

# The Timeless Way of Re-Production

Die Entwicklung gesellschaftlicher  
Selbstorganisation



...C...

Marcus Meindel



*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 Nahradas<sup>1</sup>, 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.*

## THE TIMELESS WAY

*„There is one timeless way of building. [...] It is so powerful and fundamental that with its help you can make any building in the world as beautiful as any place that you have ever seen. It is so powerful, that with its help hundreds of people together can create a town, which is alive and vibrant, peaceful and relaxed, a town as beautiful as any town in history. And there is no other way in which a building or a town which lives can possibly be made.“ (7-8)*

Die Eröffnungssätze aus „The Timeless Way of Building“ verdeutlichen sehr klar, dass Christopher Alexander absolut überzeugt von dem historischen Wert seiner Entdeckung ist. Dass jemand von seinen eigenen Ideen überzeugt ist und sie für weltgeschichtlich bedeutend hält, ist dabei nichts Besonderes. Aber als These angenommen, dass Alexander mit seiner Einschätzung Recht hat – vielleicht sogar damit, dass es der *einzigste Weg* ist –, und weiter angenommen, dass diese vorliegende Interpretation auf die Re-Produktionsweise nach Commons-Prinzipien sinnvoll ist; dann stünde uns die Gesellschaft nach Bedürfnissen und Fähigkeiten als konkrete Möglichkeit gegenüber, dann müssten wir sie „nur noch“ realisieren. Und auch wenn dieses „nur noch“ in Anführungszeichen steht, hätten wir statt einer vagen Hoffnung ein konkretes Ziel und eine konkrete Handlungsmöglichkeit, mit welcher diese neue Gesellschaftsform aus der Alten heraus entstehen kann. Und die Software selbst, zu deren Konzeption diese Interpretation dient, ist ein Werkzeug zu diesem Zweck.

Noch zwei Anmerkungen zu diesem ersten Zitat:

1. Wenn Alexander über Architektur redet, wird hier Commoning verstanden. Das erscheint erst einmal ungewohnt. Der Gedanke dazu ist, dass Alexander zwar Architekt ist, aber die Architektur nur der ihm naheliegende Gegenstand zur Anwendung einer allgemeingültigen Methode war. Die Methode selbst wird allerdings erst nach dieser Interpretation, im Anhang, erläutert.
2. Wenn auch die Analyse des Bestehenden auf das Commoning für die Sache der Transformation unbedingt notwendig ist, wird in diesem Text ausschließlich die Struktur des ununterbrochenen Commonings betrachtet. Die Abstraktion geht so weit, dass auch von bestehenden Eigentumsverhältnissen abgesehen wird und sämtliche Mittel als allgemein verfügbar und als dem Zweck der Bedürfnisbefriedigung untergeordnet verstanden werden.

*„This one way of building has always existed. But it has become possible to identify it, only now, by going to a level of analysis which is deep enough to show what is invariant in all the different versions of this way.“ (10-11)*

Commoning ist das miteinander-Teilen, das Beitragen, das Konflikte-auf-Augenhöhe-klären, das füreinander-Dasein. Es ist nichts Neues, ganz im Gegenteil: Die Kooperation zur Herstellung und Erhaltung der gemeinsamen Lebensbedingungen ist das, was uns als Spezies seit jeher ausmacht, aber in einer un-

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)*

Zuletzt zeigt sich ein fundamentaler Unterschied zwischen dem über die Software vermittelten Commonings und Alexanders Vorstellung einer Utopie: Diese auf dem technischen Fortschritt basierende Form des Commonings ermöglicht komplexe Strukturen zur generellen Bedürfnisbefriedigung und ermöglicht darin auch Räume der Selbstorganisation, welche Teil des Gesamtprozesses bleiben, in denen die Vermittlung über die Software allerdings nicht bestimmend ist. Die gesamtgesellschaftliche Koordination beruht allerdings auf dem Medium selbst, kann also nicht „erlernt“ und das Werkzeug kann nicht abgeworfen werden („*shedding it*“). Sobald keine Werkzeuge wie die hier beschriebene Software mehr verwendet werden, verliert sich auch die gesellschaftliche Transparenz. Und mit ihr nehmen die Möglichkeiten ab, wie sich in die gesellschaftliche Struktur eingebracht werden kann. Und mit ihr nimmt die Anzahl an Menschen ab, die sich prinzipiell den eigenen Bedürfnissen annehmen können und immer mehr Bedürfnisse bleiben dadurch wieder unbefriedigt. Ein wesentlicher Unterschied in der Vision – aber irrelevant für die Vorgehensweise an sich.

## THE QUALITY

*“We have been taught that there is no objective difference between good buildings and bad, good towns and bad. - The fact is that the difference between a good building and a bad building, between a good town and a bad town, is an objective matter. It is the difference between health and sickness, wholeness and dividedness, self-maintenance and self-destruction. In a world which is healthy, whole, alive, and self-maintaining, people themselves can be alive and self-creating. In a world which is unwhole and self-destroying, people cannot be alive: they will inevitably themselves be self-destroying, and miserable. - But it is easy to understand why people believe so firmly that there is no single, solid basis for the difference between good building and bad. - It happens because the single central quality which makes the difference cannot be named” (25)*

Christopher Alexander macht sich folgend auf die Suche nach dieser *Qualität ohne Namen*, wie Marx sich auf die Suche nach dem *Wert* macht. Und wie Marx den Wert in den Dingen sucht, die nicht für einen selbst, sondern für den Markt und damit für andere produziert werden, sucht Alexander die *Qualität* als Essenz von Orten, an denen sich Menschen lebendig fühlen. Für beide ist schließlich das Resultat ihrer Suche der Ausgangspunkt, von welchem aus sie ein vollständiges System beschreiben können. Doch während der Wert bei Marx die Ware als einen äußeren Träger hat, ist die *Qualität ohne Namen* bei Alexander ein innerer Zustand.

*"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 space, and out of nothing else: they are the atoms and the molecules from which a building or a town is made.” (75)*

Bei Alexander werden Muster von Ereignissen („*pattern of events*“) ins Verhältnis mit dem sie umgebenden Mustern des Raums („*patterns of space*“) gesetzt – im Commoning dagegen die Mustern von Tätigkeiten ins Verhältnis mit den dafür angewandten und daraus hervorgehenden *Mitteln* [Muster von Mitteln: SWK<sup>2</sup>]. Diese Mittel können dabei sowohl gegenständlicher, symbolischer oder sozialer Natur sein<sup>3</sup>: Von den Wohnblöcken bis zum Acker, von den Büchern bis zu den Smartphones, von der parlamentarischen Demokratie bis zum Konsensverfahren.

Die Mittel bilden dabei den *Rahmen* eines Tätigkeitsmusters: Der *Bedarf* einer Tätigkeit sind die Mittel, welche zur Ausführung einer Tätigkeit notwendig sind. Das *Resultat* ist das Mittel, welches durch die Tätigkeit entsteht. Einen zweiten Fall bilden Tätigkeiten, welche kein Mittel als Resultat haben, sondern ein Bedürfnis direkt befriedigen.

Der Zusammenhang zwischen den Mustern wird dabei in den *Tätigkeitsmustern* durchaus anders aufgefasst als bei Alexander. Bei Alexander *ist* jedes Muster Teil eines größeren Musters und *verweist* auf kleinere Muster. Zum Beispiel das Muster „LIGHTS ON TWO SIDES OF EVERY ROOM“: Das Muster selbst bleibt erhalten, während es aber Teil der *vorhergehenden* Muster „WINGS OF LIGHT“, „LONG THIN HOUSE“ und „CASCADE OF ROOFS“ sein kann, über welche überhaupt erst der Grundriss eines Gebäudes ermöglicht wird, damit in jedes Zimmer Licht von zwei Seiten fallen kann. Das Muster verweist schließlich (u.a.) auf die *nachfolgenden* Muster „WINDOW OVERLOOKING LIFE“ und „NATURAL DOORS AND WINDOWS“. Beide Muster betreffen den Ausblick aus den beiden Lichtquellen, womit sich selbstverständlich ihrem Vorhandensein selbst nichts ändert. Jedes Muster bei Alexander steht so zwar mit anderen in Zusammenhang und entfaltet sich durch diesen Zusammenhang, kann aber auch außerhalb des Zusammenhangs isoliert beschrieben und in anderen Zusammenhängen ebenso integriert werden.<sup>4</sup>

Tätigkeitsmuster dagegen beschreiben einen Prozess. Ein Tätigkeitsmuster zur Herstellung eines bestimmten gegenständlichen Produktes verweist etwa auf bestimmte gegenständliche Mittel, welche durch die Anwendung von bestimmten Werkzeug auf eine bestimmte Weise miteinander *verbunden* werden. Die verbauten gegenständlichen Mittel stehen anschließend *nicht länger für sich* – sie sind Teil eines neuen Mittels. Und das so lange, bis durch einen anderen Prozess das verwendeten Mittel wieder entnommen wird und das auch nur, falls durch den vorhergehenden Prozess das Mittel nicht unwiderruflich verbraucht bzw. verändert wurde.

*“The specific patterns out of which a building or a town is made may be alive or dead. To the extent they are alive, they let our inner forces loose, and set us free; but when they are dead, they keep us locked in inner conflict.” (101)*

Tote bzw. lebendige architektonische Muster werden übersetzt als Tätigkeitsmuster mit niedriger bzw. hoher *Qualität*. Da es sich bei Tätigkeiten aber um einen Prozess handelt, muss diese Qualität zweigeteilt werden: Da ist der *Prozess* und da ist sein *Resultat*.



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.

*„There is always repetition of the patterns. But there is always variation and uniqueness in the way the patterns manifest themselves – Each pattern is a generic solution to some system of forces in the world. But the forces are never quite the same. Since the exact configuration of the surroundings at any one place and time is always unique, the configuration of the forces which the system is subject to is also unique – no other system of forces is ever subject to exactly the same configuration of forces.“ (146-147)*

*Commoning* ist eine auf Freiwilligkeit und Selbstorganisation basierende, beständige Veränderung der Welt zugunsten der menschlichen Bedürfnisse. Ein einzelner *Commoning*-Prozess ist die Befriedigung eines bestimmten Bedürfnisses durch in Kooperation stehenden Tätigkeiten, welche auf Softwareebene durch Tätigkeitsmuster beschrieben werden. Jede Tätigkeit benötigt Mittel und fähige Personen zu ihrer Ausführung und weder die Mittel noch die fähigen Personen sind gleichmäßig verteilt. Welche Form der *Commoning*-Prozess eines lokal vermittelten Bedürfnisses annimmt ist somit nicht beliebig, sondern abhängig davon, welche Mittel lokal vorhanden sind und den Tätigkeiten zugeordnet werden können und ob Personen im lokalen Umfeld sowohl für die Tätigkeit *befähigt*, als auch daran *interessiert* sind und weiter diese Tätigkeit als *sinnvoll* sowohl *für sich* als auch *für den Gesamtprozess* betrachten. Kurz: Die *Konfiguration* (Anordnung und Auswahl) der Tätigkeitsmuster eines *Commoning*-Prozesses ist davon abhängig, an welchem Ort die Bedürfnisse vermittelt wurden, welche Mittel *dort* zur Verfügung stehen, welche Fähigkeiten die Personen *dort* besitzen, welche Strukturen *dort* bereits bestehen und weiter noch, welche Formen der Tätigkeiten/Kooperationen *dort* sinnvoll erscheinen. Dieser Ortsbezug wird folgend als der *lokale Kontext* [H/B<sup>5</sup>] bezeichnet, in welchem ein Tätigkeitsmuster angewendet wird.

Die stetige Veränderung der Mittel im lokalen Kontext betrifft direkt die Qualität eines Tätigkeitsmusters. Ein Tätigkeitsmuster, das zu einem Zeitpunkt eine hohe Qualität hatte und innerhalb der Gesellschaft weit verbreitet war, kann an einem späteren Zeitpunkt eine so niedrige Qualität haben, das es kaum noch oder gar nicht mehr angewendet wird. Und diese Qualität der Tätigkeitsmuster wirkt sich – genau wie anstehende Bedürfnisse, die Fähigkeiten und Interessen der Teilnehmenden – direkt auf die gegebenen Strukturen aus, die genauso der stetigen Veränderung unterworfen sind. Da feste Strukturen daher dem Wesen des *Commonings* widersprechen, geht es auch mit der Software nicht darum, solche aufzubauen. Wie, wie lange und ob diese sich ergeben, liegt außerhalb ihrer Funktion.

*„But when we make our own attempt to create nature in the world around us, and succeed, we cannot escape the fact that we are going to die. This quality, when it is reached, in human things, is always sad; it makes us sad; and we can even say that any place where a man tries to make the quality, and be like nature, cannot be true, unless we can feel the slight presence of this haunting sadness there, because we know at the same time we enjoy it, that it is going to pass.“ (154)*

## 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.

Die Verfügbarmachung von Mitteln, welche sich durch den Akt des Weitergebens vermehren [H/B<sup>6</sup>], endet mit dem Abschluss der bestimmten Tätigkeit und befriedigt bzw. deckt damit alle darauf verweisenden Bedürfnisse bzw. Bedarfe. Für den Fall, dass es sich um Mittel handelt, welche sich entweder im Akt der Bedürfnisbefriedigung aufbrauchen bzw. nicht in der jeweiligen Situation *gemeinsam verwendet* werden können, benötigt die Verfügbarmachung von Mitteln für verschiedene Commoning-Prozesse eine (stetige) Wiederholung derselben Tätigkeit. Wenn einer Tätigkeit über der für einen einzelnen Commoning-Prozess notwendigen Dauer nachgegangen wird, wird nachfolgend von einer *kontinuierlichen Tätigkeit* gesprochen. Kontinuierliche Tätigkeiten haben tendenziell den Vorteil, dass ein Teil der dafür notwendigen Mittel nach Abschluss einer Durchführung nicht neu organisiert werden muss und damit die Prozessplanung vereinfacht wird. Weiter kann über eine kontinuierliche Tätigkeit auf Softwareebene festgestellt werden, welche Mittel zwar noch nicht lokal verfügbar sind, aber tendenziell einfach verfügbar gemacht werden können. Hierfür braucht es für am Commoning Beteiligte die Softwarefunktion, dass die Selbstzuordnung zu einer Tätigkeit auch bestehen bleibt, nachdem dieser erfolgreich nachgegangen wurde. Falls die Tätigkeit tendenziell häufig benötigt wird, kann eine Funktion der *vorsorgenden Reservierung* von dafür notwendigen Mitteln den Gesamtprozess unterstützen. Die Mittel wären damit nicht fest zugeordnet, aber durch die Reservierung können Absprachen und Auseinandersetzungen über die Verwendung von Mitteln zur generellen Bedürfnisbefriedigung unterstützt werden.

*Kontinuität* ist eine wesentliche Triebkraft zur Herstellung einer dichten gemeinsamen Mustersprache, welche selbst wiederum Voraussetzung für die *Effizienz* des ununterbrochenen Commonings ist.

Um sich dem anzunähern, noch einmal eine kurze Definition eines Musters:

*"Each pattern is a rule which describes what you have to do to generate the entity which it defines" (182)*

Jedes Tätigkeitsmuster beschreibt eine Tätigkeit, welche mit bestimmten Mitteln ein anderes Mittel erzeugt, (orts-)verändert oder erhält bzw. ein Bedürfnis direkt befriedigt.

Und was ist eine Mustersprache?

*"A pattern language is a system which allows its users to create an infinite variety of those three dimensional combinations of patterns which we call buildings, gardens, towns." (186)*

Die *Sprache aus Tätigkeitsmustern* ist ein System, das seinen Anwendern erlaubt jeden Prozess der Bedürfnisbefriedigung durch die Aneinanderreihung von Tätigkeitsmustern zu formulieren. Und wie jede Sprache hat auch sie eine Grammatik: Jeder Satz ist ein Commoning-Prozess. Ein Commoning-Prozess beginnt mit einem *Bedürfnis*, auf welches sich das erste Tätigkeitsmuster bezieht. Zwei Tätigkeitsmuster können nur über dasselbe Mittel verbunden werden und auch dann nur, wenn es bei dem einen ein *Bedarf* und bei dem anderen ein *Resultat* ist. Ein Satz ist nur vollständig, wenn es kein Tätigkeitsmuster mehr gibt, dessen Bedarf ungedeckt bleibt.

So ein Satz kann allerdings nicht ausgesprochen werden – um in dieser Metapher zu bleiben –, wenn es keine Menschen gibt, welche sich entsprechenden Tätigkeitsmustern zuordnen und auch dazu *befähigt* sind. Um sich den Aspekt

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 vom 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. [SWK<sup>7</sup>]

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.

*"The adaption between people and buildings is profound – Each detail has meaning. Each detail is understood. Each detail is based on some person's experience, and gets shaped right, because it is slowly thought out, and deeply felt. Because the adaption is detailed and profound, each place takes on a unique character. Slowly, the variety of places and buildings begins to reflect the variety of human situations in the town. This is what makes the town alive." (231)*

Es ist nicht nur die statische Auswertung des lokalen Kontextes durch die Verfügbarkeit der Mittel oder kontinuierlicher Tätigkeiten, welche einem neu entstehenden Commoning-Prozess seine Form geben. Über die Software werden lediglich Tätigkeiten vorgeschlagen, welche den bekannten Daten nach sinnvoll erscheinen. Die Auswahl der Tätigkeitsmuster, deren Ausführung die lokale Umwelt auf eine bestimmte Weise verändert, liegt alleine in den Händen der Beteiligten, ihren Fähigkeiten, ihrem Interesse und in dem, was ihnen in dieser Welt wichtig ist. Und jedes Tätigkeitsmuster entstammt der persönlichen Erfahrung eines anderen Menschen zu einer anderen Zeit an einem anderen Ort und was dieser Mensch zur Lösung desselben Problems als sinnvoll erachtet hat. Und die Strukturen des ununterbrochenen Commonings bilden sich heraus, indem andere Beteiligte diese individuellen Erfahrungen ebenfalls als sinnvoll zur Lösung desselben Problems betrachten. Wenn sich auch noch Bewegungstendenzen innerhalb der Mustersprache herausstellen werden, können diese Strukturen, durch welche an bestimmten Orten auf bestimmte Weise die Welt verändert wird, durch keine abstrakte Logik vorhergesagt werden, sondern ergeben sich aus der Vielfältigkeit der daran beteiligten Menschen selbst.

*"Each pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution – As an element in the world, each pattern is a relationship between a certain context, a certain system of forces which occurs repeatedly in that context, and a certain spatial configuration which allows these forces to resolve themselves. [...] It is both a process and a thing; both a description of a thing which is alive, and a description of the process which will generate that thing." (247)*

Drei Aspekte stehen bei einem Tätigkeitsmuster im Zusammenhang: Das Problem, die Lösung und der Kontext.

Das *Problem* eines Tätigkeitsmusters ist der Zweck, zu welchem es abgerufen wird. Der Bedarf also, der damit gedeckt oder das Bedürfnis, das damit befriedigt werden soll. Das Problem soll durch das *Resultat* der Tätigkeit möglichst vollständig gelöst werden. Inwiefern das Resultat geeignet ist, dieses Problem zu lösen, wird über die *sinnlich-funktionale Qualität* festgestellt.

Die *Lösung* eines Tätigkeitsmusters ist die Beschreibung der Tätigkeit mit ihrer Definition der dafür notwendigen Mittel und Fähigkeiten. In ihr wird beschrieben, wie unter Anwendung der notwendigen Mittel das Resultat hervorgebracht wird. Über die *Prozessqualität* wird festgehalten, wie aufwendig die Tä-

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. [H/B<sup>9</sup>]

*"The language is a good one, capable of making something whole, when it is morphologically and functionally complete – It is morphologically complete, when the patterns together form a complete structure, filled out in all its details, with no gaps. And it is functionally complete when the system of patterns has that peculiar self-consistency in which the patterns, as a system, generate only those forces which they themselves resolve – so that the system as a whole can live, without the action of self-destroying inner conflicts" (316)*

Das Ziel ist eine vollständige *Sprache aus Tätigkeitsmustern*, durch welche jeder einzelne Prozess zur Herstellung, Erhaltung und Veränderung der gesellschaftlichen Lebensbedingungen durch hochwertige Tätigkeitsmuster beschrieben werden kann. Ein darauf beruhendes Gesellschaftssystem wäre nicht statisch, sondern in ständiger Veränderung mit den Bedürfnissen jedes einzelnen Gesellschaftsteilnehmers, durch ihre steigenden Handlungsmöglichkeiten durch die stetige gesellschaftliche Verfügbarmachung neuer Erfahrungen und durch die inneren und äußeren Konflikte, in denen sich das System ständig bewähren muss. Und gleichgültig, ob es Auseinandersetzungen um die sinnvolle Anwendung begrenzter Mittel oder systemgefährdende Wandlungen der nicht-menschlichen Natur sind: Eine vollständige Mustersprache bietet zu jedem Problem Lösungen an.

*"A language is a living language only when each person in society, or in the town, has his own version of this language. – For it is then not only an intellectual thing which expresses patterns as invariants, as rules to follow, as knowledge about what makes a building or a town work right. – It is, a deeper thing, a felt thing, a thing lived through, which expresses people's innermost attitudes about their way of life, their hopes and fears about the ways in which they live and work together" (337)*

Auf Softwareebene sind *individuelle Tätigkeitsmuster* („eigene Erfahrungen zur Lösung eines bestimmten Problems“) nur gesellschaftlich verarbeitbar, wenn es sich dabei auch gleichzeitig um geteilte Tätigkeitsmuster handelt. Geteilte Tätigkeitsmuster, denen sich individuell angenommen werden kann, sind die eigene *Bibliothek*. Jedes bewältigte Tätigkeitsmuster, dem sich mindestens einmal erfolgreich angenommen wurde, kann automatisch in dieser Bibliothek aufgenommen werden. Da Fähigkeiten („verinnerlichte Tätigkeitsmuster“) ebenso aus erfolgreich abgeschlossenen Tätigkeitsmustern bestehen, können die entsprechenden Tätigkeiten innerhalb der Bibliothek durch die Anwender markiert werden. In dem Fall kann sich zukünftig Tätigkeitsmustern angenommen werden, welche diese markierten Tätigkeitsmuster als Fähigkeiten voraussetzen.

Eine solche Bibliothek als individuelle Mustersprache ermöglicht, dass nicht nur aktiv nach Möglichkeiten gesucht werden muss, um sich in den gesellschaftlichen Re-Produktionsprozess einzubringen. Über die Bewertung bestimmter Tätigkeitsmuster in der eigenen Bibliothek kann eingestellt werden, wann eine Benachrichtigung erfolgen soll, wenn Commoning-Prozesse in der lokalen Umgebung anfallen, welche auf diese Tätigkeitsmuster zurückgreifen.



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

lokal verfügbaren Mittel den Bedarf dieser Tätigkeit stetig decken müssen und andererseits auf das Resultat dieser Tätigkeit stetig zurückgreifen können. Für jede dieser Tätigkeiten gilt dasselbe wie für die erste Tätigkeit, dass sie also umso eher *kontinuierlich* werden können, je höher ihre Qualität ist, also je weniger aufwendig ihre Ausführung bei einer hohen sinnlich-funktionalen Qualität ihrer Resultate.

Indem die Mittel der lokal anstehenden Tätigkeiten dauerhaft aufeinander verweisen, entsteht eine *Verzahnung* des lokalen Kooperationsprozesses und durch eine Integration jeder Tätigkeit in möglichst viele verschiedene Commoning-Prozesse eine *Verdichtung* der insgesamt benötigten Mittel, welche durch die Verzahnung der Tätigkeiten tendenziell weniger aufwendig verfügbar gemacht werden können. Falls schließlich neue Bedürfnisse anstehen und es Tätigkeiten gibt, welche diese Bedürfnisse weitgehend durch die lokal verfügbaren (bzw. leicht verfügbar zu machenden) Mittel befriedigen können, dann sind diese Tätigkeiten von Anfang an qualitativ hochwertig und werden *immer hochwertiger* je leichter die Verfügbarmachung der Bedarfe im lokalen Kontext wird. Insofern ihre Resultate häufig benötigt werden bzw. vielseitig sind, gliedern sich diese Tätigkeiten damit ebenfalls in den lokal verzahnten Kooperationsprozess ein.

Damit diese Bewegungstendenz anlaufen kann und ununterbrochenes Commoning damit zur individuellen Bedürfnisbefriedigung immer effizienter wird, braucht es neben der Qualität einer Tätigkeit noch einen Indikator, in wie viele Commoning-Prozesse sich eine bestimmte Tätigkeit integrieren lässt. Dieser Indikator wird folgend als das *Tätigkeitsgewicht* [SWK<sup>9</sup>] bezeichnet und steigt, je mehr Bedürfnisse darauf verweisen. Weiter aber muss es nicht nur gesellschaftlich, sondern auch *individuell sinnvoll* werden, sich Tätigkeiten anzunehmen, welche einerseits eine hohe Qualität und andererseits ein hohes Gewicht aufweisen. Möglich wird das einerseits zu durch die *Sortierung* der Tätigkeiten nach ihrer Qualität und andererseits über eine Kopplung des Gewichtes einer Tätigkeit, welcher eine Person nachgeht, mit dem *Gewicht ihrer eigenen Bedürfnisse* (näher dargestellt in der Textreihe zum Softwarekonzept<sup>10</sup>). Die Gewichtung der eigenen Bedürfnisse wirkt sich damit auf das Gewicht der damit zusammenhängenden Tätigkeiten aus und „zieht“ das ununterbrochene Commoning in Richtung der eigenen Bedürfnisbefriedigung, ohne dabei einen Zwangaspekt in das Commoning hineinzubringen.

*„Of course, this evolution will never end. - Although the process of evolution will always move towards greater depth and greater wholeness, there is no end to it – there is no static perfect language, which, once defined, will stay defined forever. No language is ever finished.” (346)*

Durch das Werkzeug der gesellschaftlich geteilten Mustersprache und entsprechenden Funktionen, diese Tätigkeitsmuster sinnvoll anwenden zu können, haben wir es mit einer Gesellschaft nach Bedürfnissen und Fähigkeiten zu tun, die nicht einmal erreicht ist, sondern sich in ständiger Anpassung an die Veränderung der menschlichen und nicht-menschlichen Natur befindet. Ermöglicht wird dies, indem jeder Mensch mit seinen eigenen Erfahrungen die Möglichkeiten aller Beteiligten erweitern kann und durch seine eigenen Fähigkeiten und Interessen die Richtung beeinflusst, in welche sich die Gesellschaft entwickelt.

*"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)*

## THE WAY

*"Assume, to start with, that some version of the pattern language has been adopted in a town, or in a neighborhood, or by a group of people or a family who adopt it as the basis for the reconstruction of their world – What is the relation between this common pattern language and the constant process of construction and destruction which gives the town its shape?" (354)*

Bisher wissen wir, wie ein Tätigkeitsmuster aussieht, wie es entsteht, dass es eine bestimmte Qualität zur Befriedigung von Bedürfnissen hat und, dass es durch die Anwendung dieser Muster eine Bewegungstendenz gibt, welche die Gesamtzahl der häufig angewendeten Tätigkeitsmuster immer weiter reduziert und solche mit hoher Qualität hervorhebt. Wir wissen auch, dass die Auswahl und Anordnung von bestimmten Tätigkeitsmustern abhängig ist vom lokalen Kontext und, dass eine bestimmte Tätigkeit im Geflecht des ununterbrochenen Commonings nicht nur zur Befriedigung eines bestimmten Bedürfnisses beiträgt, sondern Teil verschiedener Commoning-Prozesse sein kann.

*"It's a flux of millions upon millions of these tiny acts, each one in the hands of the person who knows it best, best able to adapt it to the local circumstances. – But what will guarantee is there that this flux, with all its individual acts, will not create a chaos?" (355)*

Folgend wird herausgestellt, wie durch den Prozess der *Selbstzuordnung* sinnvolle Re-Produktionsstrukturen konfiguriert werden können, welche sich schließlich manuell den Bedürfnissen der Beteiligten anpassen lassen. Außerdem wird herausgestellt, wie sich aus der Logik dieser Struktur einerseits softwareunabhängige Selbstorganisation ergeben kann, andererseits wie sich bestehende selbstorganisierte Zusammenschlüsse in die Struktur des ununterbrochenen Commonings integrieren können.

Um sich dem anzunähern, muss im Kopf behalten werden, dass *Tätigkeiten* dem Abbau von *Spannungen* dienen und diese *Spannungen* der Aufwand zwischen Bedürfnissen und ihrer Befriedigung sind. Zu diesem Zweck stellen sich sinnvolle Tätigkeiten – insofern sie einmal als Tätigkeitsmuster beschrieben wurden – anhand der *lokal verfügbaren Mittel* heraus. Worum es folgend geht, ist eine solche Spannung schrittweise in einzelne Tätigkeiten zu *differenzieren*.

*"The image of the differentiating process is the growth of an embryo. – The unfolding of a design in the mind of its creator, under the influence of language, is just the same. – A language allows you to generate an image of a building in your mind, by placing patterns in space, one pattern at a time" (370-372)*

Die Anwendung der Mustersprache ist bei Alexander der *innere Prozess* einzelner Personen, bei welcher sich die Architektur aus der schrittweisen Einteilung durch die Muster ergibt. Die Vorstellung eines Designers von gesellschaftlichen Re-Produktionsprozessen mag dagegen in Teilbereichen sinnvoll sein

[SWK<sup>11</sup>], 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 schließlich den Commoning-Prozess bestimmen. Welche Tätigkeitsmuster ausgewählt werden und wer sich ihnen zuordnet 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-

keitsmuster, das festgelegt werden muss, ist dabei die Tätigkeit, welche das Bedürfnis befriedigt. Anschließend werden die Tätigkeitsmuster festgelegt, welche den Bedarf dafür decken usw. usf.

Das sinnvollste Tätigkeitsmuster ist jeweils das mit der höchsten Qualität und ein Teil dieser Qualität ist die Prozessqualität. Die Prozessqualität entsteht aus der Analyse der im lokalen Kontext verfügbaren Mittel und setzt sich aus dem Aufwand der Tätigkeiten zusammen, die notwendig sind, um durch diese lokal verfügbaren Mittel das anstehende Bedürfnis zu befriedigen. Die Prozessqualität enthält dabei sowohl den Aufwand der Tätigkeit, die im Tätigkeitsmuster beschrieben wird, als auch den Aufwand der Tätigkeiten, die zur Verfügbarmachung des Bedarfes notwendig werden. Das heißt damit einerseits, dass jedes neu festzulegende Muster den Aufwand der gesamten nachfolgenden Konfiguration *enthält* und andererseits, dass sich die Qualität der bereits festgelegten Tätigkeitsmuster während dieses *Konfigurationsprozesses* ständig verändern kann. Denn wenn sich auch durch die Analyse des lokalen Kontextes bestimmte Tätigkeiten zur Bedarfsdeckung als *ideal* herausstellen, dann ist noch nicht gesagt, dass sich auch Personen *diesen* Tätigkeiten zuordnen werden. Ob das der Fall ist, kann allerdings erst festgestellt werden, *nachdem* die Tätigkeit festgelegt wurde. Die Aussage, ob eine bestimmte Tätigkeit im bestimmten lokalen Kontext sinnvoll ist, ist also *spekulativ*. Und diese Spekulation kann nur auf der Grundannahme geschehen, dass nach Festlegung der Tätigkeit sich weitere Beteiligte immer den nachfolgenden jeweils qualitativ höchsten Tätigkeitsmustern zuordnen werden.

Im Konfigurationsprozess wird das Muster mit der (spekulativ) höchsten Qualität zuerst für die Selbstzuordnung *freigeschaltet*. Ordnet sich in einem geregelten zeitlichen Abstand niemand dieser Tätigkeit zu oder gibt es in der lokalen Umgebung keine Person mit den notwendigen Fähigkeiten und einem prinzipiellen Interesse an der Tätigkeit, kann das Tätigkeitsmuster mit der nächsthöchsten Qualität zur Selbstzuordnung *zusätzlich* freigeschaltet werden. Der Prozess wiederholt sich, bis eine Selbstzuordnung stattfindet. Wurde sich einer freigeschalteten Tätigkeit zugeordnet und angegeben, welche Bedarfe selbstständig gedeckt werden können, werden die jeweils qualitativ höchsten Tätigkeitsmuster zur Bedarfsdeckung dieser Tätigkeit ebenfalls freigeschaltet und der Prozess wiederholt sich für jeden Bedarf erneut.

Die Qualität des jeweils übergeordneten Tätigkeitsmusters verändert sich, wenn zur Bedarfsdeckung nicht das jeweils höchste Muster gewählt wurde. Es ist daher möglich, dass ein bereits festgelegtes Muster während des Konfigurationsprozesses so weit an Qualität verliert, dass ein anderes Tätigkeitsmuster auf derselben Ebene spekulativ eine höhere Qualität zur Bedürfnisbefriedigung haben kann. In diesem Fall kann auch dieses Tätigkeitsmuster *zusätzlich* freigeschaltet werden und bildet einen *unabhängigen Strang* im Konfigurationsprozess. Tätigkeiten können von da an für beide Stränge freigeschaltet werden, bis sich ein Strang als eindeutig qualitativ hochwertiger herausstellt und die Freischaltung von Tätigkeiten für den anderen Strang dadurch unter- bzw. abgebrochen wird. Über diese Möglichkeit kann eine Konfiguration durch Selbstzuordnung herausgestellt werden, die im lokalen Kontext weitgehend ideal ist. Es folgt daraus allerdings auch, dass nicht jede Selbstzuordnung zu einer Tätigkeit führt. Für die Beteiligten, welche sich einer bestimmten Tätigkeit zugeordnet haben, muss daher der Prozess und Fortschritt des entsprechenden Konfigurationsprozesses transparent sein.

Durch den Konfigurationsprozess beginnt der Commoning-Prozess, vom Bedürfnis ausgehend, Schritt für Schritt an Form zu gewinnen („each pattern moulds the whole“). Einen Sonderfall bildet dabei die Tätigkeit der *Ortsveränderung*. Nicht im Fall der Ortsveränderung eines bereits verfügbaren Mittels, das der Tätigkeit zugeordnet werden kann, sondern die Ortsveränderung von Mitteln, die erst durch eine andere Tätigkeit verfügbar gemacht werden. Innerhalb der Konfiguration ist sie strukturell näher am Bedürfnis, kann aber erst abgefragt werden – insofern es notwendig ist – *nachdem* die Tätigkeit, welche das Mittel verfügbar macht, und der Ort ihrer Ausführung festgelegt wurden. Die Tätigkeit der Ortsveränderung eines Mittels zwischen zwei aufeinander bezogenen Tätigkeiten wird daher erst freigeschaltet, nachdem die Tätigkeiten festgelegt und ihr Ort bestimmt wurde oder aber, – falls die Ortsveränderung im jeweiligen lokalen Kontext tendenziell unproblematisch ist – erst nach Abschluss der konkreten Tätigkeit, um die Planung des Gesamtprozesses zu vereinfachen. Ob die Tätigkeit einer solchen Ortsveränderung in den Konfigurationsprozess aufgenommen werden soll oder nicht, muss als Softwarefunktion in den Händen der Personen liegen, zwischen denen das Mittel transportiert werden muss.

Der Konfigurationsprozess zu einer bestimmten Bedürfnisbefriedigung ist abgeschlossen, wenn der Bedarf jeder Tätigkeit durch eine andere Tätigkeit gedeckt werden kann.

*"When the order of the patterns in the language is correct, the differentiating process allows the design to unfold as smoothly as an opening flower." (384)*

Der beschriebene Auswahlprozess findet *für jedes einzelne Bedürfnis* statt. Bedürfnisse *derselben Art* bündeln sich damit nicht automatisch, sondern werden erst miteinander verbunden, wenn im lokalen Kontext ihrer jeweiligen Vermittlung dasselbe Tätigkeitsmuster freigeschaltet wird. Ganze Commoning-Prozesse werden somit – durch den Verweis auf dasselbe Tätigkeitsmuster im selben lokalen Kontext – miteinander verschlossen. Dass zwei oder mehr Bedürfnisse dadurch auf dieselben Tätigkeiten verweisen, erhöht das *Gewicht* dieser Tätigkeiten. Je höher das Gewicht einer Tätigkeit, desto sinnvoller ist damit die Tätigkeit für den Gesamtprozess und desto sinnvoller kann die Tätigkeit auch individuell sein, wenn das Gewicht der ausgeführten Tätigkeit mit dem Gewicht der eigenen Bedürfnisse gekoppelt ist.

Ebenso wie die Qualität einer Tätigkeit während des Konfigurationsprozesses, ist das Gewicht einer Tätigkeit bei der Selbstzuordnung *spekulativ* und davon abhängig, wie viele darauf verweisende Commoning-Prozesse *nach* der Selbstzuordnung tatsächlich aktiviert werden. Weiter ist hier zu beachten, dass das gesamte Gewicht nur bei Mitteln vollständig „abgebaut“ wird, welche sich durch Teilen vermehren (oft soziale und symbolische Mittel) bzw. welche im lokalen Kontext seriell genutzt werden können. Bei gegenständlichen Mitteln, die sich in der Bedürfnisbefriedigung aufbrauchen, ist eine wiederholte, sprich: *kontinuierliche* Tätigkeit notwendig. Als Softwarefunktion braucht es hier die Möglichkeit, sich nur einer bestimmten *Menge* der darauf verweisenden Prozesse anzunehmen.

Auch wenn für einen einzelnen Commoning-Prozess das jeweils qualitativ höchste Tätigkeitsmuster unbedingt am sinnvollsten ist, kann es für den Gesamtprozess effektiver sein, wenn Tätigkeitsmuster gewählt werden, die zwar eine niedrigere Qualität haben, dafür aber in mehr Prozesse integriert werden

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)*



An dieser Stelle muss festgehalten werden, dass auch bei über die Software vermittelter, kontinuierlicher Tätigkeit direkte zwischenmenschliche Beziehungen nicht notwendig sind. Sind die konkret verwendeten Mittel (besonders auch die Lokalität der Tätigkeit) und Zeiträume des Prozesses eindeutig definiert, braucht es an sich keinerlei näheren Kontakt zwischen den am Prozess beteiligten Personen.

Was der direkte zwischenmenschliche Kontakt allerdings ermöglicht, sind Absprachen. Sowohl zeitlicher Natur, als auch, ob noch Hilfe benötigt wird oder etwa, an welcher Stelle das zuerst gefertigte Resultat einer kontinuierlichen Tätigkeit am dringendsten gebraucht wird. Weiter wird die Erfahrung ermöglicht, dass es echte Menschen sind, die sich darauf verlassen, dass die eigene Tätigkeit gemacht wird und, dass sie gut gemacht wird. Und direkte zwischenmenschliche Beziehungen ermöglichen direkte Rückmeldung, wenn etwas gut oder mangelhaft gemacht wurde. Weiter ermöglichen direkte zwischenmenschliche Beziehungen, dass auftretende Probleme schnell und unkompliziert geklärt werden können. Statt lange darauf zu warten, dass etwas selbst Benötigtes zur Verfügung gestellt wird, schnell in der entsprechenden Produktionsstätte mit anpacken zu können oder einen Transport selbst zu übernehmen. Und weiter Menschen kennenzulernen, mit denen solche Prozesse gerne zusammen angegangen werden oder schlicht die Eigenheiten von anderen berücksichtigen zu können und dabei auch unvermittelte Fürsorge zu leisten. Direkte zwischenmenschliche Beziehungen sind daher nicht nur hilfreich, sondern auch sinnvoll für die Effizienz des ununterbrochenen Commonings.

Und was die Software elementar hierfür bereitstellen muss, sind Möglichkeiten der *Kommunikation* zwischen den am Commoning Beteiligten. Jede Person innerhalb eines eigenen Commoning-Prozesses muss kontaktiert und jede Person im gleichen Zusammenhang einem solchen Gespräch hinzugefügt werden können. In den dadurch entstehenden Gruppen muss schnell ersichtlich sein, welche Personen welcher der in Zusammenhang stehenden Tätigkeiten nachgeht. Weiter müssen Ergebnisse solcher Kommunikationen transparent für alle Beteiligten des jeweiligen Commoning-Prozesses gemacht werden können.

*"Now we come to the actual building of the building. - Again, just as before, the process is sequential. Only now the patterns operate not on a mental image, but on the building itself, as it is being built. Each pattern defines an operation, which helps to differentiate, and to complete, the building as it grows: and when the last patterns are introduced into the growing fabric, the building is complete." (459)*

Nachdem ein Commoning-Prozess aktiviert wurde und die teilnehmenden Personen in Verbindung gebracht wurden, folgen die Tätigkeiten einer klaren Reihenfolge: Angefangen von der am „weitesten“ vom Bedürfnis entfernten Ebene werden die Mittel für die nächsthöhere verfügbar gemacht. Sind alle Mittel für eine bestimmte Tätigkeit verfügbar gemacht worden, kann diese einen Bedarf der übergeordneten Tätigkeit decken usw., bis schließlich als letzter Schritt das Bedürfnis selbst befriedigt wird.

Spätestens wenn das letzte an einer kontinuierlichen Tätigkeit anstehende Bedürfnis befriedigt ist, löst sich der Commoning-Prozess wieder auf. Einzelne Tätigkeiten des Prozesses bleiben selbstverständlich bestehen, insofern sie

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.

Die Veränderung eines Commoning-Prozesses kann jederzeit vorgenommen werden, sofern sie in Übereinkunft mit allen davon Betroffenen geschieht. Rein auf *Ebene der Effizienz* kann ein solcher Reparaturprozess sinnvoll sein, um Tätigkeitsmuster mit höherer Qualität und höherem Gewicht in den Gesamtprozess zu integrieren. Besonders bei kontinuierlichen Tätigkeiten im Zusammenhang kann das notwendig werden, wenn sich die Verfügbarkeit von Mitteln im lokalen Kontext so verändert, dass es sich auf die Prozessqualität der Tätigkeiten signifikant auswirkt, sprich, diese deutlich aufwendiger werden, als es während des Konfigurationsprozesses der Fall war. Auf rein *menschlicher Ebene* kann ein Reparaturprozess sinnvoll sein, um durch die Softwarevermittlung entstandene Strukturen, welche auf Softwareebene sinnvoll wirken, an das tatsächliche Leben und dem, was sich darin sinnvoll und richtig anfühlt, anzupassen. Und weiter kann es vorkommen, dass Personen aus aktivierten Commoning-Prozessen wegfallen und niemand anderes die bestimmte Tätigkeit übernehmen kann oder will – in dem Fall ist eine Veränderung der Konfiguration notwendig, um die bestimmte Bedürfnisbefriedigung weiter gewähren zu können.

*"When things are first built, the gaps between the parts are often left unwhole. - But these gaps must be healed and made as whole as the parts on either side of them - [...] Slowly, as the »process of repair« repairs the gaps between the wholes, the structure becomes complete and whole at every level."* (483-485)

An dieser Stelle zur Erinnerung: Die Vision ist eine Gesellschaft, die aus dem Alltagsbewusstsein heraus einfach verstanden werden kann. Bei der jede ausgeführte, gesellschaftliche Tätigkeit gezielt und nachprüfbar reale Bedürfnisse befriedigt. Eine Gesellschaft, deren Strukturen sich den menschlichen Bedürfnissen anpasst, also dem, was individuell als richtig und gut empfunden wird. Und das in jedem Lebensbereich. Eine Struktur, die damit auch nicht starr ist, sondern im stetigen Wandel, wie die Umstände und die Menschen selbst. Eine Gesellschaft, in der es zwar auch Krisen und Engpässe geben mag, an denen aber gemeinsam gearbeitet werden kann, in denen auf Augenhöhe Konflikte gemeinsam gelöst werden. Die Software selbst ist ein Werkzeug – eine Vermittlungsform –, das den Weg in eine solche Welt unterstützen kann. Durch den *Konfigurationsprozess* wird versucht, eine *Effizienz der Bedürfnisbefriedigung* herzustellen, wodurch allerdings Strukturen entstehen können, die den Bedürfnissen der darin Beteiligten widersprechen. Der Reparaturprozess ist daher auch ein Werkzeug, um *gegen die Logik der Software* die Welt den eigenen Bedürfnissen nach zu gestalten.

*"When we repair something in this new sense, we assume that we are going to transform it, that new wholes will be born, that, indeed, the entire whole which is being repaired will become a different whole as the result of the repair – In this sense, the idea of repair is creative, dynamic, open."* (485)

Die Software muss also um die Funktion erweitert werden, dass Konfigurationen manuell und in Absprache mit sämtlichen davon Betroffenen verändert werden kann. Das heißt, es werden neue Tätigkeitsmuster hinzugefügt und andere aus dem Prozess herausgenommen. Ob sich dann direkt in diesen Absprachen Personen den neuen Tätigkeiten zuordnen oder eine Selbstzuord-

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-*

zessen besteht, kann uns hier nicht helfen. Wer sich dagegen intensiv mit genau solchen transformativen Prozessen auseinandergesetzt hat, ist Klaus Holzkamp.

### **Auftritt Holzkamp.**

Wie schon bei Alexander interessiert uns in Holzkamps Werk nicht sein Gegenstand, sondern seine Methode, die er in seinem Hauptwerk „die Grundlegung der Psychologie“ (1985) herausgearbeitet hat: Ein Fünfschritt, in welchem der Umschlag von der *Quantität* zu einer neuen *Qualität* präzise beschrieben wird, unter der Voraussetzung bestimmter Bedingungen und gegebener Entwicklungswidersprüche. Die erste Anwendung dieses Fünfschritts im Bereich der gesellschaftlichen Transformation stammt dabei von Stefan Meretz, welcher diese in „Kapitalismus aufheben“ (2018) gemeinsam mit Simon Sutterlütti näher ausgearbeitet hat. Und während Klaus Holzkamp seine Methode verwendet hat, um zu analysieren, *was bereits passiert ist*, wird folgend spekulativ angedacht, welche Handlungsmöglichkeiten bei steigender Komplexität sinnvoll werden *könnten* und wie die Software sie im Falle ihres Auftretens unterstützen kann. Weiter soll damit auch herausgestellt werden, wie *bestehende Formen der Selbstorganisation*, welche sich nicht innerhalb des Rahmens der Softwarevermittlung gebildet haben, sich – soweit es für die daran Beteiligten sinnvoll erscheint – in die Struktur des ununterbrochenen Commonings integrieren können.

**„Erster Schritt: Aufweis der *realhistorischen Dimension* innerhalb der jeweils *früheren Stufe*, auf denen der qualitative Umschlag sich vollzieht. [...] Es soll genau die ›Position‹ bestimmt werden, die beim qualitativen Umschlag *dialektisch* ›negiert‹ wird.“ (Holzkamp: 78)**

Die durch die Software ermöglichte *Vorbedingung* (Holzkamp mit dem Blick zurück: „*realhistorische Dimension*“) ist eine transparente Struktur aus tendenziell kontinuierlichen Tätigkeiten in lokaler Nähe, wobei jede davon für sich steht, diese aber durch ihren Bedarf an Mitteln und dem Zweck in ihrer Ausübung in Abhängigkeit zueinander stehen. Die Organisation der Mittel und die Integration der eigenen Fähigkeiten in die ununterbrochenen Prozesse der direkten Bedürfnisbefriedigung wird über die Software vermittelt.

**„Zweiter Schritt: Aufweis der *objektiven Veränderungen der Außenweltbedingungen*, mit denen der ›innere‹ *Entwicklungswiderspruch*, durch welchen die neue Qualitätsstufe in evolutionärer Progression hervorgebracht werden kann, in seinem *Umwelt-Pol* zustandekommen soll.“ (Holzkamp: 79)**

Wird die Software selbst als der „*Umwelt-Pol*“ der am ununterbrochenen Commoning Beteiligten betrachtet, dann ergeben sich die „*objektiven Veränderungen der Außenweltbedingungen*“ durch die für sie steigende Komplexität des Gesamtprozesses und die damit einhergehende, auf dem fragilen Fundament der Freiwilligkeit beruhende, zunehmende Abhängigkeit von der Ausführung bestimmter Tätigkeiten. Weiter noch, als Teil dieses „*inneren Entwicklungswiderspruches*“, kann die zunehmende Komplexität und steigende Abhängigkeit von einzelnen Tätigkeiten erfordern, dass Personen, deren Tätigkeit in verschiedene Commoning-Prozesse integriert ist, einer immer strikteren Zeitplanung unterworfen sind und somit Handlungsfreiheit, welche das Commoning

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-

stützende Funktion höchstens angeboten werden, solche Absprachen transparent zu machen, um so Planungsprozesse für andere zu erleichtern.

**„Vierter Schritt: Aufweis des Dominanzwechsels zwischen der für die frühere Stufe charakteristischen Funktion und der neuen Funktion, womit durch einen zweiten qualitativen Sprung die qualitativ spezifische Funktion auch die für die gesamte Systemerhaltung bestimmende Funktion wird. [...] Eine solche Umkehrung des Verhältnisses zwischen bestimmender und nachgeordneter Funktion als Dominanzwechsel ist, obwohl sich beide Funktionen in der Entwicklung kontinuierlich darauf zubewegen, selbst nicht kontinuierlich, sondern ein punktuelles Umkippen.“ (Holzkamp: 80)**

Der „Dominanzwechsel“ geschieht für die Beteiligten nicht plötzlich. Er entsteht, indem einzelne Beteiligte sich immer mehr auf die selbstgesetzten Regeln und Absprachen beziehen und sich damit immer weniger (im Rahmen dieses Zusammenhangs) über die Software vermitteln. Der „zweite qualitative Sprung“ sind dabei solche Absprachen, die in die Organisation der Mittel und der Tätigkeiten selbst eingreifen, wodurch die Softwarevermittlung zum *Hemmnis* wird. Wenn es also für die Beteiligten ein *kontinuierlicher Prozess* zur von der Software weitgehend unabhängigen Selbstorganisation ist, braucht es auf Softwareebene einen *Bruch* mit den für sich stehenden Tätigkeiten, welche durch etwas *qualitativ anderes* ersetzt werden müssen. Dieser Bruch innerhalb der Software muss dabei durch die Beteiligten ausgelöst werden, geschieht also nicht von selbst.

Sofern ein solcher softwareunabhängiger Zusammenschluss, in welchem sich die daran beteiligten Personen ihre eigenen Regeln setzen und Tätigkeiten und Mittel selbstständig koordinieren, Teil des ununterbrochenen Commonings bleibt, wird er als „*integrierter Zusammenschluss*“ bezeichnet. Eine definierte Gruppe von Personen schließt sich dabei auf einer Ebene zusammen, welche die Software nicht fassen kann, aber dieser Zusammenschluss muss durch die Integration in das ununterbrochene Commoning von der Software gefasst werden können. Und das dabei nur insofern, wie die daran Beteiligten sich in das ununterbrochene Commoning einbringen wollen. Denn, und das nur eine allgemeine Erinnerung, das softwarevermittelte Commoning ist lediglich *eine Vermittlungsform* und kann nur das abbilden und nur das unterstützen, was von den Beteiligten gewollt ist. Es ist ein *Werkzeug* um Commoning auf gesamtgesellschaftlicher Ebene überhaupt betreiben zu können, aber nur ein Aspekt des Commonings an sich. Wenn ein Zusammenschluss innerhalb der Softwarestruktur auftritt, bedeutet es damit nicht unbedingt, dass er erst unmittelbar entstanden ist und wenn ein solcher Zusammenschluss aus der Softwarestruktur verschwindet, bedeutet es nicht unbedingt, dass er nicht mehr existiert.

Die Integration eines Zusammenschlusses *ersetzt* die Tätigkeitsmuster an entsprechender Stelle. Und sowie die Entstehung solcher Zusammenschlüsse aus der Vermittlung über Tätigkeitsmuster heraus unterstützt werden soll, muss auch die Auflösung in Tätigkeitsmuster zurück möglich sein. Außerdem benötigt die Software die Funktion, dass ein Zusammenschluss auf Ebene der Softwarestruktur *deaktivierbar* ist, um ihn zu einem späteren Zeitpunkt wieder *reaktivieren* zu können. Diese Funktion schließt damit an die Softwarefunktio-

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.<sup>12</sup> 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



über diese Mittel („governance activities are organized in multiple nested layers“).

Wenn die integrierten Zusammenschlüsse auch starke Parallelen zu den nach Ostrom beschriebenen Commons haben, ist die Verwendung des Begriffes selbst irreführend und die Übertragung der Prinzipien darauf nicht widerspruchsfrei möglich: Das Commons bei Ostrom ist eine Besonderheit innerhalb einer durch (privates) Eigentum bestimmten Umgebung. Das zeigt sich besonders dadurch, dass es bei Ostrom als individueller Vorteil um die „Aneignung“ (*Appropriation*) der Ressource geht. Im Zentrum steht also ein Mittel (bzw. die „Ressource“), das nicht Eigentum einer konkreten Person bzw. einer Gruppe konkreter Personen ist und wodurch niemand alleine über dessen Nutzung bestimmen kann. Aus diesem Grund kann sich um dieses bestimmte Mittel ein Zusammenschluss von Personen bilden, in welchem alle dasselbe Nutzungsrecht an der Ressource haben und welcher schließlich dessen Nutzung regelt. Und dieser Zusammenschluss und dieses Mittel werden bei Ostrom als *Einheit*, als *Commons*, beschrieben. In einer von Commoning bestimmten Umgebung löst sich diese Einheit dagegen auf. Die für den Prozess des Commonings verfügbaren Mittel sind nicht länger Inseln, sondern das Fundament, auf welchem sich die Beteiligten – und damit auch Zusammenschlüsse von Beteiligten – bewegen.<sup>13</sup>

Die Übertragung der Prinzipien ist weiter deswegen problematisch, weil sich bei Ostrom sämtliche Regelungen, Sanktionen und Konfliktlösungsmechanismen den Zweck haben, die gemeinsame Verwendung des Mittels zu ermöglichen bzw. das Verhältnis des Mittels zu anderen Institutionen zu klären. Im ununterbrochenem Commoning dagegen hat der Zusammenschluss einen anderen Zweck: Die *generelle Bedürfnisbefriedigung*. Bei Ostrom entsteht dabei der individuelle Vorteil über diese *Aneignung* des Resultates der Tätigkeit am Mittel und im Gegensatz muss etwas *bereitgestellt* werden. Im ununterbrochenem Commoning dagegen wird das Resultat der eigenen Tätigkeit bzw. der Tätigkeit des Zusammenschlusses von anderen verwendet und der individuelle Vorteil entsteht auf *andere Weise* (abermals kann an dieser Stelle nur auf die Textreihe zum Softwarekonzept verwiesen werden).

In Einbeziehung dieser Unterschiede wird folgend die Software den Prinzipien von Ostrom nach differenziert, damit solche nicht über die Software vermittelten Zusammenschlüsse innerhalb des ununterbrochenen Commonings ihren eigenen Regeln nach wirken und tendenziell langlebig sein können.

Die Entstehung von integrierten Zusammenschlüssen bedeutet auf Strukturebene der Software eine Auflösung der für sich stehenden Tätigkeitsmuster innerhalb des entsprechenden Zusammenhangs, während der integrierte Zusammenschluss selbst deren Platz einnimmt. Ein integrierter Zusammenschluss kann, *muss aber nicht*, innerhalb der durch die Softwarevermittlung hergestellten Struktur und damit aus Tätigkeitsmustern heraus entstehen, ist aber ab seiner Entstehung unabhängig von diesen. Jeder integrierte Zusammenschluss ist dabei prinzipiell *einzigartig* durch die Regeln der daran Beteiligten und ihrer Organisation der Tätigkeiten. Während ein Tätigkeitsmuster daher einer möglichst allgemeingültigen Beschreibung bedarf, um es in möglichst viele verschiedene Commoning-Prozesse gleichartig durch unterschiedliche Personen integrieren zu können, ist die *Beschreibung* eines integrierten Zusammenschlusses innerhalb der Software optional. Die Beschreibung kann dazu dienen, die eigenen Strukturen und Regeln verständlich zu machen und sollte nur durch die am Zusammenschluss Beteiligten bzw. in Absprache mit

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 *funktionslos*. 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-

grierten Zusammenschluss *Beteiligten*, sondern auch die *von den Tätigkeiten des Zusammenschlusses Betroffenen* autorisiert sein müssen, um an Prozessen der Regelerstellung und Abänderung teilhaben zu können. Es braucht daher eine Softwarefunktion, durch die angegeben werden kann, selbst – in irgendeiner Form – von einem bestimmten integrierten Zusammenschluss betroffen zu sein und damit die *Autorisierung zur Teilhabe* an diesen Prozessen der Regelerstellung und -abänderung zu erlangen. In jedem Fall braucht es auch *Kommunikationsräume zur schnellen und wenig aufwendigen Konfliktlösung* zwischen am Zusammenschluss Beteiligten und nicht am Zusammenschluss Beteiligten, die auf Softwareebene zur Verfügung gestellt werden sollten. Die Überwachung und Planung der konkreten Tätigkeiten und zugeordneten Mittel, die Erstellung und Abänderung von Regeln, die Form und Höhe der Sanktionen, die Zuteilung des Bedürfnisgewichtes sowie die Formen der gemeinschaftlichen Entscheidungsfindung und Konfliktlösung *zwischen am integrierten Zusammenschluss Beteiligten* sind dabei interne Prozesse desselben. Falls es Personen gibt, welche die Tätigkeiten und den Zustand der Mittel innerhalb des Zusammenschlusses überwachen, sollten auf Softwareebene allerdings definiert sein, wer diese Personen sind. Ein *Musterspeicher für Regel- bzw. Sanktionssysteme und Konfliktlösungsmechanismen*, die unabhängig von den Tätigkeiten eines integrierten Zusammenschlusses vorgeschlagen werden können, ist denkbar, betrifft aber nicht die Softwarevermittlung selbst.

*"It is vastly more complex than any other kind of order. It cannot be created by decision. It cannot be designed. It cannot be predicted in a plan. It is the living testament of hundreds and thousands of people, making their own lives and all their inner forces manifest."*  
(510)

Die Vermittlungsform einer Gesellschaft nach Bedürfnissen und Fähigkeiten ist unbedingt komplexer als jede bisherige gesellschaftliche Vermittlungsform es war. Diese Komplexität entsteht nicht von selbst, sie muss entwickelt werden und durch diese Entwicklung in den Hintergrund geraten, für eine Welt, die sich einfach erschließen lässt. Wenn die Software diese Funktion als Werkzeug erfüllen kann, dann kann eine gesellschaftliche Ordnung entstehen, die weder planbar noch vorhersehbar ist, aber durch nichts als die Bedürfnisse, Fähigkeiten und Vorstellungen der Beteiligten geformt wird.

*"And as the whole emerges, we shall see it take that ageless character which gives the timeless way its name. This character is a specific, morphological character, sharp and precise, which must come into being any time a building or a town becomes alive: it is the physical embodiment, in buildings, of the quality without a name."* (511)

## 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

stehen, desto besser können Beteiligte darauf vertrauen, dass Tätigkeiten mit *hoher Qualität* und *hohem Gewicht* dazu beitragen, eine Gesellschaft nach Bedürfnissen und Fähigkeiten herzustellen und zu erhalten. Ob Beteiligte dann diesen Tätigkeiten nachgehen bzw. diese Tätigkeiten selbst als sinnvoll empfinden, liegt immer in ihrem Ermessen. Abhängig allerdings ist all das weiterhin davon, ob eine angemessene Möglichkeit gefunden wird, den *allgemeinen Aufwand* einer bestimmten Tätigkeit festzustellen.

Wenn Spannungen im softwarevermittelten Commoning auch auf gesellschaftlicher Ebene verlaufen und damit nicht durch Menschen selbst *empfunden* und entsprechend *aufgelöst* werden können, kann die Anwendung von Tätigkeitsmustern und damit die mögliche Ausbreitung von Commoning-Strukturen, wie auch bei Alexander, ebenso als ein *"gate"*, ein Durchgang, betrachtet werden. Indem sich Commoning als gesellschaftliche Re-Produktionsweise etabliert, etabliert sich damit auch seine gesellschaftliche Logik und durch die Normalisierung der Logik zeigt sich die Verrücktheit der heutigen Normalität: Die Abhängigkeit von existenziellen Ängsten als Arbeitsmotivation. Die ewig gleiche Arbeitswoche trotz ständigen technischen Fortschritts. Der ständige Versuch, neue Arbeitsplätze zu schaffen im scheinbar notwendigen Kampf gegen Arbeitslosigkeit. Und wenn es auch nicht das „Wiederbeleben einer Erinnerung“ und auch nicht eine „Rückkehr zur ursprünglichen Gesellschaft“ ist, so ist das ununterbrochene Commoning dennoch zeitlos, indem jede Geste der Freundlichkeit das Allgemeine dieser Gesellschaft in sich trägt. Und zwischen der Vermittlungsform der Software und der rein zwischenmenschlichen Vermittlung gibt es keinen Bruch: Durch die Anwendung der Tätigkeitsmuster in den softwarevermittelten Strukturen kann ein Verständnis der bedürfnisorientierten Re-Produktionsweise nach Commons-Prinzipien *verinnerlicht* und durch diese Erfahrungen auch *außerhalb* der softwarevermittelten Strukturen weitergetragen werden.

*"Almost everybody feels at peace with nature: listening to the ocean waves against the shore, by a still lake, in a field of grass, on a windblown heath. One day, when we have learned the timeless way again, we shall feel the same about our towns, and we shall feel as much at peace in them, as we do today walking by the ocean, or stretched out in the long grass of a meadow." (549)*

Größter Dank und größte Wertschätzung gilt dem Lehrer lebendiger Ordnung, Christopher Alexander. Doch nicht minderer Dank, nicht mindere Wertschätzung und alle Hoffnung gilt denen, die sich das Projekt zu eigen machen, sich mit Entwicklung, Design, Organisation, Übersetzung und Verbreitung einbringen, die Kritik nicht als Sport üben, sondern zum Zweck der Verbesserung, der Erneuerung, des stetigen Anpassens an neue Möglichkeiten; sowohl technischer Natur als auch, um neue Anwender und Anwenderinnen zu erreichen. Dank, Wertschätzung und Hoffnung gilt auch denen, welche die Grenzen sehen, die dem Commoning gesetzt sind und daran arbeiten, diese Grenzen auszuweiten, zu lockern und zu lösen. Für uns, die eine Gesellschaft erstreben, die sich den Bedürfnissen und Fähigkeiten nach gestaltet, sieht die Zukunft längst nicht mehr rosig aus. Wir können nicht darauf hoffen, dass unsere Zukunft – die Zukunft unseres Begehrens – eintreten wird, wenn wir nicht anfangen abseits bisherigen Traditionen der heute bestimmenden Vernunft entgegenzutreten,

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.

## Anhang

Ein Letztes: Ich habe bisher offen gelassen, warum die Interpretation von „The Timeless Way of Building“ auf das Commoning bzw. eine Software, welche Tätigkeit im gesellschaftlichen Rahmen nach Commons-Prinzipien vermittelt, meiner Ansicht nach sinnvoll funktioniert. Ich will dabei voranstellen, dass es sich mir selbst erst in der näheren Ausarbeitung des Textes vollständig erschlossen hat, wenn es sich auch von Anfang an sinnvoll *anfühlte*. Das Nachfolgende ist daher eher ein Erklärungsversuch, als eine von Beginn an feststehende Methode. Der Grund, warum ich die Methode überhaupt noch angebe ist rein pragmatisch: Vielleicht habe ich Fehler gemacht. Und falls dem so ist, kann über die Angabe der Methode möglicherweise herausgestellt werden, was ich falsch gedacht habe bzw. kann sie dabei helfen, die Interpretation leichter zu verstehen und gegebenenfalls manches nachzubessern oder zu verändern.

Für mich ist die große Gemeinsamkeit bei Alexander und dem Commoning ein im Zentrum stehender *problematischer innerer Zustand*, welcher nur durch eine *Veränderung der äußeren Welt* gelöst werden kann. Diese Veränderung der äußeren Welt, sprich: die Problemlösung, kann durch Alexanders Methode *in Einzelteile zerlegt* werden. Jedes dieser einzelnen Teile kann *für sich stehen* und als *Muster* beschrieben werden. In Bezug auf den jeweiligen Gegenstand kann einem solchen Muster eine *bestimmte Qualität* zur Problemlösung von *allgemeiner Gültigkeit* innerhalb *eines bestimmten Kontextes* zugeschrieben werden. Weiter kann ein solches Muster, unabhängig von seiner Qualität, Teil *verschiedener Problemlösungen* sein. Und noch weiter und zu Ende gedacht, kann *die Lösung sämtlicher wiederkehrender Probleme* (im Rahmen des jeweiligen Gegenstandes) durch eine Kombination solcher Muster beschrieben werden. Diese Muster dienen damit einerseits als Wissensspeicher und andererseits auch als Kommunikationsmittel. Und wieder explizit auf das Commoning bezogen, ermöglichen diese beiden Aspekte erst, dass die *Organisation der Tätigkeiten zur gemeinsamen Veränderung der Welt* auf Augenhöhe funktionieren kann.

Deswegen, ich wiederhole mich, empfinde ich die Software als so grundlegend notwendig um bestehende Herrschaftsverhältnisse auf emanzipatorische Weise überwinden zu können. Weitere sich aus der Struktur heraus ergebende Momente, wie etwa die *Bewegungstendenz zu immer weniger, aber hochwertigeren Mustern*, welche im Commoning einhergeht mit einer geringeren Zahl insgesamt verwendeter Mittel, sprich: *Modularität* wie sie auch von der „Open Source Ecology“ angestrebt wird, legen mir nahe, dass die Interpretation sinnvoll war. Der Text unterliegt dabei einer Creative-Commons-Lizenz (CC BY-NC-SA) und darf damit ohne weitere Nachfrage mit Namensnennung verbreitet und verändert werden, so lange der Text bzw. das daraus hervorgehende Werk nicht kommerziell verwendet wird und denselben Bedingungen unterliegt.

# 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



**Qualität** (S.5): Ein mittlerer Wert der sinnlich-funktionalen Qualität und der Prozessqualität eines Tätigkeitsmusters

**Prozess-** (S.7): Die Prozessqualität besteht sowohl aus dem *Aufwand der bestimmten Tätigkeit*, als auch dem Aufwand, welcher (spekulativ) zur Durchführung der Tätigkeit im *lokalen Kontext* notwendig wird.

**sinnlich-funktionale** – (S.7): Die sinnlich erfahrbare Qualität des *Resultates* eines Tätigkeitsmusters u. seine Zweckdienlichkeit

- **ohne Namen**: Der bei C. Alexander über die Anwend. von Mustern zu erreichende innere, „anspannungslose“ Zustand.

- Im Commoning (S.4): Zustand der vollständigen Bedürfnisbefriedigung

**Reparaturprozess** (S.24): Manuelle Änderung von *Konfigurationen* durch die daran Beteiligten.

**Re-Produktionsprozess**: *Produktion* (im Kapitalismus: bezahlte Arbeit zur Geldverwertung) und *Reproduktion* (im Kapitalismus: meist unbezahlte (Fürsorge-)Arbeit zum Erhalt der in der Produktion tätigen Personen) als ein Prozess gedacht.

**Selbstzuordnung** (S.19): Zentrales Prinzip sich in das *ununterbrochene Commoning* einzubringen. Tätigkeiten werden im Rahmen des *Konfigurations-* und *Reparaturprozesses* den Beteiligten vorgeschlagen, welche sich schließlich einzelnen Tätigkeiten daraus annehmen.

**Softwarefunktionen**:

- Angabe von Kontinuität: S.10
- vorsorg. Reservierung von Mitteln: S.10
- Ortsveränd. im Konfig.-prozess: S.21
- Mengenzuordnung: S.21
- Kommunikation: S.23
- Gruppen-Kontinuität: S.24
- Analyse: S. 26
- Transparenz von Absprachen: S.20
- autom. Zeitplanung: S.28
- vorsorgende Selbstzuordnung: S.28
- Auflösung von int. Zusammenschl.:
- De-/Reaktivierung von Commons: S.29
- Autorisierung zur Teilhabe: 33

**Sortierung** (S.22): Reihenfolge, in welchen Anwender\*innen freigeschaltete Tätigkeitsmuster des *Konfigurationsprozesses* vorgeschlagen werden.

**Spannung** (S.5): Vom lokalen Kontext abhängiger Aufwand zwischen dem Bedürfnis und seiner Befriedigung.

- Auflösung/Entstehung (S.7): Eine Spannung wird durch Tätigkeiten aufgelöst. Die *sinnlich-funktionale Qualität* einer Tätigkeit zeigt an, inwieweit sie eine bestimmte Spannung auflöst. Die *Prozessqualität* einer Tätigkeit zeigt an, inwiefern neue Spannungen durch ihre Ausführung entstehen.

**Störung** (S.28): Geschieht, wenn einer Tätigkeit in einem aktivierten Commoning-Prozess nicht bzw. nur problematisch (zeitverzögert oder mit niedriger sinnlich-funkt. Qualität) nachgegangen wird. Die Häufung von Störungen kann zur *Krise* führen

**Tätigkeitsmuster** (S.10): Beschreibt eine Tätigkeit, welche mit bestimmten Mitteln ein anderes Mittel erzeugt, (orts-)verändert oder erhält bzw. ein Bedürfnis direkt befriedigt.

- Rahmen (S.6): Der *Bedarf* den die Tätigkeit benötigt und das *Resultat*, welches daraus hervorkommt.

- Entstehung (S.9): Aus individueller Erfahrung heraus, welche in verarbeitbarer Form gesellschaftlich geteilt wird.

- Integration (S.10): Tätigkeitsmuster können in Beziehung gesetzt werden, wenn dasselbe Mittel Bedarf des einen und Resultat des anderen ist.

- Drei Aspekte der Beziehung (S.12): Das *Problem*, welches mit dem Resultat gelöst wird. Die *Lösung*, welche die Tätigkeit beschreibt. Der *Kontext*, in welchem die Tätigkeit auftritt.

**komplexe** - (S.11): Durch Fähigkeiten definierter Zusammenhang von Tätigkeitsmustern, denen sich „als ein“ Tätigkeitsmuster zugeordnet werden kann.

**Tätigkeitsgewicht** (S.16): Indikator der *Dringlichkeit* einer Tätigkeit durch Anzahl und Gewicht der Bedürfnisse, welche darauf verweisen

- Entstehung im Konfig.-prozess.: S.21

**Zusammenschluss ,integrierter** (S.29): Selbstorganisation mit eigener Regelsetzung, Tätigkeits- und Mittelkoordination als Teil des ununterbrochenen Commonings.

- Unterschiede zu Commons n. Ostr.: S.31

- Integration in die Softwarestr.: S. 31

- 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 Sutterlützi/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](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 eindrucklichsten 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

Wenn das Projekt noch nicht realisiert wurde, dann muss es noch realisiert werden. Hinweise zum derzeitigen Stand solltest Du im Netz finden.



Dieses Werk ist lizenziert unter einer  
Creative Commons Namensnennung – Nicht-kommerziell –  
Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz.

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.