Softwareentwicklung ohne Maus und Tastatur

Sprechen ist das neue Klicken

Dr. Wolfram Wingerath, Michaela Gebauer

Für die Bedienung des Computers brauchte man viele Jahre Maus und Tastatur – heute kann man mit Sprache, Gestik und Mimik sogar programmieren.

Obwohl Textnachrichten am Smartphone mittlerweile häufig diktiert und nicht mehr getippt werden, sind vergleichbare Bedienkonzepte noch nicht im Alltag von PC-Nutzern angekommen. Interfaces jenseits von Maus und Tastatur werden als Spielerei belächelt oder gar nicht erst wahrgenommen. „Völlig zu Unrecht", wie der Informatiker und Junior-Fellow der Gesellschaft für Informatik Dr. Wolfram „Wolle" Wingerath findet: „Handsfree Coding geht weit über das Diktieren von Texten hinaus und ermöglicht auch professionellen Anwendern die Nutzung des Computers ganz ohne Einsatz ihrer Hände."

Wolle ist 33 Jahre alt, Data Engineer und erprobt seit mehr als zehn Jahren Eingabemethoden zur Softwareentwicklung ohne Maus und Tastatur. Inzwischen setzt er fast ausschließlich auf Handsfree Coding, da er damit effizienter arbeitet. „Dadurch muss ich mir keine kryptischen Shortcuts mehr merken und kann ganz bequem mit Sprache, Geräuschen, Mimik oder Gestik den Computer und die Programme steuern", sagt er.

Beim Handsfree Coding spielt das Voice Coding eine zentrale Rolle. Hierbei wird Quellcode per Spracheingabe erstellt. Voice Coding ist jedoch nicht mit handelsüblicher Software zur automatischen Spracherkennung (Automatic Speech Recognition, ASR) vergleichbar. Es gibt zwar einige offensichtliche Parallelen zum Diktieren von Textnachrichten. Mit Standardsoftware zur Spracherkennung kann man aber nicht ohne Weiteres effizient programmieren, da ASR auf die Interpretation und Synthese einer konkreten natürlichen Sprache ausgelegt ist. Sie verwendet dafür jeweils spezifische Modelle, Grammatiken und Optimierungen bei der Ausgabe, etwa, wenn sie automatisch Satzzeichen einfügt oder Substantive großschreibt. Bei typischer ASR-Software sind Befehle stets mit einem Schlüsselwort einzuleiten und durch Sprechpausen abzuschließen. Während sich so einfache Tastenaktionen umsetzen lassen – etwa mit der Aussage „press Enter" zum Drücken der Eingabetaste –, ist die Ausführung von komplexen Aktionen oder Aktionssequenzen eher beschwerlich und ineffizient.

Programmieren mit Einsatz von Sprache

„Als ich vor etwa zehn Jahren meine Masterarbeit geschrieben habe, habe ich den gesamten Text diktiert", erinnert sich Wolle. Zuvor hatte er in der Uni ein theoretisches Seminar über Spracherkennungssoftware belegt und sich anschließend dazu entschieden, es privat auszuprobieren. „Ich erinnere mich noch daran, wie begeistert ich darüber war, dass sie zum Diktieren wirklich funktioniert hat", sagt er lachend.

Zum Programmieren eignet sich Standard-ASR-Software jedoch kaum, weshalb viele Entwickler spezialisierte Software zum Voice Coding verwenden. Diese setzt zwar auf traditioneller ASR-Software auf, ergänzt sie jedoch um Scripting-Frameworks für benutzerdefinierte und erweiterbare Befehlsgrammatiken, in denen Sprachbefehle definiert und mit Aktionen

X-TRACT

- Mit Voice Coding lässt sich Spracheingabe auch beim Programmieren nutzen.
- Software zur automatischen Spracherkennung eignet sich nicht zum Voice Coding, da sie auf die Interpretation und Synthese einer natürlichen Sprache ausgelegt ist.
- Statt kryptische Shortcuts zu nutzen, kann man mit semantisch bedeutsamen Begriffen Anwendungen steuern.
- Die Kombination aus Voice Coding, Eyetracking und Noise Recognition macht Handsfree Coding zu einer effizienten Alternative zu Maus und Tastatur.

verknüpft werden können. Die meisten Frameworks nutzen dabei Python als Skriptsprache und Englisch als Sprechsprache, da der größte Teil der Voice-Coding-Community Englisch spricht.

In den vergangenen zehn Jahren hat sich das Voice-Coding-Set-up von Wolle immer wieder verändert und weiterentwickelt. Sein aktuelles ist in Abbildung 2 zu sehen. Den Kern bildet das Scripting-Framework Talon. Auf dessen Basis baut die für das Voice Coding elementare Befehlsgrammatik auf, in der Sprachbefehle definiert und mit Aktionen verknüpft werden können. Talon kann in der kostenfreien Basisversion zum Beispiel mit der ASR-Engine Wav2-Letter genutzt werden. Sie steht für Linux, macOS und Windows zur Verfügung. In der kostenpflichtigen Beta erhält man Zugriff auf in Entwicklung befindliche Features wie die Conformer-Engine, die eine höhere Spracherkennungsgenauigkeit als Wav2Letter bietet.

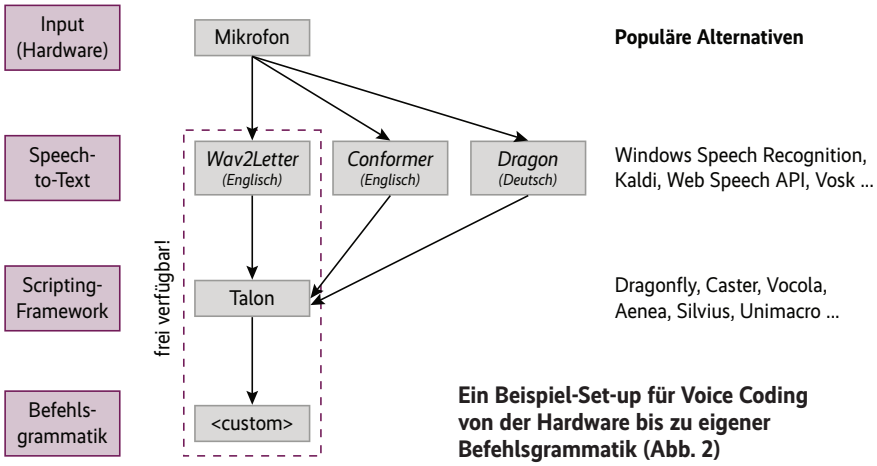
Dr. Wolfram „Wolle“ Wingerath nutzt seit zehn Jahren Handsfree Coding (Abb. 1).

Zur Softwareentwicklung und zum Verfassen von Texten auf Englisch verwendet Wolle die Conformer-Engine. „Zusätzlich habe ich die deutsche Version der ASR-Software Dragon in mein Set-up integriert, sodass ich bei Bedarf auch zur deutschen Sprache wechseln kann“, sagt er. Dazu hat er eine modifizierte Version einer Talon-Grammatik implementiert, die eigentlich für ein englisches Sprachprofil ausgelegt ist. „Auch wenn ich die meisten Befehle mit meinem deutschen Sprachprofil aufrufen kann, ist die Trefferquote hier niedriger als bei meinem englischen Sprachprofil. Daher ist Englisch auch meine Arbeitssprache und ich nutze die deutsche ASR-Engine praktisch nur zum Diktieren von Texten auf Deutsch“, erklärt er.

Buchstabieren auf andere Art

Ein zentraler Aspekt beim Voice Coding ist das Diktieren einzelner Buchstaben. Da jedoch viele Buchstaben sehr ähnlich klingen und daher leicht zu verwechseln sind, verwendet man üblicherweise ein phonetisches Alphabet zum Buchstabieren. Das phonetische Standardalphabet der NATO wurde vor diesem Hintergrund entwickelt und lässt sich auch beim Voice Coding einsetzen. Für eine höhere Arbeitsgeschwindigkeit haben sich beim Voice Coding allerdings optimierte Alphabete etabliert. „Das NATO-Alphabet enthält zu viele Silben pro Buchstabe, daher ist es nicht sehr effizient im alltäglichen Gebrauch“, sagt Wolle. Während die Buchstabenfolge *abcd* im NATO-Alphabet beispielsweise mit den Begriffen *alpha bravo charlie delta* insgesamt acht Silben umfasst, „sind es in meinem optimierten Voice-Coding-Alphabet nur vier: *air bat cap drum*“.

Nach demselben Prinzip kann man beim Voice Coding auch Symbole und Sonderzeichen eingeben. Um eine Klammer zu öffnen, sagt Wolle *paren*. Um sie zu schließen, sagt er *close paren*. Um den Cursor mit Sprache zum Beispiel nach oben oder nach links zu bewegen, sagt er *go up* oder *go left*. Um Tastenkombinationen mit der Shift- oder Strg-Taste anzusprechen, sagt er *shift air* oder *control cap*. Auch die Wiederholung von Aktionen kann man mit Voice Coding effizient umsetzen. Er kann



Ein Beispiel-Set-up für Voice Coding von der Hardware bis zu eigener Befehlsgrammatik (Abb. 2)

Durch die Verkettung von Befehlen lassen sich komplexe Ausgaben ohne Sprechpausen produzieren (Abb. 3).

zum Beispiel Ordinalzahlen als Suffix nutzen, um unter anderem große Sprünge mit dem Cursor mit nur wenigen Silben zu steuern – etwa mit dem Sprachbefehl *go up fifth* für einen Sprung um fünf Zeilen nach oben.

Diktate und Verkettung von Befehlen

Anders als bei der Standard-ASR-Software lassen sich beim Voice Coding beliebige Symbolsequenzen und Aktionen effizient diktieren, dafür sind Textdiktate der tatsächlichen Wortsequenzen durch Schlüsselwörter einzuleiten: In diesem Fall gibt die Software den erkannten Text wie ein herkömmliches ASR-Programm wieder, statt Aktionen auszuführen. Dies ist hilfreich beim Verfassen von Dokumentation im Quellcode, beim Benennen von Methoden oder bei speziell formatierten Dateinamen. Das Ende eines Textes signalisiert eine Sprechpause. Die Tabelle zeigt exemplarisch einige der typischen Varianten.

Um Sprechpausen zu umgehen, kann man auch Schlüsselwörter definieren, die den Anfang und das Ende eines diktierten Texts festlegen. So kann das Wort *say* als Schlüsselwort zum Einleiten des Diktats und *over* zum Beenden dienen (siehe Abbildung 3).

Webbrowsernavigation mit Sprache

Wolle nutzt Spracherkennungssoftware inzwischen nicht mehr nur zur Text- und Zeicheneingabe, sondern für praktisch alle Anwendungen am Computer. Für die Navigation durch das Internet verwendet er dabei die Browsererweiterung Vimium. „Dieser Ansatz klingt eigentlich danach, als würde man irgendein Handicap kompensieren wollen", erklärt er und fügt hinzu: „Was viele aber nicht wissen, ist, dass man mit Vimium schneller als mit Maus und Tastatur durchs Netz navigieren kann."

Per Tastendruck werden alle klickbaren Links auf der Webseite mit einer Buchstabenkombination ausgezeichnet (siehe Abbildung 4). Anschließend löst die Eingabe der Buchstabenkombination den Klick auf das jeweilige Element aus. Auch wenn Vimium eigentlich für die Navigation per Tastatur entwickelt ist, lässt es sich ebenso effizient per Sprachsteuerung nutzen: Die Buchstabenkombinationen werden lediglich mit dem NATO- oder einem anderen Alphabet buchstabiert – und nicht getippt.

Die Handsfree-Arbeitsweise bietet aber noch mehr Möglichkeiten. Shortcuts lassen sich zur mentalen Entlastung mit natürlicher Sprache verknüpfen, die die Benutzung des Computers intuitiver gestaltet. Statt kryptischer Tastenkürzel merkt sich Wolle semantisch bedeutsame Begriffe. „Ich sage zum Beispiel *find usages*, statt Alt + F7 zu drücken", sagt er und führt weiter aus: „Das ist für mich ein Game Changer beim Benutzen komplexer Anwendungen, da ich sonst die Shortcuts für viele Funktionen ohne Sprachinterface erst nachschlagen müsste und vermutlich oftmals gar nicht nutzen würde."

Zur Entwicklung von Software nutzt Wolle Werkzeuge wie IntelliJ oder Emacs, bei denen die Integration der Sprachsoftware noch tiefgreifender ist als bei Vimium: Talon kommuniziert hier über eine Programmierschnittstelle mit der Anwendung und ruft so Funktionen direkt auf, ohne erst Tastatureingaben emulieren zu müssen. „Dadurch kann ich per Sprachbefehl sogar Funktionen aufrufen, für die es gar keine Tastenkürzel gibt", erläutert er.

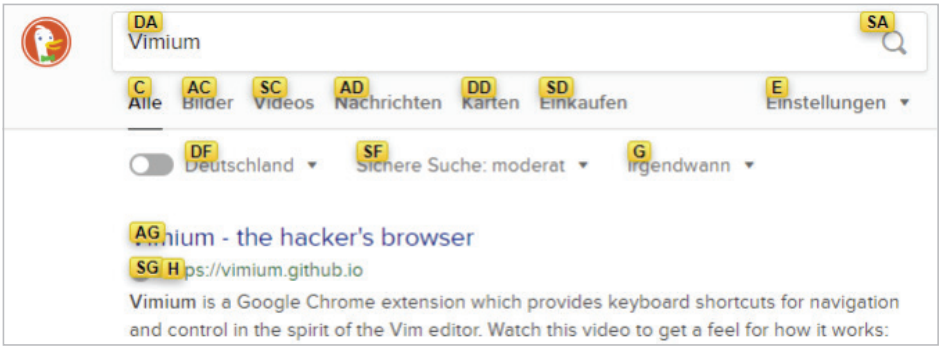
Zudem kann man Sprachbefehle überladen, um das Verhalten in unterschiedlichen Kontexten zu variieren. In Talon ist es möglich, für verschiedene Anwendungen oder Dateitypen Kontexte zu definieren, sodass etwa der Sprachbefehl zum Erzeugen einer Methode innerhalb eines JavaScript-Moduls einen anderen Output liefert als im Kontext einer C#-Klasse. Eine Äußerung wie *funky test funk* kann damit situationsabhängig sowohl „private void testFunk()" als auch „functions testFunk()" produzieren.

Eyetracking, Noise Recognition und Facial Actions

Eine ganz andere Eingabeform ist das Eyetracking. Die Hardware dafür ist heutzutage auch für Privatanwender verfügbar und sowohl Windows- als auch Apple-Betriebssysteme ermöglichen bereits die grundlegende Steuerung des Computers nur über die Augen. Auch wenn sich die einzelnen Ansätze im Detail unterscheiden, ist das Prinzip dabei immer ähnlich: Mit der Bewegung der Augen steuert man den Mauszeiger. Aktionen wie Links- oder Rechtsklicks kann man auslösen, indem man bestimmte Bereiche oder Aktionsflächen mit dem Blick für eine kurze Zeit fixiert. Jedoch ergeben sich durch das permanente Anstarren von Aktionsbereichen erhebliche Verzögerungen im Vergleich zur herkömmlichen Maus, sodass man zum Ausführen solcher Aktionen nach Möglichkeit einen anderen Mechanismus nutzen sollte. Obwohl Sprachbefehle für Links- oder Rechtsklick funktional sind, fallen hier jedoch auch zumindest spürbare Latenzen bei der Auswertung und Umsetzung der gesprochenen Worte an.

Wolle nutzt daher eine Kombination aus Eyetracking und Noise Recognition mit Talon. „Geräusche lassen sich mit deutlich geringerer Latenz als gesprochene Worte auswerten, da die im Fokus stehenden Audiosignale viel kürzer sind und nach keiner komplexen Grammatik interpretiert werden müssen", erklärt er. Deshalb macht er Pop-Geräusche, um einen Klick auszulösen. „Das ist zwar nicht für jede Büroumgebung geeignet, meine Kollegen haben sich aber mittlerweile dran gewöhnt, dass ich beim Arbeiten viele Geräusche mache", sagt er und schmunzelt.

In Talon kann man zwei verschiedene Modi zum Steuern des Mauszeigers verwenden. Im Headtracking-Modus steuert



Mit der Browsererweiterung Vimium kann man komplett freihändig durchs Internet navigieren (Abb. 4).

man mit den Augen die grobe Position des Mauszeigers über den gesamten Bildschirm hinweg, um dann mit Kopfbewegung nachzujustieren und per Geräusch schließlich eine Aktion auszulösen. Das Nachjustieren per Kopfbewegung ist notwendig, weil die Auswertung der Augenbewegung allein nicht präzise genug wäre. In Talons Zoom-Modus steuert man den Mauszeiger ebenfalls mit den Augen, allerdings vergrößert man per Geräusch zunächst den Bereich um den Mauszeiger herum und kann dann per Augenbewegung im vergrößerten Bereich nachjustieren – die Aktion wird erst mit einem zweiten Geräusch ausgelöst. Dieser Modus kann im direkten Vergleich etwas langsamer sein, ermöglicht aber hohe Präzision ohne Headtracking.

Wenn es denn mal ruhig sein soll, kann man Aktionen auch über die Mimik auslösen. Sie wird über eine Webcam registriert und von Software interpretiert. Hier bietet Talon zumindest für Mac-Nutzer grundlegenden Support, sodass man die auf Mimik basierenden Bedienungshilfen des Betriebssystems in der eigenen Befehlsgrammatik nutzen kann. Es gibt aber auch Tools, die sich extern nutzen lassen und die auch für andere Betriebssysteme verfügbar sind. Das Tool KinesicMouse wurde beispielsweise für die Nutzung mit der Microsoft Kinect entwickelt, wird jedoch – genau wie die Kinect selbst – nicht mehr vertrieben. Es gibt jedoch verschiedene Projekte, die auf der frei verfügbaren OpenFace-Gesichtserkennung aufsetzen und so eine plattformunabhängige Möglichkeit zum Auslesen von Facial Actions bieten. Bei der Integration von Soft- oder Hardware sind der Fantasie somit kaum Grenzen gesetzt. Dadurch entwickeln sich ständig neue Ansätze innerhalb der Community, die zum großen Teil aus Entwicklern besteht.

Effizientes Arbeiten

Wer effizient ohne Maus und Tastatur arbeiten will, sollte aber ein paar allgemeine Grundsätze beachten. Beim Voice Coding ist zunächst die Qualität des Audiosignals entscheidend dafür, wie präzise die Software die gesprochenen Anweisungen erkennt und umsetzt. Es empfiehlt sich deshalb, ein hochwertiges Mikrofon in einer möglichst ruhigen Umgebung zu nutzen. Das kann aber teuer sein. Zum Testen reicht ein günstiges kabelgebundenes Headset. Wer viel Geld in die Hand nehmen möchte, kann auch rund 700 Euro für das DPA 4188 ausgeben. Dieses Headset wird von vielen Handsfree-Usern gerne verwendet.

Darüber hinaus kann man ein Fußpedal nutzen, um Aktionen auszulösen oder das Mikrofon per Push-to-Talk an- und auszuschalten. Zum Erfassen der Mimik eignen sich herkömmliche Webcams. Zum Steuern des Mauszeigers mit den Augen ist jedoch ein spezieller Eyetracker notwendig, wobei die Anschaffungskosten hier abhängig vom Modell im Bereich von etwa 100 bis 300 Euro liegen. Wolle empfiehlt hier die Modelle Tobii 4C und Tobii 5.

Formatierung von Wortsequenzen

Eingabe		Ausgabe
	air bat cap	abc
uppercase	air bat cap	ABC
say	air bat cap	air bat cap
sentence	air bat cap	Air bat cap
camel	air bat cap	airBatCap
kebab	air bat cap	air-bat-cap
dotted	air bat cap	air.bat.cap
smash	air bat cap	airbatcap

Fazit: Einfach mal ausprobieren

Ob Sportverletzung, Karpaltunnelsyndrom oder Neugier: Es gibt viele gute Gründe, sich mit Alternativen zu Maus und Tastatur auseinanderzusetzen. Dabei sollte Handsfree Coding nicht als Gegenmodell zum Arbeiten mit Maus und Tastatur verstanden werden, sondern vielmehr als Ergänzung und als spielerischer Ansatz zum Steigern der Produktivität. Denn die nötige Software ist frei verfügbar, simpel in der Einrichtung und bringt obendrein Spaß beim Benutzen.

Wer also selbst mit einem Mausarm zu kämpfen hat oder schon als Kind mit einem Computer reden wollte, sollte vielleicht Voice Coding oder Eyetracking einfach mal ausprobieren – es gibt schließlich nichts zu verlieren. Mehr Informationen rund um das Thema und hilfreiche Ressourcen für die ersten Schritte gibt es auf der Webseite zur GI-Initiative „Handsfree Coding“. (mig@ix.de)

Quelle

GI-Initiative „Handfree Coding“:
www.handsfree-coding.gi.de

Dr. Wolfram Wingerath

verantwortet als Head of Data Engineering bei Baqend Entwicklung und Betrieb einer Infrastruktur für Analytics und Reporting. Bei der Arbeit mit dem Computer setzt er auf Schnittstellen wie Spracherkennung oder Eyetracking und hat rund zehn Jahre Praxiserfahrung mit Handsfree Coding gesammelt.