Stageverslag

Python Framework Installer

Pieter-Jan Robrecht

Student Industrieel ingenieur ICT
01/08/2016 - 19/08/2016

Inhoudsopgave

1 Inleiding

In het kader van mijn masterproef was het mogelijk om bij Televic Rail een stage te lopen tijdens de zomer. Gedurende deze stage heb ik mijn masterproef voorbereid en onderzoek gedaan naar de verschillende mogelijkheden die er zijn voor het uitvoeren van mijn opdracht. De opdracht die voltooid moet worden is de volgende:

> Televic Rail heeft een Python test framework ontworpen. Dit framework werd oorspronkelijk gebruikt op een volledig uitgeruste testtoren, maar het framework werd aangepast zodanig dat het onafhankelijk van de testtoren gebruikt kan worden. Aangezien het framework gebruik maakt van verschillende niet-standaard Python bibliotheken, is het installeren van het framework op een nieuw systeem een hele klus. Het doel is dan ook het maken van een installer-updater waarmee dit proces kan worden vergemakkelijkt zodanig dat er zo min mogelijk interactie van de gebruiker moet zijn.

Tijdens de duur van de stage werden de verschillende implementatie manieren onderzocht en werd er een algemene architectuur voor de installer-updater uitgedacht. In wat volgt wordt er beschreven welke stappen er werden ondernomen om het probleem onder te verdelen in enkele logische onderdelen en hoe deze het probleem dan oplossen.

2 Analyse van het probleem

Tijdens het bespreken van het probleem werd het al snel duidelijk dat er verschillende scenario's zijn waarin het framework gebruikt wordt. Het framework wordt als een standalone programma gebruikt op een laptop of desktop maar het zal ook moeten draaien op verschillende testtorens. Voor de verschillende omgevingen zal iedere keer een andere configuratie nodig zijn maar beiden omgevingen gebruiken Windows als besturingssysteem. Uiteraard moeten we ervan uitgaan dat deze situatie slechts tijdelijk is. Er wordt best een applicatie geschreven die op verschillende types van systemen kan draaien. Ook zorgen we er best voor dat het programma besturingssysteem-onafhankelijk is.

Tijdens het installeren van het framework moeten we rekening houden met het feit dat iedere computer een andere configuratie en andere drivers zal nodig hebben. We zullen de gebruiker moeten vragen om de verschillende drivers te selecteren. Mocht het mogelijk zijn dan zou er een mapping kunnen gebeuren tussen de verbonden devices en de nodige drivers zodanig dat deze automatisch worden aangevinkt in een selectielijst. Het probleem van de drivers beperkt zich jammer genoeg niet enkel tot dit. Als er nieuwe hardware ontwikkelt wordt, zullen er ook verschillende nieuwe drivers nodig zijn. De lijst van beschikbare drivers zal dan tijdig moeten worden aangepast.

De installer is best ook een programma dat zo eenvoudig mogelijk uit te breiden is zodanig dat het in de toekomst nog kan worden aangepast. De interface naar de gebruiker toe is best ook zo eenvoudig mogelijk zodanig dat er geen problemen kunnen ontstaan tijdens de installatie van het framework. Uiteraard gaan we er vanuit moeten gaan dat er af en toe een probleem zal ontstaan tijdens de installatie. Updates zullen dus dan in een afgesloten omgeving moeten gebeuren en na het correct uitvoeren in het grote geheel geplaatst worden.

De updater bezit gelijkaardige problemen. We zullen een manier moeten zoeken waarmee we gemakkelijk kunnen controleren naar de versienummers van de software. De gebruiker zal ook de optie moeten krijgen om de update uit te voeren. De gebruiker moet zelf beslissen wanneer de update uitgevoerd zal worden zodanig dat de update gebeurd als de gebruiker er klaar voor is.

3 Oplossingen

Aangezien we een algemeen beeld hebben van wat we juist allemaal moeten voorzien, kunnen we vervolgens aan de slag met het zoeken naar een goede oplossing voor alle problemen. In wat hierna volgt gaan we alle mogelijkheden overlopen die gebruikt kunnen worden om de installer-updater te implementeren. Alle verschillende opties gaan worden overlopen en de verschillende voor- en nadelen zullen besproken worden[1]. Van hieruit gaan we ook een opsplitsing maken tussen de installer en de updater. We gaan bij het zoeken naar een op-

[1] Uiteraard is het mogelijk dat tijdens de duur van de masterproef nog mogelijk oplossingen worden gevonden. In sectie 3.1 op de volgende pagina worden enkel de oplossingen besproken die tijdens de stage zijn gevonden.

lossing wel een opdeling maken tussen de oplossingen voor de installer en voor de updater. Voor beiden zijn er oplossingen aanwezig en er zal gekeken moeten worden welke combinaties mogelijk zijn.

3.1 Mogelijkheden

Laten we eerst kijken naar de besturingssysteem-onafhankelijkheid van het framework. De besturingssysteem-onafhankelijkheid van Pyhton hangt af van de code en van de bibliotheken die gebruikt worden. Zolang deze bibliotheken onafhankelijk zijn van het besturingssysteem is er geen enkel probleem. Java biedt ook een oplossing aan voor dit probleem aangezien Java ook volledig besturingssysteem-onafhankelijk is. Jammer genoeg zijn er amper tot geen programma's die een installer/updater kunnen genereren in Java. Alle code voor zo'n programma zal zelf geschreven moeten worden.

3.1.1 Qt Installer Framework [8]

Het Qt framework zou in staat zijn om eenmalig installers te maken die vervolgens op de ondersteunde platforms zou kunnen draaien. De ondersteunde platforms zijn: Linux, Microsoft Windows en OS X [10]. De software zou de gebruiker door het installatie, update en verwijderproces leiden. De programmeur moet enkel de nodige informatie over de te installeren software. Er kunnen scripts geschreven worden voor de installer zodanig dat er verschillende opties toegevoegd kunnen worden. bij iedere component kan een script toegevoegd worden zodanig dat de installatie kan worden gepersonaliseerd [9].

Het Framework biedt twee types installers aan: een online en een offline installer. Beide installers zullen een maintenance tool installer die gebruikt kan worden om componenten te updaten, toevoegen en verwijderen. Het verschil tussen de twee installers is het volgende: de offline installer zal alle nodige componenten bevatten in de installer. De online installer moet tijdens de installatie een verbinding maken met een online repository waar de nodige componenten worden gedownload. Bij de offline installer zullen alle componenten geïnstalleerd worden terwijl bij de online installer de keuze is om verschillende componenten niet te installeren [10].

Er zal wel moeten gekeken worden of dit een goede oplossing is voor ons probleem. Tijdens de installatie van het framework en drivers wordt er gebruik gemaakt van verschillende Windows executables. De installer zal deze moeten uitvoeren en tot een goed einde brengen. De grote vraag is dan: is het mogelijk om een installer te maken met deze executables in die ook draait op Linux? De installer zal ook de verschillende zip files moeten kunnen uitpakken en de setup uitvoeren.

3.1.2 WiX [14]

Windows Installer XML Toolset is een toolset waarmee Windows Installer packages gemaakt kunnen worden vanuit XML code. Met de toolset is het mogelijk

om een installer met als extensie .msi te maken. De installer zal dus enkel bruikbaar zijn in Windows. Met WiX is het mogelijk om een installer te schrijven die code van anderen gebruikt. Het is dus mogelijk om executables te includeren in de installer [15].

Dankzij het patch systeem moet het mogelijk zijn om updates uit te voeren. Er zal wel onderzocht moeten worden of een automatische version check kan uitgevoerd worden bij het opstarten. Het schrijven van een patch voor ieder update die uitkomt is geen optie. Als er dus geen automatische version check en installatie aanwezig is, dan zal gekeken moeten worden om een updater te includeren in de installer. Deze updater zal dan bij het opstarten van de software zoeken naar de nieuwe updates en eventueel de mogelijkheid geven om te updaten.

3.1.3 NSIS [7]

Nullsoft Scriptable Install System is een opensource systeem waarmee Windows installers kunnen gemaakt worden. Een installer van NSIS zal dus ook enkel kunnen draaien op Windows. NSIS is script-based waardoor verschillende mogelijkheden ingebouwd kunnen worden in de installer. Een installer van NSIS kan ook uitgebreid worden met plug-ins waardoor de functionaliteit kan worden uitgebreid. Deze plug-ins kunnen geschreven worden in verschillende talen zoals C, C++, ... [6]. Het is mogelijk om in de installer verschillende componenten te includeren. Dankzij dit systeem kunnen we executables die al bestaan, blijven gebruiken. Of de installer de mogelijkheid bezit om een zip uit te pakken en de setup te doorlopen, zal nog onderzocht moeten worden.

Er moet nog onderzocht worden of NSIS de mogelijkheid aanbiedt om een updater te includeren in de gemaakte installer. Mocht dit niet het geval zijn, dan zal ook bij deze optie gezocht moeten worden om een updater te implementeren.

Naast NSIS is er ook nog Inno Setup. Dit is een zeer gelijkaardige tool en bezit gelijkaardige features [5].

3.1.4 Chocolatey [2]

Chocolatey is een Powershell execution engine dat gebruikt maakt van de NuGet packaging infrastructuur. Het systeem is te vergelijken met de package manager apt-get van Linux. Chocolatey zal alle executables, zips, ... die nodig zijn voor een stuk software encapsuleren zodanig dat alles in één package zit. Of de software de zips dan ook kan uitpakken en de setup starten is niet geweten uit de documentatie [1]. Chocolatey biedt mogelijkheden aan zodanig dat installatie, upgrades en verwijdering van het programma in de handen van de programmeur ligt. Er zal wel nog onderzocht moeten worden hoe het installatie en update systeem werkt. De nodige programma's moeten op de juiste locatie geplaatst worden om een goede werking te verzekeren. Het grootste nadeel van Chocolatey is het feit dat de gebruiker de Powershell moet gebruiken om de componenten up te daten en te installeren. Aangezien alles in de Powershell gebeurd moet het wel mogelijk zijn dat er batch files gemaakt worden. Deze zouden dan een

deel van het werk van de gebruiker kunnen wegnemen. Net zoals de vorige oplossingen is deze ook enkel toepasbaar voor Windows.

3.1.5 WinSparkle [12]

Aangezien voor sommige mogelijkheden, zoals NSIS en WiX, niet geweten is of het update systeem voldoende is, zal er gezocht moeten worden naar alternatieven om een updater te includeren in de installer. Een mogelijke oplossing is WinSparkle. WinSparkle is enkel bruikbaar voor Windows dus deze zou, als dit mogelijk is, in combinatie met NSIS of WiX gebruikt kunnen worden. Jammer genoeg biedt de documentatie van WinSparkle niet veel antwoorden aan [11]. Doordat er niet veel documentatie aanwezig is, kan de implementatie van WinSparkle in de installer lastig zijn. In de documentatie wordt ook het volgende vermeld [13]:

> Install update downloads the new version's installer and launches it.

Het is dan ook niet duidelijk of er voor ieder update dan een nieuwe installer moet geschreven worden. Mocht dit het geval zijn, dan is dit een minder aangewezen oplossing aangezien dit voor veel overbodig werk zorgt.

3.1.6 esky [3]

Nog een optie om een updater te implementeren is esky. Esky is een auto-update framework for frozen Python applicaties. Dankzij een eenvoudige API kunnen applicaties updates zoeken, ophalen en installeren. Er zit ook een mechanisme in dat de applicatie gaat beschermen tegen failed updates. Het is zeker mogelijk om de code van een programma up te daten met dit framework. Wat niet duidelijk is, is in hoeverre het mogelijk is om verschillende componenten up-te-daten. De updater zal ook de Python code omvormen tot een executable. Deze zijn jammer genoeg enkel bruikbaar op Windows.

3.1.7 Google Omaha [4]

Eén van de laatste opties om een updater te maken is Google Omaha. Het is de open-source versie van Google Update. Er is jammer genoeg niet veel te vinden over de mogelijkheden van Google Omaha en er zal dus moeten uitgezocht worden welke mogelijkheden deze aanbied.

Alle gevonden opties zijn op dit punt algemeen uitgelegd. De documentatie van verschillende opties was jammer genoeg niet altijd voldoende om een beeld te krijgen wat de optie precies allemaal inhoud. Daarom gaan bij de meesten opties een HelloWorld applicatie gemaakt worden. Met deze applicatie is het de bedoeling om te achterhalen hoe eenvoudig het is om een tool te gebruiken maar ook om meer duidelijkheid te creëren over de mogelijkheden van deze tool. We gaan tijdens het ontwikkelen van de HelloWorld applicaties ook zoeken naar tools waarmee we Windows executables kunnen draaien op Linux.

3.2 Testen

Zoals eerder vermeld gaan we nu enkele HelloWorld applicaties maken met de tools die we hebben besproken in sectie 3.1 op pagina 3. We gaan met deze tools een applicatie maken met als doel een executable in te pakken samen met een zip. De installer moet dan beide componenten correct installeren. Sommige tools bezitten de mogelijkheid om updates door te voeren. Tijdens deze testen zullen we ook kijken of het mogelijk is om één van de twee componenten up te daten met zo min mogelijk extra code. De code die gebruikt werd is terug te vinden in sectie 4.

4 Code

Listing 1: "Hello world programma met uninstaller"

```
# define installer name
OutFile "installer.exe"

# set desktop as install directory
InstallDir $DESKTOP

# default section start
Section

# define output path
SetOutPath $INSTDIR

# specify file to go in output path
File test.txt

# define uninstaller name
WriteUninstaller $INSTDIR\uninstaller.exe


#--------
# default section end
SectionEnd

# create a section to define what the uninstaller does.
# the section will always be named "Uninstall"
Section "Uninstall"

# Always delete uninstaller first
Delete $INSTDIR\uninstaller.exe

# now delete installed file
```

```
Delete $INSTDIR\test.txt

SectionEnd
```

Referenties

[1] Chocolatey Documentation. https://chocolatey.org/docs, 2016. [Online; geraadpleegd 3-08-2016].

[2] Chocolatey Main page. https://chocolatey.org/, 2016. [Online; geraadpleegd 3-08-2016].

[3] esky Main Page. https://pypi.python.org/pypi/esky, 2016. [Online; geraadpleegd 5-08-2016].

[4] Google Omaha Main Page. https://github.com/google/omaha, 2016. [Online; geraadpleegd 5-08-2016].

[5] Inno Setup Main Page. http://www.jrsoftware.org/isinfo.php, 2016. [Online; geraadpleegd 5-08-2016].

[6] NSIS Features. http://nsis.sourceforge.net/Features, 2016. [Online; geraadpleegd 2-08-2016].

[7] NSIS Main page. http://nsis.sourceforge.net/Main_Page, 2016. [Online; geraadpleegd 3-08-2016].

[8] QT Installer Framework Documentation. http://doc.qt.io/qtinstallerframework/, 2016. [Online; geraadpleegd 4-08-2016].

[9] QT Installer Framework Documentation Scripting. http://doc.qt.io/qtinstallerframework/scripting.html, 2016. [Online; geraadpleegd 5-08-2016].

[10] QT Installer Framework Overview. http://doc.qt.io/qtinstallerframework/ifw-overview.html, 2016. [Online; geraadpleegd 5-08-2016].

[11] WinSparkle Documentation. https://github.com/vslavik/winsparkle/wiki, 2016. [Online; geraadpleegd 5-08-2016].

[12] WinSparkle Main page. https://winsparkle.org/, 2016. [Online; geraadpleegd 5-08-2016].

[13] WinSparkle User-Experience. https://github.com/vslavik/winsparkle/wiki/User-Experience, 2016. [Online; geraadpleegd 5-08-2016].

[14] WiX Toolset Main Page. http://wixtoolset.org/, 2016. [Online; geraadpleegd 3-08-2016].

[15] WiX Toolset Merge Modules. https://www.firegiant.com/wix/tutorial/upgrades-and-modularization/mergers/, 2016. [Online; geraadpleegd 3-08-2016].