



Analysetool CODEGRABBER

Der Weg aus dem Labyrinth von Code und Applikationen

Joachim Mutter
Waldstraße 4
66506 Maßweiler
Telefon + 49-172 - 7383525
Fax +49-6334 - 983405
EMail joachim.mutter@sysarc.de

IT.SYSTEM
architektur Dipl. Ing. Inf. Joachim Mutter

Veränderungen am Domino Client / Server

Bei einer Notes-Infrastruktur stellt sich immer wieder die Frage:

Werfen bestehende Applikationen Probleme auf, wenn am Domino Client/Server Veränderungen vorgenommen werden?

Abgesehen von den Haupt-Anwendungsfällen Mail, Calendaring und Scheduling gibt es bei den meisten Installationen zusätzliche Applikationen (make or buy), die als unternehmenskritisch gelten und deren Ausfall zu schweren Störungen des Betriebsablaufes führen kann.

Aufgabenstellung

Erschwerend kommt hinzu, dass die Administration häufig nicht in vollem Umfang weiß, wie viele unterschiedliche Applikationen im Unternehmen eingesetzt werden und wie komplex deren Zusammenwirken ist.

1 Migration einer 4.x Installation auf eine 5.x Installation

Dies mag ausgesprochen einfach zu bewerkstelligen sein, da alles abwärts kompatibel ist. Allein bei RNext kann es schon ganz anders sein, da hier ein komplett neuer @Function Interpreter hinzukommt und LotusScript ebenfalls stark erweitert wird.

2 Konsolidierung der vorhandenen (NT-) Server und anschließender Umzug auf eine andere OS-Plattform (Solaris, AIX, OS390, ...)

Hier müssen erfahrene Programmierer den Code der Applikationen durchsehen und eventuell Teile davon portieren, die so auf der neuen Zielsystem-Plattform nicht mehr funktionieren (ODBC, File-Zugriffe, etc). Dies kann sowohl die Syntax (File-Zugriffe) als auch die Semantik (COM, ODBC,...) betreffen.

3 Einsatz eines Notes-Clusters (unabhängig von Zielsystem)

Diese Systemänderung ist kritisch und bedeutet einen großen Aufwand hinsichtlich Analyse und späteren Codeänderungen. Es muss ein Code-Review durchgeführt werden, wobei das Hauptaugenmerk auf serverbasierende (scheduled, triggered) Agents und failover-sichere Befehle gelegt werden muss. Nötige Anpassungen der einzelnen Applikationen können hier größere syntaktische und semantische Änderungen des Source Codes bedeuten.

Die große Schwierigkeit bei den Aufgaben 2 und 3 ist es, eine Analyse durchzuführen, die aufzeigt, welche Applikation an welcher Stelle Probleme bereiten könnte. Dabei muss jeder Code (@Formulas, Lotus Script) verifiziert und auf entsprechende Fehler untersucht werden.

Dies kann man zum Beispiel manuell per Designsynopsis und anschließender Textsuche in einem Textprogramm durchführen, was allerdings mühsam und fehlerträchtig ist, da die Menge von gesuchten Funktionen umfangreich ist.

Oder aber man verwendet Tools von Drittanbietern, die, entsprechend konfiguriert, eine Applikation auf Funktionen hin untersuchen und einen Report ausgeben, wo eventuelle Schwachstellen im Code zu suchen sind.

Ein kurzes Beispiel zeigt den Aufwand exemplarisch auf:

Eine Firma muß 100 Applikationen hinsichtlich eines zu installierenden Cluster-Betriebs prüfen. Dabei ergibt sich folgender Aufwand:

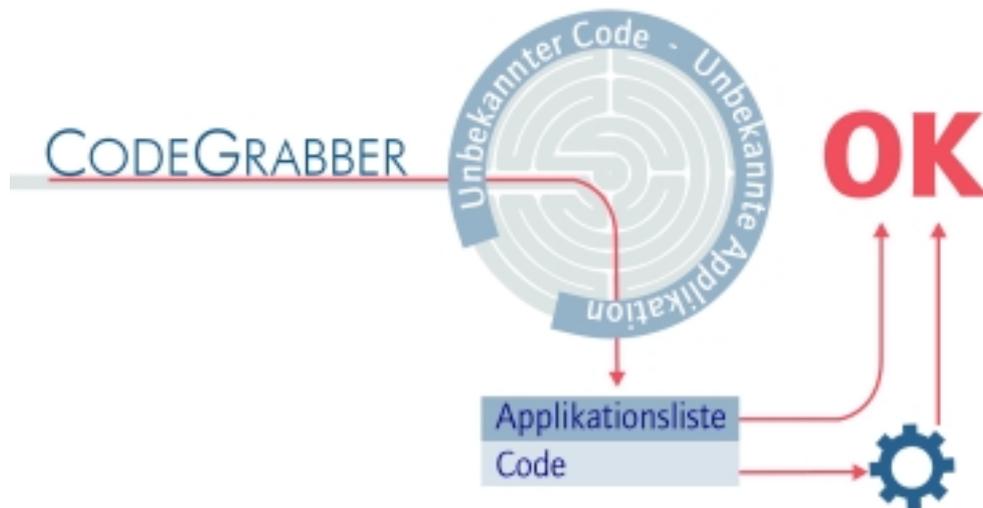
20	direkt ersichtliche Standard-Applikationen, die wir nicht beachten (was allerdings auch nicht unbedingt stimmen muss: Serverkonsolidierung und Betriebssystemänderung)	1 Tage
60	abgeleitete Standard-Applikationen und Repliken, geschätzter Aufwand für einen erfahrenen Entwickler	3 Tage
20	10 selbst entwickelte und 10 gekaufte Applikationen Analyse pro Applikation ca. 1 Tag (je nach Komplexität unterschiedlich)	20 Tage

Gesamtaufwand

24 Tage

Dies stellt nur eine grobe Schätzung dar, die je nach Komplexität der Applikation schwanken kann, liefert jedoch einen realistischen Zeithorizont.

Hierbei sind keine Codeanpassungen oder auch nur Gedanken enthalten, wie solche Anpassungen gerade im Hinblick auf Server-Agenten und Clustering durchzuführen sind.



Leistung

CodeGrabber scannt ganze Notes-Domänen, sucht automatisch nach entsprechenden Codesequenzen und klassifiziert sie.

Diese Klassifizierung wurde anhand von jahrelangen Erfahrungen festgelegt und zeigt auf den ersten Blick, wie komplex Ihre Applikationen sind und inwieweit Codeänderungen gemacht werden müssen.

Sie haben mit dieser Anwendung alle Informationen zur Hand, die Sie für Analyse, Kostenschätzung und Vorbereitung für eine spätere Codeänderung benötigen. Die Administration hat entsprechende Mittel zur Verfügung, um die einzelnen Applikationen als bearbeitet zu kennzeichnen, Kommentare hinzuzufügen, etc.

Die Menge der zu scannenden Applikationen kann per Dialog eingegrenzt werden, wobei Sie die Möglichkeit der Grobselektion hinsichtlich einzelner Datenbanken bzw. Server, Serverlisten oder ganze Domänen zur Verfügung haben. Diese Menge kann dann per Filepattern oder sonstiger Merkmale wie Design Templates, View- oder Formnamen weiter reduziert werden.

Die zu suchenden Funktionen sind per Konfiguration hinterlegt und können als Notes Regular Expressions definiert werden, wobei die Syntax erweitert wurde, um auf die speziellen Gegebenheiten eingehen zu können.

Debugging und Fehler-Log sind fein granulierbar einstellbar, spezielle Funktionen ermöglichen ein Neuaufsetzen nach einem Fehler (z.B. zu wenig Hauptspeicher beim Scannen einer Domäne etc.).

Der Einsatz dieses Tools spart ganz klar Kosten ein. Setzt man nur den Aufwand vom Beispiel oben an, reduziert sich der Aufwand auf weniger als einen Tag, bis man die benötigte Liste in der Hand hat. Bis zu diesem Zeitpunkt haben Sie auch noch keine hoch-bezahlten Spezialisten benötigt, sondern nur jemanden, der sich einigermaßen mit Notes und Ihrer Domäne auskennt. Erst ab jetzt benötigen Sie jemanden, der die Liste verifiziert und eventuell einige Applikationen in eine weniger relevante Klasse abstuft.

Sie benötigen also kein Heer von erfahrenen Entwicklern, um sich über die Komplexität Ihrer Anwendungen zu informieren. Somit kann dieses Tool auch zur Entscheidungsfindung für den Einsatz eines Clusters oder einer anderen Serverplattform in positiver als auch negativer Richtung dienen.

Database	Class	Template	Replica Count	Views	Agents	Forr
▶ DESL01/DeSL			1.00	60	59	
▼ JoeSrv/Lap			1.01	1204	702	1
admin4.nsf	B;E	StdR4AdminRequests	1	21	2	
AgentRunner.nsf	A		1	1	0	
bookmark.nsf	A	Bookmarks	1	10	8	
catalog.nsf	B;D;E	StdNotesCatalog	1	20	2	
certsrv.nsf	D	StdNotes50SSLAdmin	1	6	0	
decsadm.nsf	B;D	DECS Administrator Template	1	23	8	
domlog.nsf	E	Domino Web Server Log Template	1	1	1	
DomCfg.nsf	A	Domino Web Server Configuration Template	1	11	0	
events4.nsf	D	StdR4Events	1	46	8	
headline.nsf	A	StdNotesHeadlines5.0	1	1	1	
homepage.nsf	A		1	0	0	
log.nsf	B	StdNotesLog	1	12	0	
log_old.nsf	B	StdNotesLog	1	12	0	
mtatbls.nsf	A		1	6	0	
MigrAppl.nsf	D		1	0	1	

Infos of "JoeSrv Stats/Lap"

Database class A B C D E ▶ Servers with replicas

Working state Unprocessed Processed Adaption needed No change needed

Replica ID : 802567520070B6C6 View Count : 30

Replica Count 1 Form Count : 103

Database Size : 7168 Agent Count : 47

Template Name : StdR50Mail Critical Code ▶ Design elements

Needed Changes

Class Infos

Agents (AMgr)

```
OutOfOffice | OutOfOffice
ASSISTTRIGGER_TYPE_SCHEDULED [Once a day],
Start = 01:00, overall execution = -
```

Agents (Web)

'(wFolderCreateSave)' is a WebQuerySave/Open agent

Forms with Author Fields

```
On 'Reply' Author Field 'From' found
Formula Section : DV={FIELD Principal :=
@GetProfileField(„CalendarProfile“; „Owner“);
@UserName} - IT={ } - IV={ }
```

Critical Code in Libraries

```
CoreEmailClasses
Function FaxGetDomain ()[*..OPEN(*)]
If Not(db.IsOpen) Then Call db.Open(AddressBook,"")
```

Critical Code in Views

```
($Calendar) | Calendar
Action : 'New Calendar Entry'
@Function [*@*[[COMPOSE]*:*]*) @If(@ClientType
= „Notes“; @Environment(„CSDocType“;CSDocType) :
@Command([Compose];"Appointment") :
@Return(0);"");
```

Critical Code in Agents

```
Archive\Open Log
@Function [*@*[[FILEOPEN]*:*]*)
@Command([FileOpenDatabase];ARCSERVER:ARCLOGDBPATH);
@Prompt([OK];xxTitle;xxMessage))
```

```
(MigrateOrgGS)
Declaration section
Declaration [DECLARE * LIB *] Declare Public
Function OSLoadProgram Lib „nnotes.dll“ Alias
„OSLoadProgram“ (Byval filename As Lmbcs String,
Byval WorkingDir As Lmbcs String, Byval args As
Lmbcs String, Byval flags As Integer) As Integer
```

Critical Code in Forms

```
$ACTIONS
Action : 'Schedule a Meeting'
@Function [*@*[[COMPOSE]*:*]*) @If(@ClientType
= „Notes“; @Environment(„CSDocType“;CSDocType) :
@Command([Compose];"Appointment") :
@Return(0);"");
```