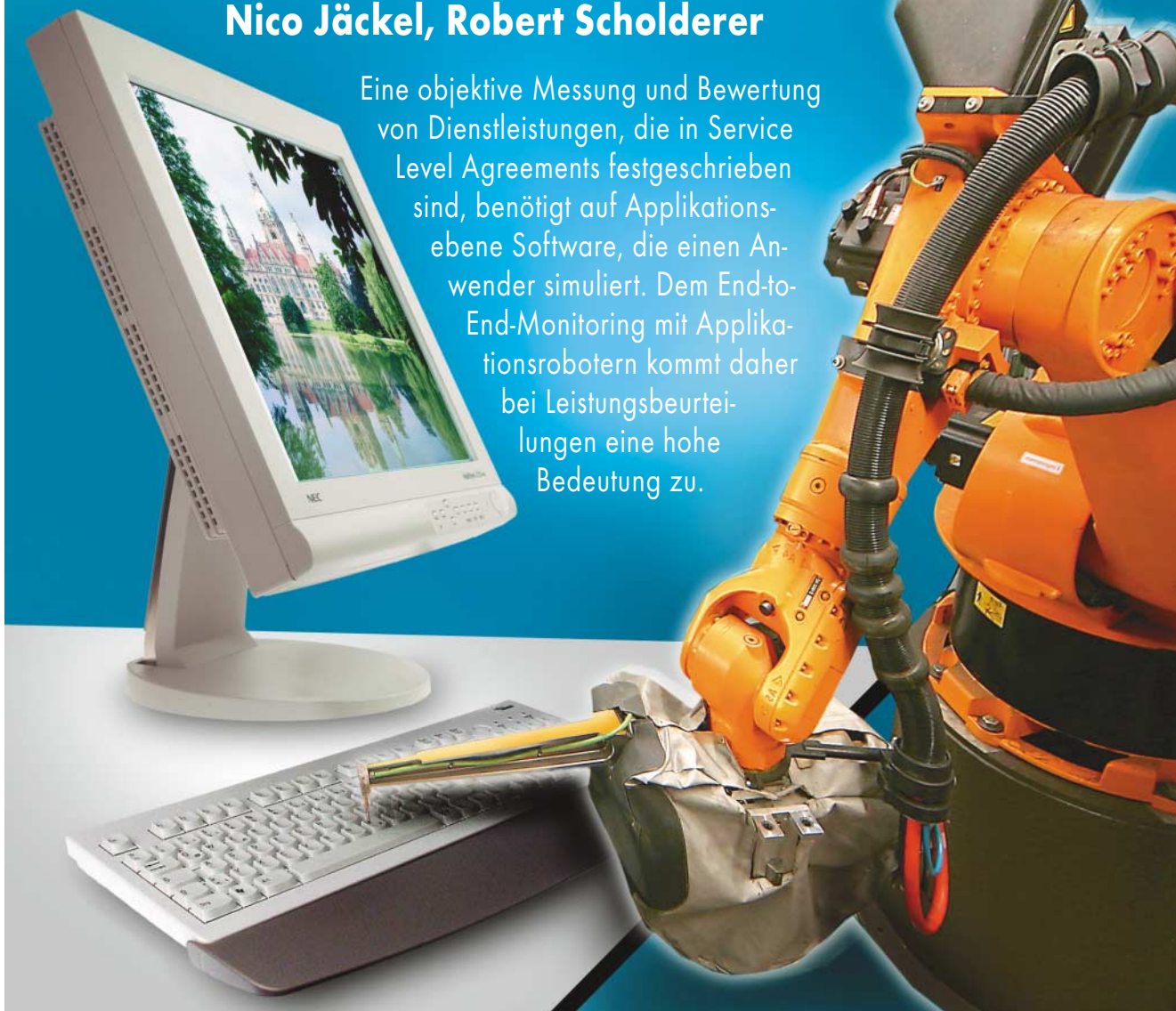


End-to-End-Monitoring mit Applikationsrobotern

Gläserner Dienst

Nico Jäckel, Robert Scholderer

Eine objektive Messung und Bewertung von Dienstleistungen, die in Service Level Agreements festgeschrieben sind, benötigt auf Applikations-ebene Software, die einen Anwender simuliert. Dem End-to-End-Monitoring mit Applikationsrobotern kommt daher bei Leistungsbeurteilungen eine hohe Bedeutung zu.



Von einem IT-Provider erwarten die Kunden, dass er die erbrachten Leistungen transparent nachweist. Der Service-Nehmer will genau wissen, wofür er bezahlt. Ein Weg, den Beweis für getane Arbeit anzutreten, ist das „End-to-End-Monitoring“ (E2E-Monitoring) mit Applikationsrobotern, das das Antwortverhalten von Anwendungen automatisiert misst.

Objektive Messungen bedeuten eine enorme Umstellung für IT-Anbieter. Ohne sie lässt sich die Gunst der Kunden lediglich mit vagen Reports

zu Qualität und Leistung erhalten, die wenig Rückschlüsse auf die tatsächliche Einhaltung des Service-Levels zulassen und rasch in technische und nicht zuletzt rechtliche Grauzonen führen.

Damit ein Dienstanbieter den berechtigten Forderungen nach mehr Transparenz nachkommen kann, muss er rechtlich präzise Verträge mit technischer Detailtiefe abschließen. Ein derartiges Qualitätsversprechen schwarz auf weiß bietet durchaus Vorteile für den Provider: Indem er den gut dokumentierten

Forderungen nachkommt, bleibt er konkurrenzfähig.

Das unbedachte Einrichten eines Leistungsreportings kann sich jedoch als selbst gelegte Stolperfalle erweisen. Verwenden IT-Dienstleister simple Basisstatistiken, die unter anderem grobe Angaben zur Systemverfügbarkeit enthalten, geben sie dem Kunden einen langen Hebel in die Hand. Denn der kann behaupten, dass ihm die versprochene Leistung nicht zur Verfügung stand. Das bringt neben unnötigen Belastungen der Geschäftsbeziehung allzu

IX-TRACT

- Die Leistung, die ein Service-Provider auf seinen Systemen zur Verfügung stellt, kommt nicht immer unbeeinträchtigt beim Kunden an.
- Streitigkeiten um die tatsächlich abgelieferten Leistungen können nicht nur das Verhältnis zum Kunden belasten, sondern auch den Gewinn schmälern.
- Für objektive Messungen aus Anwendersicht sind spezielle Werkzeuge erforderlich, die so genannten Applikationsroboter.

oft eine Minderung der Zahlung mit sich. Daran lässt sich nicht rütteln, wenn der Betrag in den Dienstleistungsvereinbarungen (Service Level Agreements, SLA) mit dem Einverständnis beider Parteien definiert ist.

Gegen derlei Ungemach sollte sich ein Provider wappnen, indem er vor allem den Spielraum für Diskussionen darüber einengt, welche Dienstqualität auf der Kundenseite tatsächlich ankam. Als Mittel der Wahl gilt eine Statistik zur Applikations- und Prozessverfügbarkeit mit dem Fokus End-to-End. Das Prinzip erscheint auf den ersten Blick bestechend einfach: Der Geschäftsprozess des Kunden lässt sich simulieren und in der operativen Umgebung messen. End-to-End-Monitoring ermöglicht es Unternehmen, Anwendungen aus derselben Perspektive wie der Endnutzer zu betrachten. Es kann bestehende Monitoring-Lösungen (etwa von Anbietern wie HP und IBM) für die IT-Infrastruktur ergänzen.

Ein solches Werkzeug finden viele Anbieter attraktiv, wie die Marktforscher von Gartner bei einer kürzlich

veröffentlichten Umfrage herausfinden. Fast 50 Prozent der zur Data Center Conference 2004 Befragten gaben an, Investitionen in ein aktives End-to-End-Monitoring zu planen und vorrangig für das interne Performance-Monitoring nutzen zu wollen.

Anwender-Simulation

Viele aktive Messmethoden arbeiten jedoch im Hintergrund und führen Benutzeraktionen aus, ohne dass der Anwender Einblick in das Geschehen hat. Deshalb steigt der Bedarf an Messungen, die für den Service-Nehmer transparent sind. Zum Einsatz kommen dafür insbesondere Applikationsroboter, also Computerprogramme, die weitgehend selbständig und ohne Benutzerinteraktion arbeiten. Der Begriff des Roboters entstammt der Automobilindustrie, in der ganze Produktionsabschnitte automatisiert ablaufen.

Ein anschauliches Beispiel für automatisierte Abläufe in der IT bietet das Customer Relationship Management (CRM). Der User navigiert stets mit einem Ziel vor Augen – nach einem bestimmten Modell – durch eine Applikation. Die Klickfolge kann wie folgt aussehen: Er startet die CRM-Applikation, meldet sich an, holt die Liste der aktuellen Kunden, bearbeitet sie, meldet sich ab und schließt den Vorgang. An dieser Stelle kommt der Applikationsroboter ins Spiel: Er hat die Aufgabe, den Ablauf getreu den Vorgaben nachzustellen.

Für einen Betrachter sieht es aus, als werde der PC von Geisterhand bedient. Ohne menschliches Zutun öffnet sich die Applikation auf dem Desktop, es werden Texte getippt und Buttons angeklickt. Mit dieser Methode lassen sich ganze Geschäftsprozesse umsetzen – und zugleich Qualitätsparameter

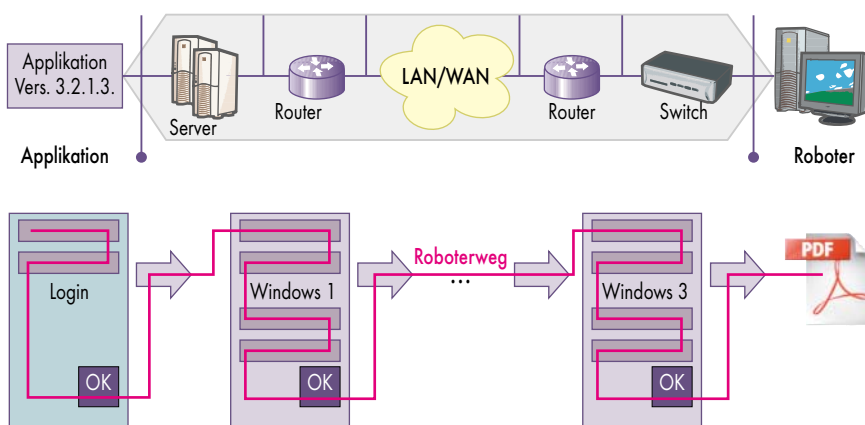
wie Verfügbarkeit und Antwortzeit messen.

Bei der Realisierung gilt es jedoch mehr Details zu beachten, als es für den verblüfften Zuschauer den Anschein hat. Der Monitoring-Roboter sollte zum Beispiel nicht an die Programmierschnittstellen der jeweiligen Anwendungen andocken (Application Programming Interface, API), denn das würde reichlich Spezialwissen und passende Programmierkenntnisse erfordern. Die einzige von Anwendungen unabhängige Konstante bietet die API des Desktops. Für ihren Einsatz spricht, dass sich Applikationen und die dazugehörigen Elemente bis auf wenige Ausnahmen direkt auf dem Desktop registrieren. Für eine Steuerung genügt demnach das Wissen über den Umgang mit dem Desktop.

Einsatzfelder der Automatisierung

Ein simples manuelles „End-to-End-Monitoring“ kann bereits aus der schlichten wie wirksamen Eingabe des Befehls *ping* auf dem Prompt oder aus dem Senden und Abrufen einer Testmail bestehen – eine probate Vorgehensweise und der erste Schritt zu einem automatisierten End-to-End-Monitoring. Auf die Dauer ist das händische Abfragen unpraktikabel, denn schließlich handelt es sich um einen neuralgischen Punkt im Geschäftsbetrieb. Bietet der Anbieter einen Fullservice, indem er seinen Kunden zum Beispiel komplett ausgestattete Arbeitsplätze mit Software zur Verfügung stellt, steht er fortwährend in der Verantwortung, dass die Applikationen richtig konfiguriert sind.

Roboter können laufend nachweisen, ob und in welcher Qualität die IT-Dienste zur Verfügung stehen. Zu den



Zwischen dem Server des Providers und der Anwendung des Kunden liegt häufig ein weiter Weg. Objektive Messungen sind ohne unabhängige Instanz nur auf Seiten des Service-Nehmers möglich (Abb. 1).

Der Roboter, ein simulierter Anwender, bewegt sich anhand fest programmierter Vorgaben durch die Applikation und dokumentiert am Ende die Ergebnisse (Abb. 2).

Anmeldevorgang bei SAP R/3

Kommando	Erläuterung
UseWindow("SAPLOGON.EXE.. .. Logon ",1) Click(Button,"&Anmelden")	Setzen des Fokus auf das Anmeldefenster Der Anmeldebutton wird gedrückt
UseWindow("... SAP R/3 1",1) SendKeys("login<Tab>password<Enter>")	Setzen des Fokus auf den Login Dialog Eintragen der Benutzerdaten und Enter drücken

Stärken der Applikationsroboter zählt darüber hinaus die Messung eines gesamten Ablaufes mitsamt aller Teilschritte. Welche Dienste auch immer das SLA festlegt, ein Roboter kann sie nachbilden.

Stärken und Schwächen

Der Service-Nehmer kann die Messungen direkt mitverfolgen und Applikationen über den Applikationsroboter remote prüfen lassen. Deshalb kommt das End-to-End-Monitoring vor allem im Outsourcing zum Einsatz. Die Messung kann sich präzise am Service Level Agreement orientieren. Beide Parteien müssen sich lediglich über eine Klickstrecke einig sein. Bestimmt der Provider innerhalb der Klickstrecke zudem bestimmte Zeiten exakt, sind auch die Messpunkte für den Kunden transparent.

Bei der Integration eines End-to-End-Monitorings sind aber auch Hindernisse zu überwinden. Besonders die Sicherheitsaspekte gilt es genauestens zu diskutieren, denn für das Monitoring befindet sich ein ungeschützter Mess-PC im Unternehmen. Wenn Roboter aktiv die Benutzeroberfläche „betreten“, sollte der Rechner nicht gesperrt sein. Den einzigen Schutz bildet die Zutrittskontrolle oder das Abschließen von Hardwareports. Damit der Rechner nicht über die Remote-Konsole angegriffen wird, ist die Beschränkung auf eine dedizierte Umgebung notwendig.

Außer den Sicherheitsvorkehrungen gilt beim End-to-End Monitoring der laufenden Wartung das Hauptaugenmerk. Wenn Applikationen häufigen Änderungen unterworfen sind, zeigen Roboter Schwächen – Roboter, sie sich dynamisch anpassen, sind heute noch nicht in Sicht. Bei jeder Änderung bedarf es der Prüfung, ob der Roboter seine Wege noch findet oder ob eine Neukonzeption des Roboters ansteht.

Der Start einer Applikation kann häufig einfach über ein auf der Shell abgesetztes Kommando erfolgen.

Kommt eine grafische Oberfläche (GUI) ins Spiel, etwa der Windows-Desktop, wird es komplizierter: Damit der Fokus auf dem richtigen Fenster liegt, muss über das GUI der Name des Fensters ausgelesen werden. Simulierte Tastatureingaben gehören ebenfalls zur Roboter-Grundausstattung, damit er Texte, Namen, Logins et cetera auf die Applikation übertragen kann. Um grafisch zu navigieren, wie dies etwa bei Citrix-Systemen der Fall ist, müssen bestimmte Bereiche auf dem Desktop markiert und als Bilder gespeichert werden.

On Air: die Aufzeichnung

Das Aufzeichnen einer Interaktion mit der Anwendung ist die zentrale Funktion, die Roboter für ein aussagekräftiges End-to-End-Monitoring benötigen. Es muss schnell und präzise erfolgen, denn die Qualität der Aufzeichnung beeinflusst den Projekterfolg und somit die Wartungskosten. Ein gutes Projektergebnis stellt sich nur ein, wenn das Projekt eindeutig definiert starten kann.

Keine leichte Aufgabe, schließlich muss für die präzise Aufzeichnung der Benutzung entweder eine gute Vorlage oder ein Experte zur Hand sein. Beim Einsatz einer Vorlage müssen die jeweiligen Abläufe klar nachvollziehbar sein. Dazu gehört die Erstellung einer kompletten Liste der Benutzereingaben.

Das Recording sollte zudem Abweichungen durch Mausclicks in unterschiedliche Bereiche der Bildschirmoberfläche sicher beherrschen können.

Vom Mitschnitt zum Foto-Shooting

Während der Aufzeichnung ist zunächst im Hintergrund die Klickstrecke zu kodieren, und der Code sollte zur nachträglichen freien Bearbeitung zur Verfügung stehen („Mitschnitt“). Unterbrechungen im Ablauf („Stop-and-Go“) sind zu berücksichtigen, damit sich etwa Bedienungsfehler beheben lassen. Es sollte möglich sein, die Aufzeichnung an jeder beliebigen Stelle zu unterbrechen und wieder zu starten. Über die Eingabefunktionen hinaus hat die Verarbeitung der Grafik-Ausgabe Bedeutung („Foto-Shooting“), wenn zum Ablauf die Referenzierung von Bildern gehört. Die Funktion „Image Synchronisation“ hilft weiter, wenn zum Beispiel einem für den Ablauf wichtigen Button unter Windows kein Name zugewiesen wurde.

Wartung und Wiederverwendbarkeit

Applikationen ändern sich bekanntlich von Zeit zu Zeit, was laufende Anpassungen des Applikationsroboters nach sich zieht. Wenn sich Aufzeichnungen von Abläufen nicht zu Teilen austauschen und ergänzen lassen oder wenn der ganze Ablauf etwa komplett neu eingespielt werden muss, geht der wesentliche Vorteil der Applikationsroboter, ihre Wiederverwendbarkeit, verloren – und die Projektkosten steigen.

Das Arbeiten mit Bildern verschärft die Situation. Den Einsatz gra-

Anmeldevorgang bei Citrix Metaframe

Kommando	Erläuterung
UseWindow("PN.exe...ICA-Verbindungen",1) Chooseltem(ListView, "1", "ICA-Verbindung ..", single, left) Chooseltem(ListView, "1", "Server-Farm", double, left)	Setzen des Fokus auf das Citrixfenster Navigation zur ICA-Verbindung und Servercluster, dann Icon wählen
PAUSE until Bitmap("C:\... \Scripts\anmeldedialog.bmp") InWindow("WFICA32.EXE ... XP-Farm - Citrix ICA-Client 1",1) InArea(0, 0, 1280, 1024) EndPause	Start einer Warteschleife, bis das Bild für den Anmeldedialog erscheint. Setzen des Fokus Suchen des Anmeldedialogs auf Bildschirm Ende der Schleife
UseWindow("WFICA32.EXE - Citrix ICA-Client 1",1) SendKeys("login<Tab>password<Tab><Tab><Enter>")	Setzen des Fokus auf das Anmeldefenster Eintragen der Benutzerdaten und Enter drücken

fischer Elemente sollte man tunlichst vermeiden, weil dadurch die Wartungskosten stark ansteigen. Schwierigkeiten und Unsicherheiten ergeben sich zum Beispiel, wenn der Roboter auf mehreren Systemen mit unterschiedlichen Bildschirmauflösungen läuft. Schließlich ist eine Skalierung von Bildern derzeit nicht möglich. Doch oft geht es nicht ohne Bilder, denn nicht alle Events und Benutzerelemente sind auf dem Desktop registriert. Als Lösung bieten sich derzeit nur zwei konträre Ansätze an: im Code integrierter oder im Code referenzierter Grafikkarten-Inhalt.

Beim integrierten Inhalt wird das Bild von der Grafikkarte gelesen und direkt im laufenden Code eingespielt, was ihn unübersichtlich macht. Dies ist insbesondere beim Verteilen von Robotern auf viele Zielsysteme zu beachten. Eine kompakte Umsetzung garantiert nur die Übertragung einer Datei, die ohne Laufzeitumgebung direkt auf dem Zielsystem starten kann.

Anbieter von E2E-Monitoring-Werkzeugen

Auf dem Markt gibt es Tools diverser Ausprägung. Häufig sind sie nicht in erster Linie für das End-to-End-Monitoring, sondern vor allem für andere Aufgaben konzipiert. Eine kleine Auswahl an Tools, die sich für den Einsatz von desktoporientierten Applikationsrobotern für das End-2-End-Monitoring eignen:

BMCs Magic Desktop Automation erlaubt eine umfangreiche Batchverarbeitung, die auch über die Oberfläche laufen kann (www.remedy.com).

Robotask erlaubt ein interaktives Zusammenstellen komplexer Clickstreams, deren Code auch Unerfahrene über Funktionstasten erweitern können (www.robotask.com).

Wintask bietet viele Extras für komfortables Ende-zu-Ende Monitoring (www.wintask.com).

Winbatch bietet mit der „Windows Desktop Automation“ eine einfache Möglichkeit für die Erstellung von Applikationsrobotern (www.winbatch.com).

Mjtnet automatisiert Geschäftsprozesse, eine Nachbearbeitung der Abläufe erfolgt per Editor (www.mjtnet.com).

Die genannten Tools haben keine explizite Integrationsschnittstelle für SLA-Werkzeuge. Eine Einbindung ist jedoch für den Administrator oder Service Level Manager technisch möglich.

Referenziert der Code den Grafikkarten-Inhalt, muss er nur auf Bilder vom Desktop und dessen Elemente verweisen, und bleibt dadurch übersichtlich. Beim Rollout sind der Roboter und seine Dateien komplett auf das Zielsystem zu überspielen – bei einer identischen Abbildung der Pfadstruktur.

Das End-to-End-Monitoring richtet sich an der zu messenden Software aus, und bei der handelt es sich um ein „lebendes Objekt“. Der Nutzer muss sich deshalb auf laufende Weiterentwicklungen, Updates und Benutzerprofile einstellen. Besonders bedeutsam ist ein lesbare Code, damit sich manuelle Anpassungen zügig durchführen lassen. Der Experte muss den Quelltext also wirklich lesen können und im Notfall Hand anlegen können.

Keine Alternative gibt es zum fortwährenden Abstimmen individueller Abläufe und zur Behandlung von Fehlern, die im laufenden Betrieb auftreten. Als Beispiel möge ein in der Praxis immer wieder auftretendes Szenario dienen: die Aufzeichnung einer neuen, dem Anwender unbekanntem Applikation. An erster Stelle steht der Test der Login-Funktion unterschiedlicher Benutzer unter Zuhilfenahme von Applikationsrobotern.

Der Aufzeichner statet die Applikation, führt den Login für einen Test-User durch und schließt die Applikation nach getaner Arbeit wieder. Ferner muss der Nutzer den Code erweitern, indem er eine Datei mit den Logindaten bereitstellt, damit der Roboter bei jedem neuen Start der Applikation einen anderen Anwender simuliert. Es handelt sich jedes Mal um die gleiche Prozedur – und dennoch können unterschiedliche Effekte auftreten. Zum Beispiel ist der Benutzername plötzlich nicht mehr bekannt, das Passwort falsch oder das passende Profil lässt sich einfach nicht mehr laden.

Monitoring in der Praxis

Bereits diese Standardaufgabe verdeutlicht, wie komplex die individuelle Prüfung einer Applikation geraten kann, wenn sämtliche Eventualitäten einfließen sollen. Verändert sich außer den Daten auch die Navigation, wie es oft bei Webseiten vorkommt, die ihren Inhalt profilabhängig ausrichten, ist die Verwirrung des Applikationsroboters perfekt. Er benötigt eine aufwendige Wartung und laufende Funktionsüberwachung.

Zwei Praxisbeispiele verdeutlichen die Bearbeitungsschritte, die für das Login von SAP R/3 und von Citrix Metaframe erforderlich sind. Für beide dienen Fragmente aus Virtual-Basic-Code zur Veranschaulichung. Das Beispiel im ersten Kasten zeigt einen Aufruf von SAP R/3 samt der Anmeldung. Der Applikationsroboter navigiert über den Namen der Fenster und trägt dort die Login-Daten ein.

Im zweiten Beispiel – Citrix Metaframe – verhält es sich prinzipiell anders. Citrix verwendet die in der Großrechnerwelt bekannte Host-Technik, sendet also einen Text und erhält vom Host das Ergebnis – in diesem Fall ein Bild für die Desktop-Emulation (etwa mit einem Dialog). Der Applikationsroboter muss also Eingaben senden und auf das entsprechende Bild warten.

Ausblick: Parallelisierung und KI

Dank des steigenden Monitoring-Bedarfs im Outsourcing treiben diverse Anbieter diese Disziplin voran. In den Bereichen Parallelisierung und KI sind deutliche Veränderungen zu erwarten. Die heute noch sequenzielle Verarbeitung auf dem Desktop kann zukünftig durch eine Parallelisierung abgelöst werden. Mehrere Applikationsroboter arbeiten dann gleichzeitig auf virtuellen Desktops. Der Hardware-Aufwand selbst bei intensiven Lasttests dürfte sich daher in Grenzen halten.

Eine semantische Auswertung für eine Steigerung der Roboter-Intelligenz steht zwar noch am Anfang, aber der Einsatz der KI birgt wirtschaftliches Potenzial. Die Flexibilität der Roboter muss im gleichen Maß zunehmen wie die Komplexität der Dienstleistungsvereinbarungen zwischen Providern und Service-Nehmern. Das Ziel ist die Abbildung beliebig komplexer Service Level Agreements. Kein aussichtsloses Unterfangen, denn Applikationsroboter werden immer schneller und intelligenter. (un)

DR.-ING. ROBERT SCHOLDERER

(Leiter Technik) und

NICO JÄCKEL M. A.

(Leiter Marketing und Vertrieb)
sind Geschäftsführer der Karlsruher
G-NE GmbH.

