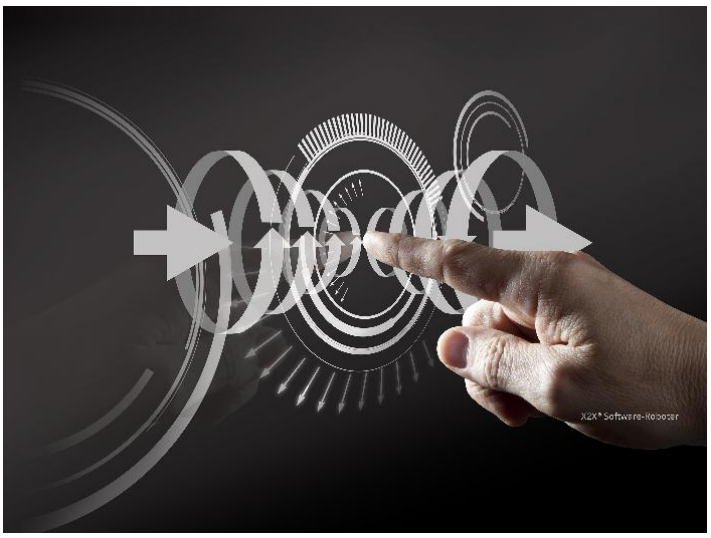
Software-Roboter Definition

Was ist eigentlich ein Software-Roboter und was ist er nicht? Nachfolgend wird eine Definition für „Software-Roboter“ festgelegt, die es erlaubt diesen von ähnlichen Begriffen abzugrenzen.



Entgegen einen normalen Roboter, der physikalisch existent ist, besteht ein "Software-Roboter" ausschließlich aus Software und kann wie folgt charakterisiert werden

"Software-Roboter" Definition

1. **Einmal gestartet führt ein Software-Roboter seine Aufgabe vollautomatisch (d.h. ohne das Zutun eines Menschen) durch, bis die Aufgabe erledigt ist.**
2. **Eins Software-Roboter ist zur Erledigung nicht nur einer Aufgabe, sondern einer ganzen Aufgaben-Klasse geeignet.**
3. **Ein Software-Roboter ist notwendiger Weise ein Interpreter, da er erst zur Laufzeit aus Instruktionsdateien erfährt, was er zu tun hat. (Ein Interpreter kann eine auf eine Aufgabenklasse reduzierte Skriptsprache interpretieren.)**

Beispiel

Der X2X Software-Roboter arbeitet auf dem Dateisystem eines Computers und kann alle Aufgaben lösen die darin bestehen Dateien einzulesen, daraus bestimmte Daten zu extrahieren und diese unmittelbar oder kombiniert mit andern Daten als neue Dateiinhalte bzw. neue Dateien abzulegen.

Seine konkrete Aufgabe wird in Instruktionsdateien in der Skriptsprache X2X beschrieben.

Der X2X ®Software-Roboter besteht aus einer Software-Applikation (Computerprogram) die herkömmlich programmiert, in ausführbarem Maschinencode übersetzt, in den Computer geladen

und dort zur Ausführung gebracht wird und mindestens einer Instruktionsdatei die die Software-Applikation als ersten Schritt lesen muss.

Ausgehend von der ersten (MAIN-) Instruktionsdatei liest die Software-Applikation eventuell weitere dort genannte Instruktionsdateien ein, bis alle Voraussetzungen bekannt sind, um die spezifische Aufgabe erledigen zu können. Dann werden diese ausgeführt.

Abgrenzung des Begriffs-„Software-Roboter“ gegen ähnliche Begriffe

Grundsätzlich kann jede Aufgabe die ein Software-Roboter erledigen kann, (bzw. die mittels Instruktionsdateien beschrieben werden kann,) auch mit einer allgemeinen Programmiersprache als Sourcecode beschrieben und mittels Compiler in ausführbaren Maschinencode übersetzt werden. Dieser Maschinencode würde dann genau das tun, was der Software-Roboter tun würde, wenn er diesen spezifischen Instruktionsdateien instruiert wird. Aber der compilierte Maschinencode ist unveränderbar und kann immer nur dieselbe Aufgabe erledigen und repräsentiert somit keinen Software-Roboter. (Im Gegensatz zu einer Programmiersprache, die ein Compiler in Maschinencode übersetzen kann, wird eine Skriptsprache zur Laufzeit interpretiert. D.h. die hinterlegten Skriptsprachenbefehle werden auf einen fest vorgegebenen Satz von vorimplementierten Funktionen abgebildet.)

Das gleiche gilt für die im Englischen als "**software robots**" oder kurz "**Bots**" bezeichneten Computerprogramme die zwar vollautomatisch ausgeführt werden, aber keine "Software-Roboter" im Sinne obiger Definition sind, weil sie nur Punkt 1. erfüllen. (Im Übrigen erwartet man von diesen "Bots", dass sie in praxistauglicher Zeit riesige Datenmengen verarbeiten können und deshalb unabdingbar in Maschinecode ausgeführt werden müssen. Typisches Beispiel sind Internetsuchmaschinen. Leider wurde der englische Begriff "software robots" in der Vergangenheit manchmal mit "Softwareroboter" übersetzt. In neuerer Zeit wird dafür sowohl im Englischen wie Deutschen der Begriff „**Software-Agent**“ verwendet.)

Der Vorteil des compilierten Maschinencodes ist der, dass er in der Ausführung deutlich schneller ist als der Software-Roboter, weil dieser die Instruktionsdateien zur Laufzeit erst interpretieren muss. Weil ein Software-Roboter aber für die Lösung einer ganzen Klasse von Aufgaben entwickelt wird, ist es für den Anwender deutlich einfacher eine konkrete Aufgabe aus dieser Klasse in Instruktionsdateien zu spezifizieren, als ein neues Computerprogramm von Grund auf zu entwickeln, das dann gerade nur eine spezifische Aufgabe erledigen kann.

Denkbar und realistisch ist es auch, ein Computerprogramm zu schreiben das die in den Instruktionsdateien verwendete Skriptsprache lesen und direkt in Maschinencode übersetzen kann. Ein solches Computerprogramm würde einem normalen Compiler/Übersetzer entsprechen. Ein solcher Compiler ist aber kein Software-Roboter, weil er keine Instruktionsdateien interpretiert, sondern die ihm übergebene (Sourcecode-) Dateien verarbeitet. Die Anweisungen wie er das zu machen hat sind "hart-codiert" in seinem Maschinencode hinterlegt.

Umgekehrt kann man zu jeder existierenden allgemeinen Programmiersprache einen Interpreter schreiben. Ein solcher Interpreter wäre nach obiger Definition ein Software-Roboter. - Allerdings ein sinnloser, denn er würde den Programmcode nur langsamer ausführen als der von einem Compiler erzeugte Maschinencode. Der Anwender hätte keinerlei Vorteil von solch einem Interpreter.

Der wesentliche Unterschied eines Software-Roboters zu einem Interpreter für eine **Programmierhochsprache** ist der, dass der Software-Roboter auf eine Aufgabenklasse spezialisiert bzw. beschränkt ist. (Siehe 2. aus obiger Definition) Dies ermöglicht es, die in den Instruktionsdateien verwendete Skriptsprache, entsprechend der Aufgabenklasse, optimal einfach zu gestalten und den Programmcode, der der gesamten Aufgabenklasse entspricht, als Bestandteil im Maschinen- bzw. Applikationscode des Software-Roboter zu hinterlegen. Als Folge davon, können Software-Roboter viel einfacher und schneller zur Ausführung einer bestimmten Aufgabe instruiert werden, als es möglich wäre ein gleichwertiges Computerprogramm zu entwickeln. (Praktische Erfahrungen zeigen einen Faktor 10 bis 100, mit entsprechender Kosteneinsparung.)

Bezogen auf das Definieren von Programmiersprachen für einen bestimmten Problemkreis sind sogenannte "**Domänen Spezifische Sprachen**" (DSL) im selben Lichte zu betrachten. DSL sind Programmiersprachen, die auf das effiziente Umsetzen von Lösungen für einen bestimmten Problemkreis vorgesehen sind. In der Literatur finden sich aber nur DSL's die mittels eines spezifischen Compilers in Maschinencode übersetzt werden. Entsprechende Interpreter, die Voraussetzung für einen Software-Roboter wären, lassen sich nicht finden.

DSL erfordern eine sehr aufwendige Spezifikation um den notwendigen Compiler erstellen zu können, so dass dieses Verfahren, obwohl seit ca. 20 Jahren bekannt, in der Praxis nur sehr selten erfolgreich umgesetzt werden konnte und sich deshalb nur in besonderen Einzelfällen durchgesetzt hat. (Weder ein Compiler für eine DSL, noch das von so einem Compiler erzeugte, ausführbare Programm repräsentieren einen Software-Roboter im Sinne obiger Definition.)

Besonderheiten des X2X Software-Roboters

Der X2X Software-Roboter ist insofern universal, als die Aufgabenklasse des Software-Roboters gerade die sein kann, dass er Dateien mit dem Inhalt einer anderen Skriptsprache in Dateien mit dem Inhalt seiner eigenen Skriptsprache umzuwandeln kann, um diese in einem nachfolgenden Arbeitsgang als Instruktionsdateien zu verwenden. (Er kann z.B. vom Anwender "ausgefüllte" formularähnliche Schablonen in eigene Instruktionsdateien umwandeln.)

Betrachtet man eine Klasse von Skriptsprachen die ineinander umgewandelt werden kann, so kann der X2X Software-Roboter so instruiert werden, dass er beliebige Skripte der einen Skriptsprache aus dieser Klasse in eine andere Skriptsprache dieser Klasse übersetzen kann. Dies gilt insbesondere für die Klasse der Skriptsprachen, die seine "eigene" Skriptsprache enthält.

Weiterführende Dokumente:

http://www.sss.de/x2x-downloads?file=files/triple-s/downloads/downloads_X2X_2015/Sourcecode_generieren.pdf

http://www.sss.de/x2x-downloads?file=files/triple-s/downloads/downloads_X2X_2015/Modellbasierte%20Softwareentwicklung.pdf

http://x2x.sss.de