

Die hierin beschriebene Software ist eine neuartige Möglichkeit, sämtliche Tätigkeiten, welche zur Befriedigung menschlicher Bedürfnisse notwendig sind, auf Augenhöhe zu vermitteln und somit sowohl für andere direkt da zu sein, als auch sich in komplexe (Re-)Produktionsprozesse einzubringen. Sobald Bedürfnisse vermittelt wurden, wird automatisch das lokale Umfeld nach verfügbaren Mitteln (Werkzeugen, Maschinen, Arbeitsmaterialien, Räumlichkeiten, Anleitungen, etc.) analysiert und Benutzern mögliche Tätigkeiten angeboten, denen diese sich den eigenen Fähigkeiten und Interessen nach zuordnen können. Diese Tätigkeiten selbst sind Erfahrungen anderer Beteiligter, die einmal vor demselben Problem standen. Die Software kann herausstellen, welche Tätigkeiten im jeweiligen Umfeld am effizientesten dazu beitragen, möglichst viele unterschiedliche Bedürfnisse mit möglichst geringem Aufwand zu befriedigen und diese Informationen den Benutzern zur Verfügung stellen.



Commons Institut e.V.

The Timeless Way of Re-Production

Die Entwicklung gesellschaftlicher
Selbstorganisation



...C...

Marcus Meindel

- 1 *History time*: Auf folgender Veranstaltung wurde erstmalig im deutschsprachigen Raum die Verbindung zwischen Alexanders Theorie und Commoning geschlagen: <https://lebensfuelle.files.wordpress.com/2010/03/musterlebendigkeit.pdf> (Zuletzt abgerufen: 28.10.19)
- 2 *Softwarekonzept*: In gewisser Weise sind es auch „Muster von Mitteln“, da nicht einzigartige Mittel gemeint sind, sondern solche, die tendenziell häufiger vorhanden und im Sinne ihrer Anwendung „gleich“ sind. Auf das „Muster“ bei den Mitteln wird fortlaufend verzichtet. Im *Softwarekonzept* wird näher darauf eingegangen.
- 3 Die Einteilung der Mittel wurde von Sutterlütti/Meretz übernommen (Kapitalismus aufheben, S.138)
- 4 Die Muster finden sich in Alexanders „*The Pattern Language*“
- 5 *Helfrich/Bollier*: In diesem Text wird der Einfachheit halber von „lokalen Kontext“ gesprochen, wobei damit auch „virtuell“ gemeint sein kann. Die „kosmo-lokale Produktion“ bei Helfrich und Bollier deutet auf ein ähnliches Konzept hin: „So nennt man ein System des bedürfnisorientierten Schaffens und Bereitstellens, in dem Menschen »leichte« Dinge wie Wissen und Design über das Internet anderen zur Verfügung stellen und gemeinsam nutzen, auch durch Lernen mit Gleichrangigen, jedoch »schwere« Dinge wie Maschinen, Autos, Wohnraum, Möbel und elektronische Geräte vor Ort bauen“, H/B, „Fair, Frei, Lebendig“ (2019): S.79-80
- 6 *Helfrich/Bollier*: Die beiden Autoren unterscheiden drei Formen des Teilens: „Aufteilen“ (Dinge nutzen sich ab oder werden durch das Teilen weniger), „gemeinsam nutzen“ und „weitergeben“ (Dinge werden mehr, wenn sie geteilt werden (z.B. Daten/Informationen/etc.). H/B, „Fair, Frei, Lebendig“: S.85
- 7 *Softwarekonzept*: In der Textreihe wird weiter zwischen *Fähigkeit* und *Qualifikation* unterschieden. Eine Fähigkeit kann dabei selbstständig erreicht, eine Qualifikation muss durch eine andere Person oder Institution zugeschrieben werden.
- 8 *Helfrich/Bollier*: Auf diesem Weg verändert sich das Bewusstsein der Teilnehmenden selbst. Die Veränderung eines Weltbildes ist für Helfrich und Bollier ein „Onto-Wandel“: „Er bezeichnet eine Verschiebung des (eigenen) Seinsverständnisses (Ontologie) – das heißt, der grundlegenden Annahmen über das (Mensch-)Sein, die Natur und die Struktur der Wirklichkeit. Seinsverständnisse sind tief verankert. Sie spiegeln sich in unseren Wahrnehmungen, unserer Art, in der Welt zu sein und folglich auch darin, welche politische Ökonomie und Koordinationsstrukturen wir für möglich und wünschenswert halten. Ein Onto-Wandel ist Voraussetzung für eine tiefgreifende Veränderung der politischen, ökonomischen und sozialen Verhältnisse. Kurz: der Kultur.“ (H/B, S. 82)
- 9 *Softwarekonzept*: Eine nähere Ausdifferenzierung des Tätigkeitsgewichtes findet sich in der *Softwarekonzept* Textreihe.
- 10 *Softwarekonzept*: Die Bedürfnisgewichtung wird im *Softwarekonzept* auch als Transformationsvariable bezeichnet, da durch sie Entstehung und Verbreitung überhaupt erst ihren Anfang nimmt, aber ihre Wichtigkeit mit der Ausbreitung des Commonings nachlässt.
- 11 *Softwarekonzept*: Bei manchen Projekten kann es sinnvoll sein, wenn der „Vision“ einer einzelnen Person bzw. Gruppe gefolgt wird, welche die entsprechenden Tätigkeitsmuster von vornherein fest anlegt und eventuell zur Selbstzuordnung freigibt. In der *Softwarekonzept*-Reihe wird das als Projektplanung bzw. „manueller Konfigurationsprozess“ bezeichnet.
- 12 Die Designprinzipien im Wortlaut: https://www.nobelprize.org/uploads/2018/06/ostrom_lecture.pdf. S. 422. (zuletzt abgerufen: 20.9.19). Übersetzung von Silke Helfrich in „Frei, Fair, Lebendig“: S.317-318
- 13 Johannes Euler hat die hier wohl eindrücklichsten neuen Definitionen von Commons und Commoning gefunden: „Commons is the social form of (tangible and/or intangible) matter that is determined by commoning“. Und: „Commoning shall be described as voluntary and inclusively self-organized activities and mediation of peers who aim at satisfying needs“ (*Conceptualizing the Commons*, in *Ecological Economics* 143, S.12).

Stand: 25. Januar 2020

Eine Gesellschaft, welche sich ausschließlich den menschlichen Bedürfnissen, Fähigkeiten und Interessen nach gestaltet, ist bisher bloße Sehnsucht, könnte aber durch das im Internet liegende Potential realisiert werden. Aber wie kann sich dieses Potential – diese Möglichkeit eines jeden Menschen mit potentiell allen Menschen auf Augenhöhe Informationen auszutauschen – dahingehend verwirklichen, dass diese von uns zwar noch nicht vollständig begriffene, aber doch bereits angestrebte Gesellschaftsform erreicht wird? Und wie kann die Kooperation von Tätigkeiten innerhalb komplexer Re-Produktionsprozesse auf Augenhöhe koordiniert werden, wenn deren gemeinsamer Zweck als möglichst weitreichende Befriedigung von Bedürfnissen definiert ist? Wie es der Zufall will, könnten es die Arbeiten eines Architekten – Christopher Alexander – sein, über welche sich diese Fragen beantworten lassen.

Silke Helfrich und David Bollier waren es, die, inspiriert von der Arbeit Helmut Leitners und Franz Nahradas¹, in „Die Welt der Commons – Muster des gemeinsamen Handelns“ zum ersten Mal Alexanders Theorie mit Commoning in Verbindung brachten und diesen Weg in „Frei, Fair und Lebendig – die Macht der Commons“ konsequent weitergingen. Helfrich und Bollier erforschten hierfür die Praxis des Commonings und schlossen dabei durch Alexanders Methode auf wiederkehrende Muster, welche in langlebigen Prozessen der freiwilligen Selbstorganisation unter Gleichrangigen immer wieder in Erscheinung treten. Diese Muster sind praxisnahe Hilfsmittel, um Commoning zu verstehen, anzuwenden und zu verbreiten. Wenn dieser vorliegende Text auch in anderer Weise mit Alexanders Theorie umgeht, wäre er ohne die gemeinsamen Forschungsergebnisse von Helfrich und Bollier kaum möglich gewesen.

Was also ist das Besondere an Christopher Alexanders Arbeit? Alexander hat sich intensiv damit auseinandergesetzt, was Lebendigkeit bedeutet, wie diese entstehen kann und wie er – in seiner Rolle als Architekt – unterstützend wirken kann, damit diese Lebendigkeit von jedem einzelnen Menschen aus selbstständig verbreitet werden kann. Was er zu diesem Zweck herausgestellt hat, nennt er eine „Mustersprache“ und wie eine solche Mustersprache entstehen kann, beschreibt er in seinem Werk „The Timeless Way of Building“. Anhand direkt zitierter Abschnitte wird nachfolgend der Struktur dieses Werkes gefolgt werden. Hierdurch wird nicht nur auf die Grundstruktur einer Software für ununterbrochenes Commoning geschlossen, sondern auch dargestellt, wie Formen softwareunabhängiger Selbstorganisation aus dieser Vermittlungsform entstehen bzw. sich in diese Struktur integrieren können. Ziel dieser Arbeit ist es herauszustellen, wie diese Software tatsächlich zeitlos werden und ihre Funktion als Werkzeug zur Herstellung und Erhaltung einer von Commoning bestimmten Gesellschaft immer effizienter erfüllen kann.

Die Textreihe „Ein Softwarekonzept für ununterbrochenes Commoning“ ist die Konkretisierung dieser Interpretation.

Register

Aufwand (S.7): Die Anstrengung, welche unabhängig von ihrer zeitlichen Dauer und des Ortes ihres Auftretens mit der Tätigkeit *im Allgemeinen* zusammenhängt. Der Aufwand bestimmt die *Prozessqualität*.

Bedarf (S.6): Die Mittel, welche zur Ausführung einer Tätigkeit notwendig sind.

Bedürfnis: Der Begriff wird hier auf alles reduziert, was Personen als Mangel erleben und daher *brauchen* bzw. *wollen*. Auf Softwareebene weiter beschränkt auf die Vermittelbarkeit.

Bedürfnisgewicht (S.16): Wirkt sich auf das Tätigkeitsgewicht aus und wird durch Beteiligung am Commoning angesammelt. „Zieht“ den Gesamtprozess in Richtung der eigenen Bedürfnisse

Bibliothek (S.14): Sammlung aller *Tätigkeitsmuster*, welcher sich eine Person prinzipiell annehmen würde. Möglichkeit *Fähigkeiten* zu definieren und Benachrichtigungen einzustellen.

- im Commons (S.32): Sammlung aller *Resultate*, welche das Com. erzeugen kann

Commoning (S.2): Freiwillige und selbstorganisierte Tätigkeit auf Augenhöhe zum Zweck der Bedürfnisbefriedigung.

ununterbrochenes -: Die im Text beschriebene Form des softwaregestützten Commonings, in welcher jede Tätigkeit prinzipiell unabhängig von *bestimmten* Prozessen der Bedürfnisbefriedigung ist und über solche *hinausgehen* kann.

Commoning-Prozess (S.8): Ein einzelner *Commoning-Prozess* ist die Befriedigung eines bestimmten Bedürfnisses durch in Kooperation stehenden Tätigkeiten, welche auf Softwareebene durch Tätigkeitsmuster beschrieben werden.

Aktivierung eines - (S.22): Kann geschehen, wenn sich zu jeder zu einer bestimmten Bedürfnisbefriedigung notwendigen Tätigkeit Personen zugeordnet haben.

Commons: In der neueren Forschung (siehe Endnote 13) die soziale Form von *Mitteln* im Prozess des *Commonings*.

- Bei Ostrom: S.30

Effizienz: Bezieht sich im Commoning immer auf die Bedürfnisbefriedigung und ist direkt abhängig von der *Qualität* (S.7) und indirekt abhängig von dem *Gewicht* und der damit ermöglichten *Kontinuität* (S.10) von Tätigkeitsmustern.

Fähigkeit (S.11): Auf Softwareebene durch *verinnerlichte* Tätigkeitsmuster definiert. Definierte Fähigkeiten bieten die Möglichkeit zur Selbstzuordnung in komplexe Tätigkeitsmuster.

Freischaltung (S.20): Moment des *Konfigurationsprozesses*, in welchem neue Tätigkeitsmuster zur Selbstzuordnung Beteiligten vorgeschlagen werden.

Konfiguration (S.8): geschlossene Anordnung der Tätigkeitsmuster, welche einen Commoning-Prozess formen.

-sprozess (S.20) Schrittweises *Freischalten* von Tätigkeitsmustern zur Selbstzuordnung, bis ein geschlossener Commoning-Prozess entsteht.

Kontext (lokaler) (S.13): Lokale Umgebung, in welcher eine Tätigkeit auftritt und sich auf ihre *Prozessqualität* auswirkt

Kontinuität (S.10): Moment, wenn einer Tätigkeit über einen einzelnen Commoning-Prozess hinweg nachgegangen wird.

- Rolle im Verdichtungsprozess: S.15

Krise (S.28): Das Zusammenwirken verschiedener *Störungen*, welche den Gesamtprozesse dermaßen beeinträchtigen, dass eine hohe Zahl aktiver Prozesse der (existenziellen) Bedürfnisbefriedigung ausfallen.

Mittel (S.6): Gemeint sind alle gegenständlichen, symbolische oder sozialen „Dinge“, welche innerhalb eines Commoning-Prozesses verwendet werden können.

- in Commons: S.32

Mustersprache: Ein von C. Alexander entwickeltes System, das Anwender*innen erlaubt, lebendige Architektur zu entwerfen.

- Sprache aus Tätigkeitsmustern (S.10): Ansammlung gesellschaftlich verwendeter Tätigkeitsm., welche den Re-Produktionsprozess dieser Gesellschaft bestimmen.

- Ziel einer Mustersprache: S.14

begrenzten Form noch nie gesellschaftlich bestimmend sein konnte. In gewissen Sinne ist diese Form des Commonings, welche mit der Entstehung des Internets überhaupt erst denkbar wurde, damit die Rückbesinnung auf eine vorgesellschaftliche Beziehungsform, die romantisch verklärt sein könnte, aber jetzt in einem *grenzenlosen* Maßstab realisiert werden kann.

„At this level of analysis, we can compare many different building processes. And it turns out that, invariant, behind all processes which allow us to make buildings live, there is a single common process. But though this method is precise, it cannot be used mechanically. Indeed it turns out, in the end, that what this method does is simply free us from all method.“ (12-13)

Zweck des Commonings ist die Befriedigung vermittelter Bedürfnisse. Welche Bedürfnisse eine Person hat, genauso welchen Bedürfnissen sich eine Person annehmen möchte, kann diese dabei nur selbst wissen. Es gibt keine Autorität – sei es ein Planungskomitee, sei es ein Computer-Algorithmus –, welche eine Gesellschaft nach Bedürfnissen einrichten kann. Die zu entwickelnde Software ist daher auch keine Entscheidungsinstanz. Es ist ein Werkzeug, das ununterbrochene gesellschaftliche Kooperation zur *generellen Bedürfnisbefriedigung* ermöglichen soll. Aber die Software selbst ist nur die vermittelnde Instanz. Davon abstrahiert bleiben nur Menschen, die sich gegenseitig unterstützen, weil das gute Leben des einen abhängig ist von dem guten Leben des anderen.

“To purge ourselves of these illusions, to become free of all the artificial images of order which distort the nature that is in us, we must first learn a discipline which teaches us the true relationship between ourselves and our surroundings.” (15-16)

Über die Software sollen die gesellschaftlichen Kooperationsprozesse von der Verwendung des Naturvermögens und deren Erhaltung bis zur komplexen Produktion und dem Bereich der Fürsorge transparent werden. Und was bei Alexander das Erlernen einer Disziplin ist, wird hier zur Funktion einer Software. Wenn es an etwas fehlt, dann muss einsichtig sein, wo sich eingebracht werden kann, um diesen Mangel zu beheben. Wenn etwas gelernt werden will, dann muss einsichtig sein, wo es gelernt und ausgelebt werden kann. Die eigenen Bedürfnisse mögen auf den eigenen Körper begrenzt sein, aber sind bezogen auf sämtliche Orte, an denen wir sind, die Natur, die uns umgibt und auch auf das Wohlbefinden der Menschen, mit denen wir zu tun haben. All das umfasst die Struktur des ununterbrochenen Commonings in der Befriedigung der eigenen Bedürfnisse. Ob das die „wahre Beziehung“ („*true relationship*“) zwischen uns und unserer Umwelt ist oder nicht – es ist diejenige, welche wir anstreben.

“Then, once this discipline has done its work, and we pricked the bubbles of illusion which we cling to now, we will be ready to give up the discipline, and act as nature does. This is the timeless way of building: learning the discipline – and shedding it.” (16)

welche die zerstörerischen Auswirkungen in sich trägt, die wir sowohl sehen als auch spüren können. Weil es auch unsere Welt ist, die damit zerstört wird und weil nichts akzeptiert werden darf, das dieser Welt ihre Schönheit nimmt. Und auch, weil es nicht länger sein kann, dass wir – sowohl ich, als auch die Menschen, die ich liebe – unsere Lebenszeit lang für die Bereicherung anderer arbeiten müssen, nur um unsere Existenz zu rechtfertigen. Denn Commoning bedeutet nicht nur, hinsichtlich der menschlichen Bedürfnisse selbst, sinnvoll tätig zu sein. Es bedeutet nicht nur, die Freiheit von zerstörerischen gesellschaftlichen Mechanismen und die Möglichkeit, mit der Natur so umzugehen, wie wir es als richtig empfinden. In letzter Konsequenz bedeutet Commoning ganz besonders: Endlich und dauerhaft unsere Ruhe haben – vor den Chefs, den Vermietern und dem Arbeitsamt.

Und ich bin jeder Person aus ganzem Herzen dankbar, die sich – in welcher Weise auch immer – an diesem Prozess beteiligt.

"And [the quality] happens when our inner forces are resolved – And when a person's forces are resolved, it makes us feel at home, because we know, by some sixth sense, that there are no other unexpected forces lurking underground. He acts according to the nature of the situations he is in, without distorting them. There are no guiding images in his behavior, no hidden forces; he is simply free. And so, we feel relaxed and peaceful in his company" (51)

Im parallel zu „*The Timeless Way of Building*“ entstanden Werk „*The Pattern Language*“ hat Christopher Alexander architektonische Muster herausgestellt, welche diese Qualität ohne Namen hervorbringen sollen. Diese Qualität entsteht, indem diese Muster schrittweise angewendet werden um innere Spannungen („*inner forces*“) aufzulösen.

Für das Commoning wird die *Qualität ohne Namen* als Zustand interpretiert, in welchem *alle eigenen Bedürfnisse befriedigt sind*. Die *Spannungen* werden dabei als Aufwand betrachtet, welcher für die Bedürfnisbefriedigung notwendig ist. Ein Bedürfnis „hat“ daher nicht eine bestimmte Spannung, sondern diese ergibt sich erst durch die Möglichkeiten der Befriedigung und ist umso höher, je *aufwendiger* die Bedürfnisbefriedigung ist. Und jede *Tätigkeit* im Commoning hat den Zweck der Bedürfnisbefriedigung, sprich, der Auflösung von solchen Spannungen. Auf Softwareebene sind diese Tätigkeiten dabei als *Tätigkeitsmuster* beschrieben. Als solche Tätigkeitsmuster lassen sie sich über die Software abrufen und können potentiell jeder Person beschreiben, wie eine bestimmte Spannung abgebaut werden kann. Die *Qualität eines Tätigkeitsmusters* sagt dabei sowohl aus, inwiefern eine darin beschriebene Tätigkeit dazu geeignet ist, *im jeweiligen Umfeld* das Bedürfnis zu befriedigen und außerdem, welcher *Aufwand* dazu notwendig ist.

“In order to define this quality in buildings and in towns, we must begin by understanding that every place is given its character by certain patterns of events that keep on happening there.” (55)

Diese „*events*“ (Ereignisse) werden als *Tätigkeiten* interpretiert. Betrete ich ein Hotel, dann sehe ich Menschen, die anderen Schlüsselkarten überreichen. Ich sehe Menschen, die große Wäschesäcke in Karren transportieren. Ich sehe Menschen telefonieren und Reservierungen vornehmen. Ich sehe Menschen, die andere unterweisen. Und besuche ich Freunde, dann sehe ich Menschen, die Wein trinken, die kochen, die Kinder ins Bett bringen, die sich gegenseitig Geschichten erzählen und so weiter. Allein durch die Tätigkeiten, die an diesen Orten stattfinden, kann ich beides voneinander trennen: Das Hotel und die Wohnung der Freunde.

Für das Commoning, zumindest soweit es über die Software vermittelt werden kann, ist allerdings nicht jede einzelne Tätigkeit relevant. Relevant hierfür sind nur solche Tätigkeiten, welche der *direkten Bedürfnisbefriedigung* dienen und dabei eine Abhängigkeit von dem Beitragen anderer besteht.

“These patterns of events are always interlocked with certain geometric patterns in the space. Indeed, as we shall see, each building and each town is ultimately made out of these patterns in

THE KERNEL OF THE WAY

"From what you have read so far, it may seem as though the life of buildings, and the timeless character they have when they are living, can be created simply by the use of pattern languages. If the people have a living language, it seems that what emerges from their acts of building will be alive; it seems as though the life of towns can be created simply by the use of languages. - And yet, we wonder, can it be so simple? Can any theory be so powerful? - These doubts are right. There is a kernel at the center of the timeless way, a central teaching, which I have not described till now. - The essence of this kernel is the fact that we can only make a building live when we are egoless." (535)

Das Beste am softwarevermittelten Commoning ist, dass wir alle unser Ego behalten können. Für Alexander ist dieser Punkt allerdings zentral, da die Anwendung der Mustersprache für ihn nicht nur ein *innerer Prozess* ist, sondern sogar eine Form der *Therapie*: Seiner Ansicht nach, sind wir alle befähigt, die Welt lebendig zu gestalten („*The language, and the processes which stem from it, merely release the fundamental order which is native to us*“, 531), müssen uns aber – um dieses Wissen wieder zu erlangen – von dem bisher Gelernten freimachen („*the builder must let go of all his willful images, and start with a void*“, 538) und können erst von diesem Zustand aus durch die Anwendung der Mustersprache eine lebendige Welt erschaffen („*At this stage, the buildings life will come directly from your language*“, 539). Doch Orte können aus „guten Mustern“ bestehen und doch tot sein („*one place can have »good« patterns in it, and yet be dead*“, 541), sowie andere Orte auch ohne jegliche Muster lebendig sein können („*Another place can be without the patterns which apply to it, and yet still be alive*“, 542). Wenn es bisher auch so schien, ging es Alexander nicht um die Mustersprache an sich, sondern die konsequente Anwendung der Muster ist für ihn ein *Lernprozess*, durch welchen der Mensch (wieder) dazu befähigt wird, die auf ihn wirkenden *Spannungen* zu erkennen und die Welt mit dem durch die Muster erworbenen Wissen so zu gestalten, dass diese Spannungen aufgelöst werden. „*The language is the gate*“, schreibt Alexander, „*which leads you to the state of mind, in which you live so close to your own heart that you no longer need a language. - This is the final lesson of the timeless way*“ (547).

Die Software selbst erzeugt die verschiedenen Möglichkeiten der gesellschaftlichen Kooperation, die zur Auflösung der *Spannungen* zwischen vermittelten Bedürfnissen und deren Befriedigung dienen können. Sie ist dabei nicht geprägt von Bildern und Konzepten, von denen sie sich erst distanzieren muss. Sie wertet anhand des jeweiligen lokalen Kontextes den ihr bekannten Daten nach aus, welche Tätigkeiten *idealerweise* dazu beitragen können *möglichst viele* Spannungen abzubauen. Ihre Grenzen sind dabei gesetzt durch die eingespeisten und gepflegten Meta-Daten der verfügbaren Mittel (Lokalität, Zustand, Menge, Nutzungsbedingungen, etc.) und den durch sie verarbeitbaren Mustern, diese Mittel durch Tätigkeit den Bedürfnissen entsprechend *umzuwandeln* bzw. *anzuwenden*. Je mehr dieser Daten der Software zur Verfügung

Die Qualität des *Resultates* (etwa ein Moment der Fürsorge oder ein gegenständliches Produkt) ist durch *Sinne* erfahrbar und muss *funktional* zur Befriedigung eines bestimmten Bedürfnisses oder Deckung eines bestimmten Bedarfes sein. Diese Qualität wird daher als *sinnlich-funktionale Qualität* bezeichnet und ist umso höher, je vollständiger sie eine bestehende Spannung auflöst.

Die *Prozessqualität* richtet sich dagegen nicht danach, inwiefern eine bestehende Spannung abgebaut wird, sondern ist umso höher, je *weniger neue Spannungen* durch die Tätigkeit entstehen. Die Prozessqualität besteht dabei sowohl aus dem *Aufwand der bestimmten Tätigkeit*, als auch dem Aufwand, welcher zur *Verfügbarmachung* der dafür notwendigen Bedarfe benötigt wird. Der Aufwand der bestimmten Tätigkeit ist dabei die Anstrengung, welche unabhängig von ihrer zeitlichen Dauer und des Ortes ihres Auftretens mit der Tätigkeit *im Allgemeinen* zusammenhängt, sobald sämtliche dafür notwendigen Mittel vorhanden sind. Der Aufwand zur Deckung des Bedarfes ist dagegen abhängig davon, welche Mittel in lokaler Umgebung verfügbar sind und entweder direkt zugeordnet oder durch andere Tätigkeiten in die notwendigen Mittel *umgewandelt* werden können. Die Prozessqualität eines bestimmten Tätigkeitsmusters ist damit umso höher, je mehr „aufwendige“ Mittel zur Durchführung der Tätigkeit bereits lokal verfügbar sind bzw. je weniger aufwendig die *Verfügbarmachung* der nicht direkt zuordenbaren Mittel ist. Eine hohe Prozessqualität heißt damit, dass tendenziell *weniger neue Spannungen* durch geistige und körperliche Erschöpfung sowohl im Rahmen der Tätigkeit als auch im Rahmen des Commoning-Prozesses entstehen. Es ist dabei eine noch *offene Frage* wie sich der zeit- und ortsunabhängige Aufwand einer Tätigkeit bestimmen lässt. Damit die Software ihren Zweck erfüllen kann, muss diese Frage noch in einer angemessenen Weise beantwortet werden.

Beide Qualitäten einer Tätigkeit sind aufeinander bezogen: Einerseits indem die Durchführung einer Tätigkeit nicht sinnvoll ist, wenn das Resultat daraus nicht von ausreichender *sinnlich-funktionaler Qualität* ist. Andererseits indem bei einer niedrigen *Prozessqualität* ein verhältnismäßig hoher Aufwand entsteht, sprich neue Tätigkeiten angestoßen werden müssen. Der allgemeine Begriff der *Qualität* bezeichnet folgend einen mittleren Wert beider Aspekte.

“The more living patterns there are in a place – a room, a building or a town – the more it comes to life as an entirety, the more it glows, the more it has that self-maintaining fire which is the quality without a name.” (123)

Je höher die Qualität der einzelnen Tätigkeitsmuster im Commoning, desto weniger aufwendig können Bedürfnisse befriedigt werden, desto sinnvoller wird Commoning zur Herstellung und Erhaltung der gesellschaftlichen Lebensbedingungen. Oder anders herum ausgedrückt: Je niedriger die Qualität häufig verwendeter Tätigkeitsmuster im Commoning, desto schlechter ist tendenziell die Qualität der Resultate, desto aufwändiger die Re-Produktionsprozesse, desto weniger sinnvoll und tendenziell weniger befriedigend ist es für einzelne Personen sich dem Commoning anzunehmen. Die Selbsterhaltung und Verbreitung des Commonings durch eine hohe Effizienz zur Bedürfnisbefriedigung – „*that self-maintaining fire*“ – ist damit direkt abhängig von der Qualität der Tätigkeitsmuster.

ihnen durchgeführt werden können. Auch der *Aufwand* zur Erzeugung eines bestimmten Resultates muss innerhalb des integrierten Zusammenschlusses *neu bestimmt* werden. Damit geht einher, dass die Kopplung von Beteiligung und individuellem Vorteil nicht mehr auf rein individueller Ebene stattfindet, sondern auf Ebene des Zusammenschlusses gehoben und von hier auf die daran Beteiligten verteilt wird. Auf Tätigkeitsmuster direkt bezogene Softwarefunktionen, wie die *vorsorgende Selbstzuordnung* oder die individuelle Angabe von *Kontinuität*, werden mit deren Auflösung *funktionlos*. In der Struktur der Software ersetzt der integrierte Zusammenschluss also Tätigkeitsmuster, ähnelt diesen auch oberflächlich, wenngleich es – im Gegensatz dazu – nicht nur ein *Resultat* sondern *auch mehrere Resultate* hervorbringen kann. Jeder Zusammenschluss braucht eine *Bibliothek von möglichen Resultaten*, durch welche es in den Konfigurationsprozess integriert werden kann. Der *Bedarf an Mitteln* für ein bestimmtes Resultat ist nicht an Tätigkeitsmuster gebunden und muss dementsprechend durch die am Zusammenschluss Beteiligten veränderbar sein. Eine entsprechende optionale Beschreibung, warum bestimmte Mittel benötigt werden, kann dazu beitragen, sich als Zusammenschluss nach außen verständlich zu machen und so eventuell aufkommende Auseinandersetzungen besser lösen zu können und Misstrauen vorzubeugen.

Die Beteiligten an einem Zusammenschluss sind klar definiert – auf Softwareebene dabei nur, insofern sie auch im ununterbrochenen Commoning beteiligt sind, während der Zusammenschluss an sich größer sein könnte. Die Beteiligten an einem neu entstandenen Zusammenschluss *könnten* dabei genau die Personen sein, die zuvor die isolierten Tätigkeiten des damit ersetzten Zusammenhanges ausgeführt haben, allerdings muss das nicht der Fall sein. Wieder: Solche Zusammenschlüsse entstehen auf einer Ebene, die für die Software ungreifbar ist. Wie die direkten zwischenmenschlichen Absprachen zwischen den Beteiligten aussehen, wer zusätzlich zu einem Zusammenschluss hinzu kommt und wer Teil des Zusammenhangs der für sich stehenden Tätigkeiten war, sich bei einem Zusammenschluss aber zurückzieht usw., muss der Software übermittelt werden.

Dieser *Umbruch* von den Tätigkeitsmustern zu dem integrierten Zusammenschluss auf Softwareebene und der dafür notwendigen Definition der Beteiligten, braucht dabei die Zustimmung jeder davon betroffenen Person.

Ab dem Moment, in welchem ein Zusammenschluss definiert ist, liegt die Entscheidungsmacht darüber, wer zu diesem Zusammenschluss gehört und wer nicht bzw. wer aufgenommen bzw. ausgeschlossen wird, allein bei den Beteiligten dieses Zusammenschlusses. „Aufnahme und Ausschluss“ von Mitteln liegt dagegen nicht nur in der Entscheidungsmacht der daran Beteiligten: Diese Mittel des integrierten Zusammenschlusses *könnten* zwar ebenfalls wieder die Mittel sein, welche zuvor den einzelnen Tätigkeiten des Zusammenschlusses zugeordnet waren, allerdings ordnen sich Mittel – im Gegensatz zu Personen – nicht selbst zu und müssen daher auch für diejenigen, die nicht an diesem Zusammenschluss beteiligt sind, weiter für die eigenen Tätigkeiten *zuordenbar* bleiben. Dem integrierten Zusammenschluss müssen sich daher zwar Mittel frei zuordnen lassen, die Zuordnung muss dabei aber transparent bleiben. Diese Mittel *könnten* schließlich in die *Verantwortung und Organisation* des Commons gestellt werden, allerdings nur, wenn diese Funktion durch andere *anerkannt* wird, welche diese Mittel ebenfalls benötigen und das gleiche Recht zu deren Verwendung haben. Über vermittelnde Regeln kann die Verfügung über die Mittel innerhalb und außerhalb des Zusammenschlusses geklärt werden. Diese Verwendung der Mittel ist ein Grund, weshalb nicht nur die am inte-

THE GATE

"This quality in buildings and in towns cannot be made, but only generated, indirectly, by the ordinary actions of the people, just as a flower cannot be made, but only generated from the seed" (157)

Folgend geht es darum, wie ein Tätigkeitsmuster entsteht, wie sich Tätigkeitsmuster zu sinnvollen Prozessen der Bedürfnisbefriedigung verschließen lassen, wie Fähigkeiten durch Tätigkeitsmuster definiert werden und welche Bewegungstendenz durch die Anwendung qualitativ hochwertiger Muster innerhalb der Mustersprache selbst entsteht.

"When a thing is made, it has the will of the maker in it. But when it is generated, it is generated, freely, by the operation of egoless rules, acting on the reality of the situation, and giving birth, of their own accord..." (160)

Tätigkeitsmuster entstehen, wenn individuelle Erfahrungen gesellschaftlich verfügbar gemacht werden. Jemand nimmt sich einem Bedürfnis an, das ein anderer vermittelt hat, und zwar auf die Weise, die er gewohnt ist oder die er als sinnvoll erachtet. Bezeichnet die Person, welche das Bedürfnis vermittelt hat, das Bedürfnis anschließend als befriedigt, dann war die ausgeführte Tätigkeit zur Befriedigung *dieses* Bedürfnisses sinnvoll. Die Tätigkeit wird schließlich zum Tätigkeitsmuster, wenn der genaue Prozess und die dafür notwendigen Mittel in verarbeitbarer Form so genau beschrieben werden, dass es anderen ohne Vorkenntnis möglich wird, sie ebenfalls durchzuführen. Tritt *dasselbe* Bedürfnis dann an anderer Stelle wieder auf, kann den lokalen Anwendern und Anwenderinnen der Software dieses Tätigkeitsmuster als *eine Möglichkeit* zur Befriedigung des Bedürfnisses vorgeschlagen werden.

"But of course, autonomous creation of the parts, if taken by itself, will produce chaos – The parts will not form any larger whole, unless the individual adaption of the parts is under some sort of deeper regulation, which guarantees that the local process of adaption will not only make the local part truly adapted to its own processes, but that it will also be shaped to form a larger whole." (164-165)

Im Commoning gibt es keine Form der höheren Instanz, welche (komplexe) Re-Produktionsprozesse organisiert. Was dagegen auf Softwareebene möglich wird, ist sinnvolle Konfigurationen zu bestimmten Bedürfnisbefriedigungen im lokalen Kontext herauszustellen und der Qualität der Tätigkeitsmuster nach zu sortieren. Aber *bestimmte* Tätigkeiten an *bestimmten* Orten können nicht nur zu der Befriedigung eines *bestimmten* Bedürfnisses beitragen, sondern zu *verschiedenen*. Lokale Commoning-Prozesse können durch Tätigkeitsmuster verbunden sein, wenn an einer beliebigen Position innerhalb dieser Prozesse der Bedarf nach demselben Mittel ansteht. Es ändert sich damit nicht die Tätigkeit selbst, welche damit Teil verschiedener Prozesse zu verschiedenen Bedürfnisbefriedigungen ist, sondern – im Fall gegenständlicher Mittel – nur ihre Dauer.

nen der Angabe zur *Kontinuität von Tätigkeiten* und der Abspeicherung von *Konfigurationen bestimmter Gruppen* an.

„Fünfter Schritt: Aufweis der *Umstrukturierung und neuen Entwicklungsausrichtung des Gesamtsystems*, nachdem die *qualitativ spezifische Funktion* für die Systemerhaltung *bestimmend* geworden ist. Hier ist sowohl zu zeigen, welche *älteren Dimensionen im neuen Zusammenhang funktionslos* werden, als auch, wie sich die *Funktion früherer Dimensionen neu bestimmt*, und wie sich unter der *neuen Leitfunktion spezifische strukturelle und funktionale Differenzierungen* in der weiteren Entwicklung ergeben“ (Holzkamp: 80)

Welche *Umstrukturierungen* müssen zur Integration solcher Zusammenschlüsse an der Software vorgenommen werden, welche ihrer bisherigen Funktionen müssen *neu bestimmt* und welche *funktionslos* werden? Um sich dem anzunähern, müssen wir zurück zu den Anfängen der modernen Commons-Forschung und wenden uns hier Elinor Ostrom und ihren Design-Prinzipien für langlebige Commons-Institutionen zu. Durch eine knappe Kenntnis dieser über Jahrzehnte der Feldforschung erschlossenen Prinzipien für langlebige Commons soll herausgestellt werden, welche *„spezifischen strukturellen und funktionalen Differenzierungen“* auf Softwareebene vorgenommen werden müssen.

Holzkamp tritt ab.

Folgend die acht Prinzipien von Ostrom in Kürze.¹² Um die Verbindungen dieser Prinzipien zur Softwarekonzeption verständlicher zu machen, sind einige Begriffe dem Kontext nach abgeändert. Bei der ersten Verwendung der interpretierten Begriffe sind die von Ostrom verwendeten Begriffe in Klammern angehängt – deutsch, wenn die Übersetzung problemlos möglich scheint; englisch, wenn dem nicht so ist. Weiter wird dem strikt gesetzten Rahmen des ununterbrochenen Commonings treu geblieben und auch Institutionen staatlicher Macht (Behörden/Regierung) werden innerhalb dieses Rahmens interpretiert:

1. Es gibt klar definierte *Grenzen*, *wer* oder *was* zu einem Commons gehört. 2. Die Regeln der gemeinsamen Tätigkeit („Aneignung und Bereitstellung“) innerhalb eines Commons („an einer Ressource“) entsprechen den lokalen Bedingungen. Es gibt einen klaren Zusammenhang zwischen Aufwand und Nutzen („Aneignung und Bereitstellung“). 3. Die meisten *von der Tätigkeit eines Commons betroffenen* Personen sind autorisiert um an Entscheidungen zur Veränderung oder Erstellung von Regeln für das Commons teilzuhaben. 4. Einzelne am Commons beteiligte Personen sind verantwortlich, die Tätigkeiten der anderen am Commons Beteiligten und den Zustand der Mittel („Ressourcen“) zu *überprüfen*. 5. Wenn gegen die festgelegten Regeln verstoßen wird, führt es zu Sanktionen, die sich durch die Häufigkeit der Regelverletzungen verstärken. 6. Es gibt schnelle und wenig aufwendige („kostengünstige“) *Konfliktlösungsmechanismen* und Räume, an denen diese Konflikte zwischen an Commons Beteiligten und solchen, die nicht am Commons beteiligt sind („Behörden“) gelöst werden können. 7. Das Recht der Beteiligten sich ihre eigenen Regeln machen zu können, wird von denjenigen, die in aktiven Zusammenhang mit im Commons verwendeten Mitteln stehen („Regierung“) *anerkannt*. 8. Wenn die in einem Commons verwendeten Mittel („common-pool resources“) Teil einer größeren, qualitativ andersartigen Vermittlungsstruktur sind („connected to a larger social-ecological system“) gibt es vermittelnde Regeln zur Verfügung

der *Fähigkeiten* anzunähern, muss die Mustersprache, welche jede Person *in ihrem eigenen Kopf hat*, von einer *gesellschaftlich geteilten* Mustersprache unterschieden werden.

“Every person has a pattern language in his mind – Your pattern language is the sum total of *your* knowledge of how to build. The pattern language in your mind is slightly different from the language in the next person's mind; no two are exactly alike; yet many patterns, and fragments of pattern languages, are also shared.“ (202-203)

Christopher Alexander unterscheidet also zwischen einer *individuellen Mustersprache* und einer *geteilten Mustersprache*. Jedes geteilte Muster entstammt dabei einem individuellen Muster, sprich, einer Erfahrung. Erst wenn etwas für sich selbst erkannt wurde, kann das Wissen darüber auch geteilt werden. Im ununterbrochenen Commoning geschieht die gesellschaftliche Verfügbarmachung individueller Erfahrung in Form von Tätigkeitsmustern, welche von da an in unterschiedliche Commoning-Prozesse integriert werden können.

Durch diese Grenze zwischen individuellen und geteilten Tätigkeitsmustern lässt sich der Begriff der *Fähigkeit* neu denken. Die These ist: Wenn jedes Werkzeug menschliche Handlungsfähigkeit zur Lösung eines bestimmten Problems erweitert – die Maschinerie sogar weitgehend von körperlichen und geistigen Vermögen eines einzelnen Menschen unabhängig wurde –, dann kann jede einzelne Person prinzipiell alles bisher menschlich Geschaffte völlig ohne Vorkenntnisse ebenfalls vollbringen, wenn sie nur Zugriff auf die dafür notwendigen Mittel (d.h. auch Werkzeuge/Maschinen) hat und ihr jeder einzelne Schritt genau erklärt wird. Die Person kann dabei umso effizienter tätig sein, je weniger Tätigkeiten ihr Schritt für Schritt erklärt werden müssen, sprich, je mehr für den Prozess notwendige Tätigkeiten sie *verinnerlicht* hat. Wenn sie mehrere in Zusammenhang stehende Tätigkeiten verinnerlicht hat, kann eine Tätigkeitsbeschreibung darauf aufbauen und einzelne Schritte als bekannt voraussetzen. In Tätigkeitsmustern formuliert, können einzelne, aber innerhalb eines Commoning-Prozesses zusammenhängende Tätigkeitsmuster als ein Einzelnes formuliert werden, dessen Anforderung die verinnerlichte Kenntnis der darin eingeordneten Tätigkeitsmuster voraussetzt. [SWK7]

Über die Software vermittelt, wird also individuelle Erfahrung geteilt und geteilte Erfahrung individuell verinnerlicht. Weiter werden daher auf Softwareebene Fähigkeiten als *verinnerlichte Tätigkeitsmuster* bzw. als Komplex von verinnerlichten Tätigkeitsmustern gedacht.

Diese Vorgehensweise ermöglicht

1. eine Definition von Fähigkeiten durch Tätigkeitsmuster im Rahmen der Softwarestruktur und
2. eine beliebige Skalierung von Tätigkeitsmustern durch die Definition der dafür notwendigen Fähigkeiten.

Die Folge daraus ist, dass sich sämtliche in Zusammenhang stehende Tätigkeiten durch ein einziges *komplexes Tätigkeitsmuster* beschrieben werden können und umgekehrt, dass jedes komplexe Tätigkeitsmuster wieder in eine Reihe anderer Tätigkeitsmuster mit geringeren Anforderungen aufgelöst werden kann.

eigentlich erweitern soll, wieder eingeschränkt wird. Und gleichzeitig ist jede beteiligte Person dem Umstand ausgesetzt, dass andere, von deren Tätigkeit die eigene Tätigkeit abhängig ist, dieser *nicht* bzw. nur auf *problematischer Weise* nachgehen. Diese problematische Weise kann eine unzureichende sinnlich-funktionale Qualität der Resultate oder etwa eine stärkere zeitliche Verzögerung sein. Treten mehrere solcher *Störungen* innerhalb des Commonings zeitnah auf und beziehen sich viele verschiedene einzelne Commoning-Prozesse auf dieselben Tätigkeiten, denen unerwartet nicht mehr nachgegangen wird, kann es möglich sein, dass schon wenige solcher Störungen weite Teile des ununterbrochenen Commonings so stark beeinträchtigen, das (existenzielle) Bedürfnisse nicht befriedigt werden und von einer *Krise* gesprochen werden kann.

Eine erste Softwarefunktion, um diese Störungen bzw. Krisen abzumildern, kann eine *halbautomatische Zeitplanung* sein. Durch eine Kenntnis der ungefähren Dauer von bestimmten Tätigkeiten und der Zeiträume, denen sich die beteiligten Personen für das Commoning nehmen wollen, können über die Software sinnvolle Zeitpläne *vorgeschlagen* werden. Eine solche Funktion kann die Zeitplanung für die Beteiligten erleichtern, aber den Ausfall einzelner Tätigkeiten (bzw. ihre problematische Ausführung) nicht verhindern. Eine zweite Funktion zur Abmilderung dieser Störungen bzw. Krisen, kann daher die *vorsorgende Selbstzuordnung* zu Tätigkeiten sein, die entweder bereits ausgeführt werden oder zu denen sich bereits andere Personen zugeordnet haben. Falls eine Person damit einer geplanten Tätigkeit nicht nachgeht, kann die sich der aktivierten Tätigkeit als nächstes zugeordnete Person damit angefragt werden.

Die Software sollte solche Funktionen anbieten. Beide Funktionen unterstützen allerdings nicht die Entwicklung einer von der Software weitgehend unabhängigen Form der Selbstorganisation, sondern bewegen sich weiterhin innerhalb des Rahmens der Softwarevermittlung.

„Dritter Schritt: Aufweis des Funktionswechsel der (im ersten Schritt) aufgewiesenen relevanten Dimensionen als ›Organismus-Pol‹ des Entwicklungswiderspruchs, damit der Entstehung des ersten qualitativen Sprungs der Herausbildung der Spezifik der neuen Funktion unter den veränderten Außenweltbedingungen.“ (Holzkamp: 79)

Verbindliche Absprachen und direkte Konfliktklärung zwischen den Personen, die bei der Ausübung ihrer Tätigkeiten in direkter Verbindung zueinander stehen, ist die andere Möglichkeit, solche Störungen bzw. Krisen vorzubeugen. Diese Form der direkten zwischenmenschlichen Beziehungen ermöglichen es überhaupt, die Bedürfnisse, Besonderheiten und Lebensumstände der anderen in das eigene Handeln einzubeziehen und aufkommende Konflikte schnell und unkompliziert lösen zu können. Anstehende und damit den Tätigkeiten kontinuierlich zweckgebende Bedürfnisse vorausgesetzt, lässt sich spekulativ sagen, dass die Tätigkeiten der Personen tendenziell länger störungsfrei ausgeführt werden können, welche direkte zwischenmenschliche Beziehungen in Form verbindlicher Absprachen und direkter Konfliktklärung eingehen.

Der „erste qualitative Sprung“ sind solche direkt zwischenmenschlichen Absprachen und Konfliktlösungen, die der Erhaltung der über die Software hervorgebrachten Struktur dienen. Da es aber auf einer Ebene geschieht, welche die Softwarevermittlung selbst nicht betrifft, kann über die Software als unter-

tigkeit im Allgemeinen ist und welchen Aufwand die Verfügbarmachung der notwendigen Mittel (spekulativ) mit sich bringt.

Der *Kontext* eines Tätigkeitsmusters ist der *Raum*, in welchem das Tätigkeitsmuster auftritt. Der Raum besteht aus Spannungen zwischen unbefriedigten Bedürfnissen, den zur Befriedigung potentiell verfügbaren Mitteln und möglichen Tätigkeiten zur Verfügbarmachung notwendiger Mittel.

“For although the system of forces in a situation is very hard to define analytically, it is possible to tell, in a holistic way, whether the pattern is alive or not. - The fact is that we feel good in the presence of pattern which resolves its forces. - And we feel ill at ease, uncomfortable, when a pattern leaves its forces unresolved” (286)

Ein Vorteil der Softwarevermittlung ist, dass die Qualität eines Tätigkeitsmusters in einem bestimmten Kontext nicht gefühlt werden muss, sondern sehr eindeutig definiert werden kann. Voraussetzung dafür ist allerdings, dass erstens die Meta-Daten der Mittel (Lokalität, Menge, Verfügbarkeit, etc.) aktuell gehalten werden und die Software Zugriff auf diese Datenbank hat, zweitens die Sprache der Tätigkeitsmuster weitgehend vollständig ist, also mögliche Lösungen zu auftretenden Problemen durch eine Konfiguration von Tätigkeitsmustern beschrieben werden können und drittens der Aufwand jeder Tätigkeit möglichst klar definiert wurde.

In diesem Netz aus möglichen und aktiven Commoning-Prozessen ist eine Tätigkeit ideal, wenn ihr Resultat auf möglichst viele Bedarfe verweist und möglichst viele ihrer Bedarfe durch (lokal) zur Verfügung stehende Mittel mit möglichst wenig Aufwand bereit gestellt werden können.

“By seeming to be unethical, by making no judgments about individual opinions, or goals, or values, the pattern rises to another level of morality. [...] And, when we succeed, finally, in seeing so deep into a man made pattern, that is no longer clouded by opinions or by images, then we have discovered a piece of nature as valid, as eternal, as the ripples in the surface of a pond” (304)

Ein Tätigkeitsmuster mit hoher Qualität kann dem widersprechen, was von einer Person als richtig und sinnvoll angesehen wird. Doch wenn die Software funktional ist, dann bedeutet eine hohe Qualität eines Tätigkeitsmusters eindeutig, dass die entsprechende Tätigkeit am *effizientesten* zur Befriedigung von Bedürfnissen ist, welche sämtliche Lebensbereiche umfassen können. Die Qualität eines Tätigkeitsmusters ist damit der rote Faden zu einer Gesellschaft nach Bedürfnissen und Fähigkeiten, von welcher kein Mensch bisher weiß, wie sie *aussehen* und wie sie sich *anfühlen* wird. Die Realisierung dieser Gesellschaft kann damit den eigenen *Bildern* davon und den eigenen *Idealen* widersprechen. Die Verantwortung, diesem roten Faden zu folgen, liegt allerdings nie bei Einzelnen. Über die Software werden Personen niemals *direkt* angesprochen: Jedes Tätigkeitsmuster ist ein Vorschlag für jede befähigte Person sich dieser anzunehmen – ob sich eine Person dann einem qualitativ hochwertigen Tätigkeitsmuster annimmt oder nicht, liegt alleine in ihrem eigenen Ermessen. Relevant ist nur, dass es überhaupt einen *Indikator* dafür gibt, was sinnvoll zur Herstellung und Erhaltung dieser ersehnten Gesellschaft ist. IH/B⁹

nung dafür freigegeben wird, ist nicht relevant. Beides sollte über die Software möglich sein.

An dieser Stelle sind Momente der *von der Software unabhängigen Selbstorganisation* in Form von Absprachen zwischen konkreten Personen nicht nur sinnvoll, sondern unbedingt vorausgesetzt. Erste Formen von *Konfliktlösungsmechanismen* auf Softwareebene können in diesem Prozess hilfreich sein. Eine *Analysefunktion*, in welcher etwa ausgewertet werden kann, welche Konfigurationen unter Beibehaltung bestimmter Tätigkeitsmuster möglich wären oder welche Veränderungen die Integration eines neuen Musters nach sich zieht, kann diesen Prozess außerdem unterstützen.

"In this framework, we gain an entirely new view of the process through which a sequence of acts of building generates a whole" (486)

Wenn wir solche Prozesse als sinnvolle Handlungsmöglichkeiten ansehen, wie sie durch die Softwarevermittlung entstehen können, dann beobachten wir an dieser Stelle, wie *nicht-Software-vermittelte Selbstorganisation* in Teilbereichen des Gesamtprozesses immer bedeutender wird. Die Frage ist, wohin wir gelangen, wenn diese in diesem Kontext neue Form der Selbstorganisation für einzelne am Commoning Beteiligte *bestimmend* wird. Und durch die Beantwortung der Frage müssen Rückschlüsse auf die Software genommen werden, um diese mögliche Entwicklung unterstützen zu können, ohne, dass diese Formen der Selbstorganisation aus dem ununterbrochenen Commoning herausfallen. Diese Form der Selbstorganisation, welche von der Softwarevermittlung weitgehend unabhängig ist, trotzdem aber Teil der generellen Bedürfnisbefriedigung bleibt, wird folgend als eine *neue Qualität* begriffen.

"This is commonplace, in the growth of an organism, where all the larger patterns are generated, merely as the end products of tiny, daily transformations. - At any given moment, in a growing organism, there is no sense of the "end" or of the final "goal" of growth. There is, instead, a process of transformation, which is able to take the present state of the organism, and move it slightly, in the next minute of growth - in such a way that when the same process is then repeated in the minute after that, and in the minute after that, slowly, inexorably, the necessary patterns come into being - not according to some plan, but as the product of a sequence of transforming steps." (500)

Auch bei Alexander gibt es einen qualitativen Umbruch. Er entsteht durch die schrittweise Integration von Mustern in die Ganzheit des Designs, wodurch *größere Muster* als Produkt dieser kleinen transformativen Schritte („transforming steps“) entstehen. Dagegen gibt es keine „größeren Tätigkeitsmuster“. Es gibt durch Fähigkeiten höher skalierte *komplexe Tätigkeitsmuster*, die allerdings nur eine Bündelung einzelner Tätigkeitsmuster sind und keine neue Qualität im Sinne einer anderen Form der Selbstorganisation darstellen. Um die neue Form der Selbstorganisation mit entsprechenden Softwarefunktionen unterstützen zu können, muss die Transformation dorthin möglichst exakt beschrieben werden. Und Christopher Alexander, dessen Mustersprache *nicht aus Pro-*

Eine Benachrichtigung kann sowohl sehr früh erfolgen, wenn sich der Tätigkeit gerne angenommen wird bzw. erst im Notfall, wenn eine Dringlichkeit für eine bestimmte Tätigkeit besteht, welcher sich zumindest angenommen werden kann.

"Just so, a common pattern language is defined by a pool of patterns. - Suppose that every person in society has his own personal pattern language. Now imagine the collection of all the patterns which anyone has in his language. Call this collection of patterns the pattern pool. Some of the patterns will occur much more often in the pattern pool than others. [...] The common pattern language is not any one language which any one person has in his mind - it is defined by the overall distribution of patterns in the pattern pool." (343)

In die eigene Bibliothek werden Tätigkeitsmuster hinzugefügt, wenn sich ihnen erfolgreich angenommen wurde. Sie können aber auch daraus entfernt bzw. deaktiviert werden, wenn die entsprechende Person diese nicht mehr als sinnvoll ansieht oder das Interesse an der darin beschriebenen Tätigkeit verliert. Werden schließlich alle einzelnen Bibliotheken betrachtet, dann stellen sich Tätigkeitsmuster heraus, die insgesamt besonders häufig in den Bibliotheken vorhanden sind und solche, die es nur in einigen wenigen gibt. Besonders häufige Tätigkeitsmuster können dabei als gesamtgesellschaftlicher Konsens betrachtet werden - sie erscheinen so sinnvoll, dass sie möglicherweise sogar weltweit auf gleiche Weise genutzt werden, um bestimmte Probleme zu lösen. Andere Tätigkeitsmuster werden besonders häufig in lokalen oder virtuellen Communities und Interessensgemeinschaften verwendet und können eine bestimmte Lebensform oder sogar Lebenseinstellung ausdrücken. Ein bestimmtes Set dieser Tätigkeitsmuster ist damit wie ein Dialekt, der auf eine bestimmte Prägung oder Herkunft hinweist, aber trotzdem Teil derselben Sprache ist. Und wieder andere Tätigkeitsmuster werden nur von sehr wenigen Personen verwendet; sie bilden damit Eigenheiten dieser Personen ab, sind aber immer noch Teil des Ganzen, des ununterbrochenen Kooperationsprozesses.

"And, once people share a language in this way, the language will begin evolving of its own accord. - [...] As good patterns get shared more widely, and bad patterns die out, the pattern pool will gradually contain more and more good patterns [...] - in this sense, even though any one person's pattern language will always be unique, the overall collection of languages in society will gradually drift towards a common language, represented by the overall character of the pool of patterns." (344)

Im ununterbrochenen Commoning gibt es eine Bewegungstendenz zu einer immer geringeren Zahl häufig angewandter Tätigkeitsmuster mit gleichzeitig einer immer höheren Qualität. Wie folgend dargestellt, entsteht diese durch die *Kontinuität* qualitativ hochwertiger Tätigkeiten:

Voraussetzung zur Kontinuität ist, dass das Resultat der Tätigkeit in möglichst vielen lokalen Commoning-Prozessen angewendet werden kann, also tendenziell *vielseitig* einsetzbar ist. Die Kontinuität einer Tätigkeit zieht dabei tendenziell eine Kontinuität anderer Tätigkeiten nach sich, welche einerseits durch die

Teil anderer Commoning-Prozesse sind. Als Softwarefunktion müssen Teilbereiche der Konfiguration als feste Zusammenhänge von sowohl den Tätigkeiten als auch sich ihnen zugeordneten Personen gespeichert werden können, damit bei Wiedereintritt desselben Bedürfnisses dieselben Personen die Möglichkeit haben, sich diesen wieder anzunehmen und den Prozess der Selbstzuordnung damit zu überspringen. Die Funktion ist damit eine Erweiterung der Angabe von *Kontinuität* auf Gruppenebene und kann genauso eine *vorsorgende Reservierung von Mitteln* beinhalten. Eine solche Sicherung von Zusammenhängen muss zwischen den beteiligten Personen abgesprochen sein und benötigt die Zustimmung jeder beteiligten Person. Im Konfigurationsprozess kann ein solcher Zusammenhang benachrichtigt werden, sobald Tätigkeitsmuster für das „höchste“ Resultat des Zusammenhangs freigeschaltet werden.

"No building is ever perfect. - Each building, when it is first built, is an attempt to make a self-maintaining whole configuration. - But our predictions are invariably wrong. People use buildings differently from the way they thought they would. And the larger the pieces become, the more serious this is. [...] - It is therefore necessary to keep changing the buildings according to the real events which actually happen there." (479-480)

Die durch Selbstauswahl aktivierten Commoning-Prozesse sind aus unterschiedlichen Gründen nicht ideal. So ist es etwa problematisch, dass im Konfigurationsprozess in zeitlichen Abständen immer mehr, aber damit auch immer qualitativ niedrigere Tätigkeitsmuster vorgeschlagen werden und es damit auch dem Zufall überlassen ist, wann sich jemand zuordnet und damit die weitere Richtung des Prozesses bestimmt. Genauso können sich Zusammenhänge von Tätigkeitsmustern, die rein auf Softwareebene sinnvoll erscheinen, als unpraktisch in der konkreten Anwendung herausstellen. Möglich kann genauso sein, dass Personen nicht zuverlässig sind oder sich etwa selbst überschätzt haben und dadurch der Prozess nicht reibungslos vonstatten geht. Oder aber: Eine Tätigkeit fügt sich zwar sehr gut in einen bestimmten Commoning-Prozess ein, allerdings kann die Verwendung eines anderen, aber ähnlichen Tätigkeitsmusters, dazu führen, dass sie sich besser in den Gesamtprozess integriert.

"It is therefore necessary to keep changing the buildings, according to the real events which actually happen there - And the larger the complex of buildings, neighborhood, or town, the more essential it is for it to be built up gradually, from thousands of acts, self-correcting acts, each one improving and repairing the acts of the others." (480)

Der *Konfigurationsprozess* durch Selbstzuordnung ist eine Möglichkeit, wie ein Commoning-Prozess über die Software vermittelt geformt werden kann, der *Reparaturprozess* eine zweite. Diese Zweite baut auf existierenden Commoning-Prozessen auf und lässt die daran Beteiligten diese Prozesse manuell ihren Vorstellungen nach verändern.

"Yet, changing as it is, each language is a living picture of a culture, and a way of life - The patterns it contains, widely shared, reflect a common understanding about attitudes to life, about the ways that people want to live, the way they want to rear their children, the way they want to eat their meals, the way they want to live in families, the way they want to move from place to place, the way they work, the way they make their buildings look towards the light, their feelings about water, above all, their attitudes towards themselves." (347)

Indem wir die verwendeten Tätigkeitsmuster betrachten - von einzelnen Personen, von Communities, von der gesamten Gesellschaft - erkennen wir ihre Einstellung dem Leben gegenüber. Und je zwangloser Menschen in einer Gesellschaft tätig sind - je weniger Macht Geld über sie hat und je weniger Abhängigkeit von konkreten anderen Personen besteht -, desto deutlicher zeigt sich, was diesen Menschen, was *uns* als Menschen, *wirklich* wichtig ist, wie wir leben wollen, was uns gut tut. Und eine geschlossene Sprache aus Tätigkeitsmustern eröffnet den Weg zu dieser Gesellschaft.

"In early times the city itself was intended as an image of the universe - its form a guarantee of the connection between the heavens and the earth, a picture of a whole and coherent way of life. A living pattern language is even more. It shows each person his connection to the world in terms so powerful that he can re-affirm it daily by using it to create new life in all the places round about him. - And in this sense, finally, as we shall see, the living language is a gate." (348)

können, sprich: ein höheres Gewicht haben. Und da Tätigkeiten nur auf Grund ihrer Qualität *freigeschaltet* werden und sich erst ab diesem Moment ein mögliches *Gewicht* ergibt, ist es sinnvoll, Tätigkeiten für die Beteiligten nicht ihrer Qualität, sondern ihrem Gewicht nach zu sortieren. Neben den eigenen Fähigkeiten und Interessen, welche in der jeweils eigenen Bibliothek entsprechend angegeben werden, ist damit das Gewicht der Tätigkeiten das zweite zentrale Kriterium lokal-mögliche Tätigkeiten personenbezogen vorzuschlagen und eine Möglichkeit für die Anwender und Anwenderinnen, den Gesamtprozess gezielt zu durchsuchen.

"But you cannot create a pattern at full intensity, so long you are worrying and thinking about other patterns, which you will have to deal with later in the sequence. - [...] When you start to think about compromises between patterns, you are not taking account of the fact that every pattern is a rule of transformation. The fact that every pattern is a rule of transformation means that each pattern has the power to transform any configuration by injecting a new configuration into it, without essentially disturbing any essentials of the configuration which was there before." (399-400)

Wenn sich zu allen zu einer bestimmten Bedürfnisbefriedigung notwendigen Tätigkeiten Personen zugeordnet haben, kann durch deren Einwilligung ein Commoning-Prozess *aktiviert* werden. Ab diesem Moment wird nicht nur klar ersichtlich, wie der Kooperations-Prozess zu einer bestimmten Bedürfnisbefriedigung ablaufen wird, sondern es entsteht auch eine plötzliche Verbindung zwischen konkreten, einander möglicherweise unbekanntenen Personen. Der Prozess zur Bedürfnisbefriedigung ist nicht mehr oder weniger als das, was diese Personen konkret machen, in der Reihenfolge, dass die eine Person der nächsten bereit stellt, was diese benötigt, um selbst tätig zu werden, bis schließlich das Bedürfnis befriedigt ist. Über diese Tätigkeit in ihrer Ausführung wird damit die Welt einem bestimmten Muster nach verändert („*rule of transformation*“). Im Fall von *Kontinuität* können dabei erste Ansätze von *Gewohnheit* entstehen und dadurch Strukturen zugelassen werden, in denen direkte zwischenmenschliche Beziehungen im bis dahin nur über die Software vermittelten Commoning dauerhaft sinnvoll werden und sich von dort an auf die softwarevermittelte Struktur zurückwirken können.

"And we see then, how a group of people can design a complex building – Once they agree about the language, the actual emergence of the form is simple and fluid. When a group of people try to do something together, they usually fail, because their assumptions are different at every stage. But with a language, the assumptions are almost completely explicit from the start. - Of course they no longer have the medium of a single mind, as an individual person does. But instead, the group uses the site "out there in front of them", as the medium in which the design takes its shape." (449)

[SWK⁴¹], ist allerdings auf gesamtgesellschaftlicher Ebene autoritär und widerspricht dem elementaren Prinzip des Commonings; der Selbstorganisation.

Bei Alexander sind die architektonischen Muster unterschiedlicher Größe so angelegt, dass sie sich in ihrer konkreten Ausführung durch die umgebenden Muster verändern, ihre Essenz dabei aber nicht verlieren. Die im ununterbrochenen Commoning verwendeten Tätigkeitsmuster sind dagegen weniger flexibel. Durch das Resultat und die dafür notwendigen Mittel ist jedes Tätigkeitsmuster klar definiert und damit in dieser Hinsicht unveränderbar. Ihre Anpassungsfähigkeit an den jeweiligen Kontext erhalten die Tätigkeitsmuster dabei nicht durch die Wechselwirkung mit anderen Mustern, sondern durch ihre reine Vielzahl. Im ununterbrochenen Commoning entsteht dabei die grundlegende *Konfiguration* der Tätigkeitsmuster nicht im Kopf einer einzelnen Person, sondern durch die *Selbstzuordnung* einzelner Beteiligter zu den Tätigkeiten, welche im Zusammenhang ausschließlich den Commoning-Prozess bestimmen. Welche Tätigkeitsmuster ausgewählt werden und wer sich ihnen zuordnen wird, ist damit nicht durch einzelnen Personen bestimmbar oder vorhersehbar.

Die Frage ist allerdings, wie Selbstzuordnung sinnvoll ablaufen kann, damit sich daraus qualitativ hochwertige Commoning-Prozesse ergeben. Für die Befriedigung eines jeden Bedürfnisses gibt es potentiell unzählige Möglichkeiten – eben genauso viele Möglichkeiten, wie dieses bestimmte Bedürfnis jemals aus individueller Erfahrung heraus befriedigt wurde und jemand anschließend diese Möglichkeit, als Tätigkeitsmuster beschrieben, in die entsprechende Datenbank eingespeist hat. Und wie jedes Bedürfnis durch unterschiedliche Tätigkeiten befriedigt werden kann, kann auch jeder einzelne Bedarf dafür durch unterschiedliche Möglichkeiten gedeckt werden, genauso wie jeder zur Bedarfsdeckung notwendige Bedarf wieder auf unterschiedliche Weisen gedeckt werden kann usw. usf. Von dem vermittelten Bedürfnis aus ergibt sich damit eine sich immer weiter aufgabelnde Kette von zusammenhängenden Möglichkeiten zu seiner Befriedigung. Und keine dieser Tätigkeiten muss ausgeführt werden, wenn es auch eine Alternative mit demselben Resultat gibt bzw. es eine Alternative zu der vorherigen Tätigkeit gibt usw. usf.

"The sequence of the patterns for a design – as generated by the language – is therefore the key to that design. - For once you find the proper sequence, the power to design coherent things follows from it almost automatically [...] We can take patterns, step by step, one at a time, because each pattern moulds the whole – and each pattern can mould the whole which is the product of the previous patternings." (382-383)

Die ideale Konfiguration eines Commoning-Prozesses zeichnet sich durch die insgesamt höchste Qualität der darin enthaltenen Tätigkeitsmuster aus. Je vollständiger die Mustersprache ist – also je mehr mögliche Probleme damit gelöst werden können und je mehr mögliche Lösungen (mit unterschiedlichen dafür notwendigen Mitteln) für jedes Problem es darin gibt – desto idealer kann eine solchen Konfiguration ausfallen. Eine ideale Konfiguration kann dabei durch die Software im jeweiligen lokalen Kontext herausgestellt werden. Das heißt allerdings noch nicht, dass sie auch auf diese Weise realisiert wird – es braucht noch Personen, die sich den einzelnen Tätigkeiten annehmen, sprich, sich ihnen *zuordnen*. Und da sich Tätigkeiten der Bedarfsdeckung erst ergeben, *nachdem* eine Tätigkeit für ein bestimmtes Resultat festgelegt wurde, muss diese Selbstzuordnung Schritt für Schritt geschehen. Das erste Tätig-