

Orm Finnendahl

Open Source

Kooperative Softwareentwicklung als kompositorisches Arbeitsmodell

erschienen in Positionen 69 (August 2006)

In den vergangenen Jahrzehnten hat in der Computergemeinde eine Bewegung für Aufsehen gesorgt, die –unter dem Namen *open source* allgemein bekannt geworden– ein neuartiges Modell der Herstellung und Lizenzierung von Programmen propagiert. In deren Folge haben sich andere Formen gemeinschaftlicher Entwicklung, Bewertung und Organisation etabliert, die sich längst nicht mehr ausschließlich auf die Herstellung von Software beschränken. Ich verfolge diese Entwicklung seit einigen Jahren mit großem Interesse und sehe in den aufgeworfenen Fragen viele Parallelen zu Fragen, die mich auch im Hinblick auf Musik beschäftigen. Daher interessieren mich die dort erprobten Lösungsstrategien sehr. Ich bin davon überzeugt, daß sich in der Folge des *open source*-Gedankens eine fundamental neue Form von Zusammenarbeit entwickelt, die aufgrund ihrer Dynamik beispielhaft für Kooperationsformen in der Kunst sein kann.

Open source tauchte als Synonym für den sehr viel älteren (aber mißverständlichen) Begriff der *free software* erstmals in den Jahren 1997-98 auf <fn1>. Im Unterschied zur davon unterschiedenen *closed source* meint der Begriff, daß Programme nicht nur in ihrer ausführbaren Version (dem sogenannten Executable) zur Verfügung stehen, sondern darüber hinaus die Quellen offengelegt werden, die es ermöglichen einzusehen, wie das Programm im einzelnen funktioniert und die es darüber hinaus ermöglichen, Modifikationen am Programm durchzuführen und es neu zu übersetzen. In der Regel sind *open source*-Programme frei erhältlich, während es sich bei kommerziellen Programmen fast ausschließlich um *closed source*-Programme handelt.

Bekannt geworden ist der Begriff im Zusammenhang mit dem Computerbetriebssystem GNU/Linux <fn2>, das vollständig aus *open source*-Programmen besteht. Die Offenlegung des Quellcodes hat weitreichende Konsequenzen für die Arbeitsform bei der Entwicklung von Programmen. Die Entwickler arbeiten zumeist unentgeltlich und freiwillig und sind über das gesamte Internet verstreut. Ihre Zusammenarbeit ist über öffentliche Mailinglisten koordiniert. Die Programmquellen liegen auf einem zentralen Rechner, zu dem nicht nur die Entwickler, sondern auch alle interessierten Anwender jederzeit Zugang haben. (Die Anwender dürfen allerdings keine Änderungen an den Quellen vornehmen). Zusätzlich wird der Inhalt des Zentralrechners regelmäßig auf sogenannte »Mirrors« gespiegelt, um mehr Anwendern den gleichzeitigen Zugang zu den Quellen zu ermöglichen. Das hat zur Folge, daß jederzeit der aktuelle Stand der Software heruntergeladen und genutzt werden kann. Veränderungen können eingesehen und verfolgt werden und ermöglichen jederzeit Reaktionen auf den Mailinglisten. Im Vergleich hierzu werden neue Versionen von *closed source*-Programmen üblicherweise in Abständen von mehreren Monaten bis Jahren veröffentlicht und beziehen die Anwender nicht direkt in die Diskussionen über die Entwicklung ein. Die Entscheidungsprozesse sind dezentral und aufgrund der Freiwilligkeit zumeist weniger hierarchisiert, als in einer Softwarefirma. Man könnte annehmen, daß eine solche Organisationsform langsam, wenig zielstrebig und ein daraus resultierendes Programm schlecht strukturiert und daher fehlerbehaftet ist. Es zählt daher zu den großen Überraschungen, daß *open source*-Programme in der Regel stabiler und zuverlässiger funktionieren, als deren *closed source*-Äquivalente.<fn3>

Ein anderes Beispiel, das eng mit dieser Arbeitsform verwandt ist, ist das sogenannte Wiki, das vermutlich am bekanntesten durch das Online Lexikon Wikipedia geworden ist. Auch hier ist

das Prinzip Freiwilligkeit und Quelloffenheit: Jeder darf Änderungen an jedem Textbeitrag (Artikel) durchführen. Wie auch bei der Programmentwicklung führt dies keineswegs zu einem chaotischen, schlecht recherchierten Lexikon, sondern ist mittlerweile in mancher Hinsicht klassischen Lexika sogar überlegen.

Die wichtigsten Eigenschaften und Gemeinsamkeiten von *open source*-Projekten seien hier zusammengefaßt:

1. Zumeist existiert zunächst keine eindeutige Hierarchisierung der Entscheidungsstrukturen. In der Regel beginnen ein paar Programmierer ein Ziel zu formulieren und stellen ihre ersten Arbeitsergebnisse zur Verwirklichung dieses Ziels ins Netz. Die weitere Entwicklung wird dann davon bestimmt, ob und inwiefern sich andere Personen an der Entwicklung, Fehleridentifikation und Dokumentation beteiligen.
2. Voraussetzung dafür ist die Zugänglichkeit des kompletten Materials für alle Beteiligten.
3. Die Kommunikation über die Entwicklung findet, für alle offen, in Form von Mailinglisten statt. Diese Mailinglisten sind der soziale Ort, an dem sich mit der Zeit Entwickler und Anwender herauskristallisieren, was letztlich durch den Kenntnisstand und die Kompetenz der Beteiligten bestimmt wird. Die Grenzen zwischen Anwendern und Entwicklern sind allerdings fließend und können im weiteren Verlauf durch die Vergabe von Schreibrechten auf die Quelldateien der zentralen Rechnerarchive angepasst werden.

Open Source in der Musik

Auch für Musik existieren mittlerweile netzbasierte Formen der Zusammenarbeit, wie beispielsweise Datenbanken für Klänge, netzbasierte Performances, Mailinglisten und Blogs, die jedermann offenstehen. In der Regel findet dort jedoch selten ein Austausch über die musikalischen Ergebnisse statt. Genau hier setzt mein Interesse an: Ich frage mich, ob und wie man diese Kooperationsweisen auch als Modelle für musikalische Reflexion während des Entstehungsprozesses nutzen kann, die nicht nur die rigide Trennung von Komponist, Interpret und Publikum aufheben, sondern durch die mögliche Kommunikation zwischen den drei Hauptbeteiligten am Musikprozeß das Potential für qualitativ andersartige Erlebnisse und Formen beinhalten. In der traditionellen Arbeitsweise existieren nur wenige Möglichkeiten des Austauschs zwischen Komponisten, Interpreten und Publikum. Zumeist arbeitet der Komponist während der Erstellung einer Komposition relativ unabhängig von den Interpreten und dem Publikum. Ich sehe in den langen Herstellungszeiten und der im Vergleich dazu kurzen Einstudierung und den wenigen Konzerten eine Korrespondenz zu den langen Veröffentlichungszyklen von *closed source*-Software. Gerade die auf breiterer Ebene geführte Auseinandersetzung über ästhetische Fragen oder über musikalische Mittel während der Erarbeitung aber halte ich in einer Zeit für zentral, in der Standardisierungen, die der Betrieb durch die Abhängigkeiten von Verlagen, Konzertforen, Ensembles oder all dem, was der »Markt« scheinbar vorgibt, fragwürdig geworden sind. Ich möchte in der Folge an drei Beispielen aufzeigen, in welche Richtung mein Denken geht und was ich unter einer Übertragung des *open source*-Gedankens auf zeitgenössische Arbeitsformen verstehe, ohne daß diese notwendigerweise immer internetbasiert sein müssen.

East by West

Im Jahr 2002 wurden in Zusammenarbeit der *TransMediaAkademie* (Hellerau), *Ars Electronica* (Linz), *V2* (Rotterdam) und *o1plus* (Nürnberg) Medienkünstler, Techniker und Programmierer eingeladen, um gemeinsame Ausstellungs- bzw. Präsentationsprojekte über einen längeren Zeitraum zu entwickeln. Die Teilnehmer kannten sich vorher nicht, sie lernten sich an einem

Wochenende durch gegenseitige Vorträge kennen. Wer sich mit wem zu welchem Projekt zusammentat, war vorher nicht festgelegt, sondern wurde den Teilnehmern überlassen. Das Projekt, an dem ich schließlich mitwirkte, wurde von dem Choreographen Johannes Birringer (Ohio, USA), der Medienkünstlerin Sher Doruff (Amsterdam) und mir erarbeitet.

Für mich war die Kommunikation mit Johannes Birringer und Sher Doruff während der Projektvorbereitung sehr ungewohnt. Sie fand ausschließlich über e-mail statt. Zunächst ohne genaue Vorstellung von der Art oder der Präsentationsform des Projektes entstand über einen Zeitraum von zirka einem Jahr die (technisch sehr aufwendige) interaktive, audiovisuelle Installation *East by West*, die in Hellerau und beim V2-Festival in Rotterdam präsentiert wurde: Zwei voneinander getrennte Räume werden durch einen internetbasierten Videostream verbunden, indem in beiden Räumen das gleiche Videobild an die Wand projiziert wird. Das Videobild besteht aus der Überlagerung der Kamera-Live-Aufnahmen beider Räume, so daß die Videoprojektion ermöglicht, wie durch ein Fenster in den anderen Raum zu sehen und eigene Handlungen mit den Handlungen auf der anderen Seite visuell zu verbinden. Zusätzlich sind dem Bild sich bewegende, farbig ausgefüllte Kreise als grafische Objekte zugemischt, die Klänge repräsentieren. In beiden Räumen gibt es zudem kugelförmige Objekte (an Fäden hängende Orangen in einem Raum beziehungsweise Boulekugeln im anderen Raum), deren Bewegungen mit Hilfe von Motiontracking-Systemen die Bewegungen der Kreise und ihre Überlagerungen steuern, was Auswirkungen auf den Klang und seine räumliche Verteilung (die in beiden Räumen identisch ist) hat. Anders ausgedrückt: die Zuschauer »spielen« auf beiden Seiten mit ihren jeweiligen Objekten ein gemeinsames Instrument, das nur dann klingt, wenn die Bewegungen auf beiden Seiten koordiniert ausgeführt werden und erzeugen gleichzeitig ein projiziertes visuelles Bild, das mit ihren Aktionen verbunden ist. Außer diesen gegenseitigen Abhängigkeiten von Handlungen und ihren relativ leicht nachvollziehbaren, visuellen und zugleich akustischen Auswirkungen, ist nichts festgelegt. Es handelt sich also im übertragenen Sinn um eine *open source*-Situation. Es war für uns sehr überraschend, daß die Besucher bei den Präsentationen, die bisher stattfanden, in beiden Räumen sehr schnell und auf sehr spielerische Weise miteinander zielgerichtet kommunizierten, indem sie beispielsweise die Boulekugeln so auf dem Boden plazierten, daß sie im projizierten Wandbild auf den Köpfen der Besucher im anderen Raum zu liegen schienen. Der Fokus der Handlungen wechselte dabei oft zwischen der visuellen oder akustischen Ebene, so daß sich immer wieder neue Angebote an die gegenüberliegende Seite ergaben, die jedoch, da es keine verbale Möglichkeit des Austauschs gab, nur mit Hilfe des Mediums selbst ausgedrückt werden konnten. Dennoch entstand aufgrund des gegenseitigen Bezuges von Visuellem und Akustischem und der Kommunikationssituation nicht der Eindruck von Beliebigkeit des Ergebnisses. Im Nachhinein ist mir aufgefallen, daß Entwicklung und Präsentation reziprok aufeinander bezogen sind: Gab es bei der Entstehung ausschließlich die verbale schriftliche Kommunikation, ist sie bei der Installation gerade die Form, die ausgespart ist. Ich könnte mir vorstellen, daß dies kein Zufall ist, obwohl es uns während der Arbeit nicht bewußt war.

Rote Asche

In dem bei den Weltmusiktagen 2004 am Opernhaus Luzern uraufgeführten Musiktheaterstück *Rote Asche* hat Johannes Schöllhorn ebenfalls einen offenen Ansatz gewählt, ohne daß das Ziel eine offene Form im Sinne des Begriffs von Umberto Eco war: Es wurde keine fertige Partitur in die Proben gebracht, sondern das Stück in den Proben gemeinsam entwickelt. Dazu gehörte auch, daß alle Beteiligten, also Komponist, Regisseur, Videokünstler, die drei Schauspielerinnen und die drei Interpreten Entscheidungen in allen künstlerischen Bereichen fällen konnten beziehungsweise sämtliche Entscheidungen prinzipiell zur Disposition standen, was nicht selten zu harten Auseinandersetzungen führte. Nur durch diese Möglichkeit der Mitbes-

timmung kam es beispielsweise dazu, daß die Schauspielerinnen in einer Szene gemeinsam mit den Musikern große Trommeln spielten, oder die Musiker auf der Bühne saßen und auch als Schauspieler an der Szene beteiligt waren. Bei Proben für eine Gewaltszene zwischen Musikern und Schauspielerinnen, bei der die Akteure von einem speziellen Trainer unterrichtet wurden, entstand durch die Bewegungen und Geräusche ein eigener Rhythmus, der die Grundlage für die musikalische Erarbeitung bildete.

Es wurden sehr viele einzelne Szenen erstellt, die gemeinsam diskutiert wurden und erst im Laufe der Probenarbeit entstand dann durch Auswahl und Gliederung die endgültige Form für die Aufführung. Um den notwendigen Spielraum zu erhalten, wurde die Musik über kleine Zellen, die aus momenthaften und unentwickelten musikalischen Materialien bestanden mit Hilfe von »Spielregeln« erarbeitet, die den Interpreten bis in die Aufführung hinein große Freiräume gaben. Auch hier war das Ergebnis keineswegs beliebig oder wirkte kompositorisch banal oder unausgearbeitet. Im Gegenteil hatte ich den Eindruck, daß sich lediglich die überzeugendsten Ideen durchgesetzt hatten und –angesichts der Arbeitsform sicher nicht überraschend– eine hohe Identifikation mit der Produktion bei allen Aufführenden spürbar war. Im Resultat entstand eine Form des Musiktheaters, bei der die verschiedenen Ebenen Schauspiel, Video und Musik in einer Form integriert und aufeinander bezogen waren, wie ich es bisher noch nicht erlebt habe.

grep

Auch in meiner Komposition *grep* für Altsaxophon, Perkussion und Computer aus dem Jahr 2003 habe ich verschiedene Ansätze verfolgt, vom *open source*-Gedanken angeregte Ideen der Kooperation in meine Arbeit zu integrieren. Der Computer wurde als Bindeglied zwischen Komposition und Improvisation eingesetzt: Eine für die Musiker sichtbare Grafik verdeutlicht die Transformationsverfahren des Rechners im zeitlichen Verlauf, ohne, daß die Ausgangsklänge, die transformiert werden, festgelegt sind. Die Aufgabe des Komponisten besteht darin, einen differenzierten formalen Ablauf der Aufnahme und Weiterverarbeitung von Live gespielten Klängen und Klangpartikeln als Struktur festzulegen und transparent zu machen. Die Aufgabe der Musiker besteht darin, Klänge, Partikel und Klangfolgen (oder auch Pausen) im Kontext der festgelegten Struktur zu erproben. Dabei ist ihnen freigestellt, wie sie sich zu dieser Struktur verhalten. Die Probenarbeit bestand vor allem aus einem Austarieren von Form und Spielaktionen, das in gemeinsamer Auseinandersetzung von allen Seiten Änderungen erforderte. Durch die Verwendung des Computers für die Generierung syntaktischer Verknüpfungen der klanglichen Transformationsverfahren, ist es mittlerweile möglich, auch komplexe und umfangreiche Änderungen der Struktur in sehr kurzer Zeit vorzunehmen. Diese Form der Arbeit hat mich vor allem deswegen gereizt, weil es ein nicht triviales Wechselverhältnis zwischen der trotz ihrer Flexibilität präzisen Vorhersehbarkeit der Transformationen und der Offenheit des Ergebnisses gibt. Mein Eindruck war, daß gerade diese Kombination die Beteiligten provozierte, miteinander zu kommunizieren, eine Form der gemeinsamen Auseinandersetzung und eine Sprache dafür zu finden.

Anmerkungen

<fn1> siehe hierzu auch: http://de.wikipedia.org/wiki/Open_source

<fn2> GNU wird von der von dem Programmierer Richard Stallmann gegründeten Free Software Foundation entwickelt/Linux bezeichnet einen Unix-ähnlichen Betriebssystemkern, der seit 1990 von Linus Torvalds (mittlerweile in Zusammenarbeit mit vielen anderen Programmierern) entwickelt wird.

<fn3>siehe hierzu auch den Artikel "Die Kathedrale und der Basar" von Eric S. Raymond (http://de.wikipedia.org/wiki/Die_Kathedrale_und_der_Basar) oder die folgende Aussage der Computerfirma Apple (von <http://opensource.org/weblog/>):

"Apple has stated that they are depending on Open Source to spare OS X the security woes plagued by Microsoft.

The operating system update, due to debut in the first half of 2005, is based on the Unix platform, and Apple executives reckon the open-source nature of the product means it's inherently more secure than certain proprietary offerings.

Bertrand Serlet, senior vice president of software at Apple, said Wednesday that having a greater number of people keeping an eye on source code leads to better software security. „A lot of security problems derive from the core," he said. With open-source code, „thousands of people look at the critical portions of source code and...check those portions are right. It's a major advantage to have open-source code."

(C|net via OSD|r)"