ZUR PROBLEMATIK LANG- UND KURZFRISTIGER PROJEKTE IM BEREICH DER COMPUTERLINGUISTIK

Manfred Thiel
Sonderforschungsbereich 100 /A2
Universität des Saarlandes
6600 Saarbrücken

Phänomene wie Theorien oder Systeme besitzen eine gewisse Lebensdauer und manche weisen auch in diesen Zeitgrenzen eine zyklische Entwicklung auf. Nicht nur auf dem Gebiet der LDV sind diese Zyklen eng mit dem Begriff des Redesign verbunden. Im folgenden soll auf die Problematik hingewiesen werden, die bei einer phasenverschobenen zyklischen Entwicklung von linguistischen Theorien, computerlinguistischen und softwaretechnologischen Systemkonzeptionen entstehen kann. Diese Überlegungen sollen also versuchen, die Probleme aufzuzeigen; es wird nicht die Absicht verfolgt, Lösungen anzubieten, höchstens eine: Erhaltung der Pluralität der Verhaltensalternativen von Projekten. Entsprechend abrupt endet der Beitrag.

Hat ein computerlinguistisches Projekt Zeit genug, um bestimmten Fragen auf den Grund zu gehen, entsteht zwangsläufig das Problem, daß es eigentlich alle 2-5 Jahre die linguistischen und/oder informatischen Grundlagen neu überdenken und Systeme neu konzipieren müßte. Tut man dies in einem langfristigen Projekt nicht, so ist es nach einem derartigen Zyklus

einerseits selten in der vordersten Reihe der Forschung, wenn es darum geht, z.B. neue linguistische oder informatische Konzepte zu realisieren (dies bedeutet nicht: nicht zu perzipieren!).

andererseits aber ist es ziemlich einfach, innerhalb von 2 oder 4 Jahren zu zeigen, daß z.B. eine linguistische Theorie in der Computerlinguistik anwendbar ist und zumindest einen Teil der Sprache beschreibt. Es bleiben aber immer Fälle offen, die man in dem relativ kurzen Zeitraum nicht erreicht. Z.B.: wieviele Jahre muß man an einem Parser arbeiten, bis er stabil und linguistisch derart mächtig ist, daß man damit einigermaßen sicher juristische Texte analysieren kann, in denen Sätze vorkommen, die länger als 100 Wörter sind? Wie verhält sich eine linguistische Theorie, wenn es darum geht, sog. "Ausnahmen" analysieren zu können, wobei "Ausnahmen" immer in Bezug auf das implementierte linguistische Modell zu verstehen ist? So können Regeln nicht bis zu einer filigranhaften Exaktheit im Detail restringiert werden, ohne daß ihre Allgemeingültigkeit verloren geht. Es hat sich außerdem gezeigt, daß die letzten Prozente der zu lösenden Probleme einen immer höheren Aufwand an Zeit und Arbeitseinsatz benötigen.

Daraus lassen sich folgende Thesen ableiten:

1. In unregelmäßigen Abständen entstehen neue theoretische Konzepte.
2. Ein System sollte nach einer gewissen Zeit einem Redesign oder einer Neukonzeption unterworfen werden, um den Anschluß an neue theoretische Konzepte nicht zu verlieren.
3. Nur wenn man den für ein System gewählten theoretischen Ansatz lange und intensiv genug verfolgt, können grundlegende Erkenntnisse, wie sie oben angedeutet wurden, gewonnen werden.

Der Widerspruch zwischen der 2. und 3. These könnte so mancher linguistischen und informatischen Konzeption Schwierigkeiten bereiten. Bei der Konzeption eines langfristigen Projektes muß also großer Wert auf die Planung der projektinternen Lebensdauer der gewählten theoretischen Grundlagen gelegt werden. Es ist dabei davon auszugehen, daß viele (d.h. thematisch anspruchsvolle) computerlinguistische Projekte eine verhältnismäßig langfristige Angelegenheit sein **müssen**. Andererseits darf man aber nicht vergessen, daß der Aufwand, die zu einem beliebigen Zeitpunkt eines Projektes übriggebliebenen Fragen zu lösen, überproportional steigt. So stellt sich für eine linguistische Theorie in einem sprachverarbeitenden System irgendwann das Problem, daß die Eleganz ihrer Konstruktion in Gefahr gerät, von Notlösungen und angebauten Balkönchen überwuchert zu werden. Lohnt es sich dann noch, weiter zu machen, oder soll man sich einer neuen Theorie zuwenden? Geschieht mit dieser dann das gleiche? Der Widerspruch liegt auch darin, daß diese Balkönchen gebaut bzw. ihre Ursachen erst einmal entdeckt werden müssen, um die Grenzen einer Konzeption abstecken und neue Ansätze aus diesen Erfahrungen aufbauen zu können. Ein sehr einfaches, aber doch typisches Beispiel für das oft beträchtliche und manchmal auch unübersehbare Wachstum eines Systems auch für den Fall, daß nur ein triviales Problem gelöst werden soll, ist folgendes: Zur Unterstützung der Wörterbuchkodierung wird eine Regel benutzt, die folgenden Inhalt hat:

> *wenn ein Substantiv auf das Suffix 'ung' endet, dann ist sein Genus feminin.*

Sofort fallen uns die problematischen Fälle ein: *"Dung"*, *"Sprung"* usw., also Beispiele, in denen *'ung'* nicht Suffix ist. Über die eigentliche linguistische Regel hinaus muß sich der Computerlinguist also noch etwas einfallen lassen. Es bieten sich dabei verschiedene Alternativen an:

1. Die Graphemfolge 'ung' wird gegebenenfalls als Suffix präkodiert. Damit ist aber prinzipiell nichts gewonnen: ob man kodiert: "ist feminin" oder: "ist Suffix" läuft auf das gleiche hinaus.
2. Die Wörter, bei denen 'ung' kein Suffix ist, müssen vorher alle im Wörterbuch stehen. Dies ist eine Notlösung und verursacht ein wenig wissenschaftliche Bauchschmerzen.
3. Eine morphologische Analyse stellt dagegen eine adäquate Vorgehensweise dar.

Allerdings tritt hier der diskutierte Effekt ein: die morphologische Analyse schafft selbst wieder neue Probleme, die, genau wie das hier diskutuierte, zu ihrer Lösung wieder anderen, möglicherweise

ähnlich großen oder noch größeren Aufwand erfordern. Sollte die Computerlinguistik als Vorlage für ein Bild von Escher herangezogen werden können? Dies scheint nicht der Fall zu sein: Es entstehen zwar neue Komponenten, die hochgradig voneinander abhängig sind, aber es ist zu erwarten, daß sich der Kreis irgendwann schließt und keine neuen Komponenten hinzukommen. Allerdings besteht immer die Gefahr, daß die Veränderung einer Komponente aufgrund der Vernetzung Auswirkungen auf andere haben kann.

Folgende Schlußfolgerungen sind daraus zu ziehen:

1. Die meisten Probleme in der Computerlinguistik sind nicht trivial.
2. Computerlinguistische Systeme sind von Natur aus sehr komplex; die Aussagekraft von Systemen, die sich auf einen Ausschnitt der gesamten Problematik beschränken, ist mit äußerster Vorsicht zu bewerten. Es scheint oft so etwas wie ein "Alles-oder-nichts-Effekt" einzutreten.
3. Es ist notwendig, zur Unterstützung der Entwicklung von sprachverarbeitenden Systemen Werkzeuge zu besitzen.

Steht ein langfristiges Projekt diese Durststrecke des Beharrens auf einem gewählten Ansatz durch, um danach sagen zu können, welche Anforderungen eine linguistische Theorie oder eine Softwarekonzeption erfüllen muß, ist dies aufgrund der breiten Variation an Grammatikalität und Akzeptanz in der natürlichen Sprache zugegebenermaßen recht undankbar. Es soll aber auch nicht die Problematik vergessen werden: Derartige "praktische" Arbeiten müssen irgendwann einmal auch in wissenschaftliche Neuansätze münden.