germany

Abstract: Whiledevelopingadistributedinfrastructureforpersonalagentswewanttogive allkindsofagentsapossibilitytocommunicatewiththeiruserbymeansofacentraland consistentuserinterfaceintegratedintheirwebbrowser.Newagentsshouldbeabletojoin thisinfrastructureinadynamicmannerwithoutneedtochangetheframeworkandthere shouldbenobuilt-inrestrictionstothecomplexityoftheuserinterface.Wedescribeaspecial usercommunicationagentthatallowsagentstodescribetheiruserinterfacedynamically, displaytheuserinterfacesforanynumberofagentsinaspecialappletandhandleremote eventdelegation.Theresultisanewkindofultra-thinclientwherethewebbrowserandJava Appletmerelyworksasagenericdisplayserver.

Introduction

Ourresearchgroupiscurrentlystudyingdifferentaspectsofsoftwareagentsystemsinaprojectcalled CIA [Karglet.al.99].Wearedevelopinganinfrastructureintowhichsoftwareagentscaneasilybeintegrated.Inour projectallagentsbelongingtooneuserformasocalledAgentCluster.Thisclustersupportstheagentswithall kindsofcommonlyneededserviceslikedifferentcommunicationmodels,persistentstoragecapabilities,security servicesetc.TheagentsareabletocommunicateviaasoftwarebussystemcalledAgentBuswhichwe implementedontopoftheiBussystem[Softwired1999].

Forinteractionbetweenusersandagentsweidentifiedanumberofpossiblemechanism:

- Eachagenthasalocaluser-interface.Themajordisadvantageofthissolutionisthatauseralwayshastobe atthecomputertheagentisrunningonifhewantstointeractwithit.
- Eachagenthasaspecificcontrolapplet.NowauserisabletocontroleachagentfromwithinhisWeb Browser.Howeverthereisnointegrationofthedifferentagentuserinterfaces.Ifauserwantstocontrol manyagentsatoncehehasalotofwindowsonhisdesktop.InteractionbetweenthedifferentagentUIsis difficultaswell.
- There'sacentralcontrolapplicationforallagentsthatisaccessibleviaanapplet.Thisapproachcanbe founde.g.withmostcurrentnetworkadministrationsystems.Agentscollectdataonallthemanagedhosts andtransmitittoacentralmanagementapplication.Theusercanattachtothissingleapplicationusingan appletorastandaloneuser-interfaceapplication.Thisapproachworksonlyifyouhavesimilaragents runningonmanyhosts.Howeverifyouragentsareveryheterogeneousandnewagentswithnew capabilitiesandnewuserinterfaceswilloccuronaregularbase,thecentralcontrolapplicationhastobe adaptedeachtime.

- There’sagenericdisplayapplicationorappletthatisindependentoftheagent’suserinterfaces.Theagents usethedisplayapplicationtodisplaytheiruserinterfaceandrelayuserinputback.

Wehavechosenthelastapproachaswewantedtoavoidthedisadvantagesoftheotherapproaches.Itisrealized intheUserCommunicationAgent(UCA).Thisagenthandlesdisplayofthegraphicaluserinterface(GUI) integratedintothewebbrowser,userauthenticationandmanythingsmore.Ontheotherhand,theUCAhasno knowledgeofthespecificdetailsofthedifferentagentsinthecluster.

WithinanAgentClusterthereisanalwayschangingnumberofdifferentagents.Eachoftheseagentshas differentdemandsregardingitsGUI.Someagentsmaywanttodisplayonlyasimplestatusmessageoratextlist andhavenouserinteractionwhereasothersneedtheuserinterfaceofafullscaleapplicationwithdifferent windows,menus,interactiveelementsetc.Theuserinterfacesofdifferentagentsshouldn'tinterfere inadvertently.Ifexplicitlyneededcommunicationbetweendifferentcomponentsofdifferentuserinterfaces shouldbepossible.Thereisacommonpartintheuserinterfacethat'sindependentofsingleagents.Functions likelistingallactiveagents,connecting/disconnectingfromtheAgentCluster,userauthenticationetc.fallinthis category.ThisfunctionalityshouldberealizeddirectlyintheUCAwithoutneedtoreimplementitforeach agent.

GeneralDesign

TheUCAispartitionedintoagenericJavaAppletthatrunsintheuser'swebbrowserandaservermodulethat connectstotheAgentBusandhandlescommunicationwithotheragents.[Figure 1]showstheprincipal architectureofthewholesystem.Thedetailswillbeexplainedthroughoutthistext.

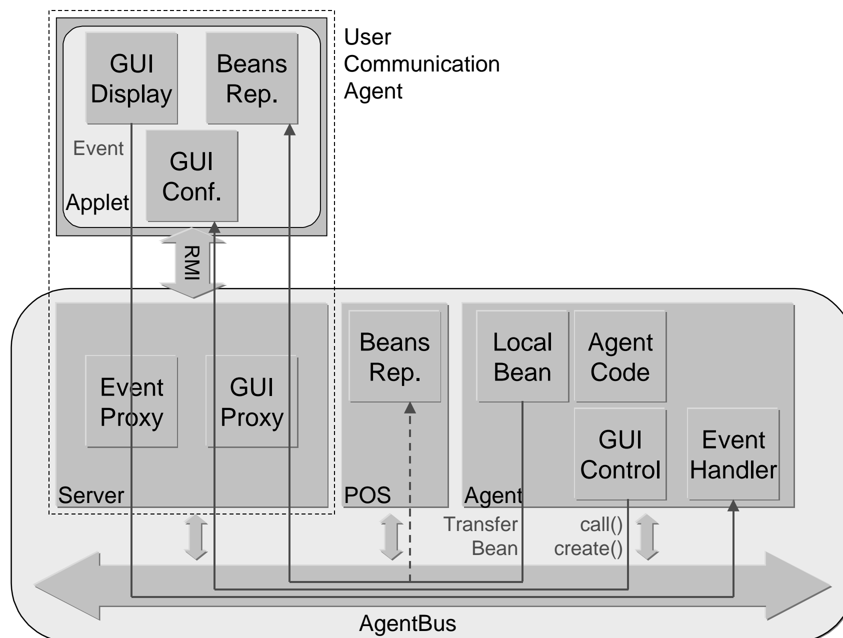

Figure 1:SystemArchitecture

FormaximumflexibilitywemodeledoursysteminaccordancetoamodifieddistributedDisplay-Control-Model (DCM)designpattern[Buschmannet.al.1996].TheUCAisresponsibleforthedisplaypart.Thecontroland modelpartsareusuallyimplementedintheagentinformoftheGUIControlandtheAgentCodeitself.For performancereasonspartsoftheGUIControlmaybetransferredtotheappletinformofJavaBeans.Our frameworkprovidesfortheseamlesscommunicationbetweenthepartshidingnearlyallaspectsofdistribution.

IneffectthewholesystemresemblesalittlebittheX11-system[Scheifleret.al.1992]withtheagentbeingthe X11-clientandtheUCAbeingtheX11-server.IncontrasttoX11oursystemiswrittenentirelyinJava,is completelyobject-orientedandusesadvancedcommunicationmechanismslikesoftwarebusesandRemote-Method-Invocation(RMI).Anothermajordifferenceisthatweareabletotransfermobilecodetothedisplay appletwhichcanbeusedforenhancedfunctionalityofperformance.

Whiledesigningthegenerallayoutofthismodel,weidentifiedthreemajorproblems:

- Theappletmustbeflexibleenoughtodisplayanyuserinterfaceimaginableforanynumberofagentsin parallel.
- Theagentsneedawaytodescribethecompositionandstructureoftheiruserinterfaceatrun-time.
- Userinteractiongenerateseventslikebuttonpushesetc.thatmustbedeliveredtotheappropriateagent.

OurprototypeimplementationoftheinfrastructureisbasedonJava2soaJava2solutiontotheseproblemshad tobefound.

UCAApplet

[Figure 2]showstheappletwithanactivediaryagentdisplayingitsuserinterface.Eachagentisassigneda separatetabwhereitcandisplayitsuserinterface.Asthistabisageneralcontainerthatmakesnopre-assumptions,agentsarecompletelyfreetobuildtheiruserinterface.Theycanevenspawnnewwindowsthatare totallyindependentoftherest.

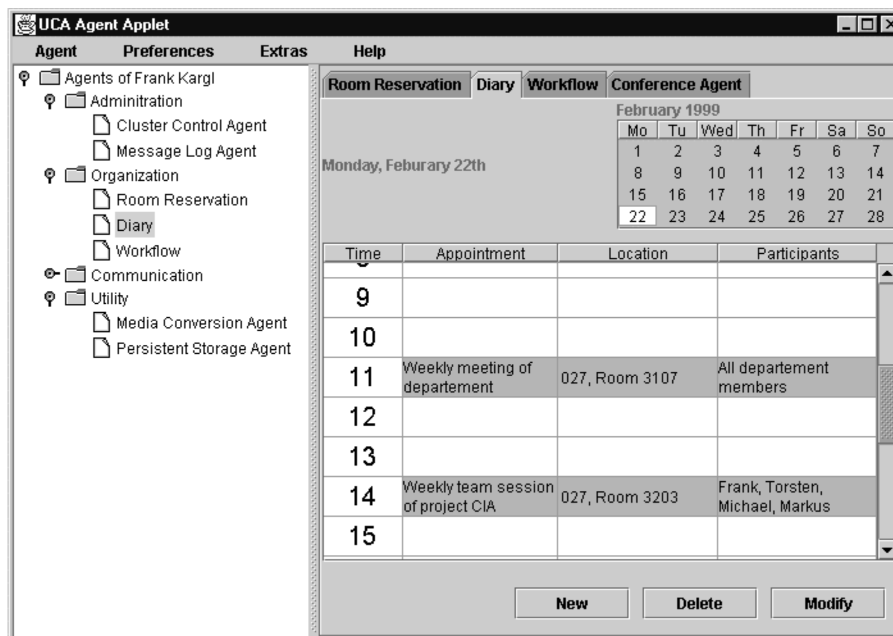

Figure 2:TheUCAApplet

UCAServer

Theserverpartconsistsprimarilyoftwoproxies,theGUIandEventProxy,thattranslateAgentBuseventsto RMIcallsandviceversa.Whenanagentwantstodoanychangestoitsuserinterface,itgeneratesspecific AgentBuseventsthataretranslatedtoRMIfunctioncallsintheGUIProxy.Inreversedirectionevents generatedbytheuserinterfacearecaughtbyRMIevent-handlersandarethentranslatedtoAgentBuseventsin theEventProxy.

DynamicInterfaceConstructionandManipulation

Asexplainedabove,eachagentisassignedanemptycontainerwithinatabbedpaneintheUCAappletwhereit canconstructormanipulateitsuserinterface.Itdoessobyusingour *DynamicInterfaceConstructionand Manipulation* mechanism(shortDICM)whichsuppliesthefollowingpossibilities:

- Creatingnewcomponentsusing *create()*
- Manipulatingexistingcomponentsusing *call()*
- IntegratingexistingJavaBeansatruntimeintotheapplet

AllthesemechanismsworklocallyaswellasremoteviatheAgentBusorRMI.Foreaseofusewehave encapsulatedDICMinacompletesetofremotecomponents,soauserjustneedstobuildalocalcomponent hierarchythatisautomaticallymirroredanddisplayedintheUCAapplet.

Creatingnewcomponentsusing *create()*

Startingfromtheinitialcontainer,remoteobjectscancreatenewcomponentsintheappletusingthe *create()* call.Thecalltakesthreearguments:abasecontainerwherethecomponentisaddedto,theclassofthe componenttocreateandanamethatisassignedtothenewlycreatedcomponent.AseveryAWTcomponenthas anameattributeandallcomponentsarearrangedinahierarchy,wecanaccesseverycomponentstartingfroma singlecontainer.Anexampleusageof *create()* maybe:

```
guicont.create("startcontainer.firstpanel", "javax.swing.JButton", "okbutton");
```

Thiswillcreateanew *JButton* component,nameit"okbutton"andadditthecomponentnamed"firstpanel" containedin"startcontainer".

Modifyingexistingcomponentsusing *call()*

Ifwewanttomodifyexistingcomponentswecanusethe *call()* method.Weneedtosupplythreearguments:the nameofthecomponenttomodify,themethodtocallwithinthecomponentandanargumentlisttothemethod. CallinvocationisdoneusingtheJavaReflectionmechanism[Sun1999].Thefollowingcodewillsetthetext attributeinthebuttoncreatedabove:

```
String[] args = { "new buttontext" };
Object result =
     guicont.call("startcontainer.firstpanel.okbutton", "setText", args);
```

TransferringexistingJavaBeans

Both *create()* and *call()* workwellforsmallgraphicalinterfacesorslightmodificationstoexistingonesbutitis quitecumbersomebuildinglargeuserinterfacesthisway.Consequentlythereisathirdwayforanagentto displayandmanagecomplexstructuresintheUCA:mobilecode.Acompleteuserinterfaceorcomplexsubparts

ofitcanbedevelopedasoneorseveralJavaBeans.AnagentthentransfersthecompletebeanintoaBean RepositorylocatedintheUCAappletandinstantiatesitasneeded.Subsequentmethodcallstothisbeanare againdoneusingthe *call()* methoddescribedabove.Thetransportofbeancodetotheappletisaquitecomplex process.Inourcurrentprototypetheagenttransmitsthebeancode(readinfromtheclassfile)totheGUIProxy viatheAgentBus.ThebeanisthensenttotheBeansRepositoryintheapplet.There'sacustomclassloaderfor instantiatinganyofthebeansintherepository.

InafutureimplementationwewilluseJavaSpaces[Sun1999]forthesamepurpose.Beanscanthenevenbe storedinaPersistentObjectSpace(POS)[seeKarglet.al.1999]andcanberetrievedasneeded.Beansare instantiatedintheappletusingacustomclassloader.

RemoteEventModel

Nowthatwehaveconstructedourremoteinterfaceweneedawaytorelayusergeneratedevents(likemenu selectionsetc.)backtotheagent.Startingwithversion1.1Javausesaneventdelegationmodel.Components (likebuttons,menusetc.)generateeventsthataredeliveredtopreviouslyregisteredeventhandlers.Wehave extendedthismodelsoeventhandlerscannotonlybelocal(thatiswithinthesamevirtualmachine)butalso remoteeventhandlerscanbeused.ThisrequiresslightmodificationsofthestandardAPI(makingevent handlersthrowjava.rmi.Remoteexceptions)butworksabsolutelyfineotherwise.ThechangedstandardAPIis onlyusedfortheproxiesandtheapplet.AgentsaredevelopedusingthenormalJDK.Nowanagentsimply registersitseventhandlerroutinewiththerespectivecomponentinordertogetnotificationofallevents generatedbythiscomponent.There'saeventproxyfortranslatingtheRMIcallstoAgentBusevents,sothe remoteeventmechanismisevenindependentoftheunderlyingtransportmechanism.

Wehaveconductedvariousperformanceteststhatshowtheeffectivenessofourapproach.[Table 1]showsthe resultsmeasuredincallspersecond.Thefirsttestcalled *LocalEventHandler* createsacomponentandfires actioneventstoalocaleventhandlerroutine. *RMInoargument* callsamethodusingRMIandpassesno arguments. *RMIActionEventargument* doesthesamebutpassesanActionEventobjecttothemethodcalled. Finally *RemoteEventHandler* doesthesameas *LocalEventHandler* butcallsaremoteeventhandlerroutine. Alltestsweredoneusing200MHzPentiumSystemsrunningWindowsNT4.0,JDK1.2.Themachineswere connectedtoalocal10MbitEthernetsegment.

Calls/s	Local Event Handler	RMI noargument	RMI ActionEvent argument	Remote Event Handler
local	6405	839	372	386
remote	N/A	831	358	358

Table 1:PerformanceTests

Howcanweinterprettheseresults?Remoteeventhandlershaveapotentialthroughputofabout350eventsper second.Asthe *RMIActionEventargument* and *RMInoargument* testsindicate,thisseemstobeprimarilydueto theRMIcallandmarshaling/unmarshalingoftheActionEventargument.UsingsuchRMIcallsforEvent Handlersaddsnoadditionaloverhead.Although350eventspersecondaremuchlessthanthemaximumof6400 inalocalscenario,itisenoughgiventypicaluserinterfacescenarios.Normallyeventsaregeneratedbymouse-clicksandnouserwillproducemorethanoneortwoclicksasecond.Evenresizeeventsthataffectmany componentsatoncearen'tcriticalgiventhatthereareseldommorethan50componentsinoneuserinterface. Ourprototypeconfirmsthisasanordinaryusercan'tdetectanysignificantdifferencebetweenlocalandremote eventhandling.Ifanyuserinterfacewouldgetintoperformancedifficultiesduetoeventhandling,itcanstillbe integratedintoabeanwhichcanbeexecutedlocallyintheapplet.

ConclusionandOutlook

Displaysystemsliketheonedescribedinthispapermaybecomenecessarywheneveryouwanttocontrolor operateadistributedsystemofheterogeneousapplications(notonlyrestrictedtosoftwareagents)fromwithin oneapplicationorapplet.OtheragentsystemslikeIBMAglets/Tahiti[IBM1999]oftendon’taddressthis problem,astheyonlyallowthedisplayofuserinterfacesfromagentsresidingonthelocalhost.

Asourworkclearlyindicatesitispossibletorealizeultra-thinclientswithJavathathavepracticallyno applicationspecificlogicbuilt-in.Insteadtheymerelyrepresentahighly-functionaldisplayserversimilartothe wellestablishedX11-servers.DuetothebroadavailabilityofJavaenabledbrowsersoursystemcanbeused nearlyeverywhere.Byusingasetofremotecomponents,applicationscanusethis"displayserver"inanearly transparentmannerthroughaneasytouseinterface.Whentransferringcomplexcomponentstothedisplay applicationthecommunicationoverheadcanbeminimized.

Whenrealizingthebeantransferandremoteeventmodelyouofteninterferewiththesecuritymechanismsin Javaappletssopolicyfilesarenecessaryforgrantingcertainrightstotheapplet.Theefficientinstallationof thesepoliciesattheuserisoneoftheunsolvedproblemsyet.

OurnextextensionwillbeanevenmoregeneralizedDICMmechanismsouserinterfacesdescribedbyagents willautomaticallyadapttoawiderangeofpossibledisplayslikePCs,PDAsorpossiblyevenvoicecontrol.

References

Buschmann,F.et.al.(1996). *Pattern-OrientedSoftwareArchitecture.* NewYork:JohnWiley&Sons,Ltd.

IBM(1999).http://www.trl.ibm.co.jp/aglets/tahiti1.1/index.html

Kargl,F.&Illmann,T.&Weber,M.(1999).CIA-aCollaborationandCoordinationInfrastructureforPersonal Agents. *DistributedApplicationsandInteroperableSystemsII* ,1999,IFIPTC6WG6.1.213-218.

Scheifler,R.W.&Gettys,J.(1992). *XWindowSystem* .Burlington,MA:DigitalPress

SoftwiredInc.(1999).http://www.softwired.ch/products/ibus/

SunMicrosystemsInc.(1999). *ReflectionFAQ* .
http://java.sun.com/products/jdk/1.1/docs/guide/reflection/faq/faq.html

SunMicrosystemsInc.(1999). *JavaSpacesSpecification* .
http://java.sun.com/products/javaspaces/specs/index.html