

Marcus Meindel & das *Global Commoning System*-Projektteam

Ein Softwarekonzept für ununterbrochenes Commoning – Zweiter Teil: das Tätigkeitsmuster, Konfigurationen, Fähigkeiten, Qualifikation

Bevor wir uns weiter damit beschäftigen die unterstützende Vermittlungsform einer Gesellschaft nach Bedürfnissen, Fähigkeiten und Interessen zu konstruieren, will ich eines nochmal ganz deutlich sagen: Ja, es handelt sich um ein Softwarekonzept und ja, es richtet sich in erster Linie an Entwickler und Entwicklerinnen, da es letzten Endes auf eure Arbeit ankommt. Aber alles was die Software macht, ist reales Commoning zu *unterstützen*.

Die Software erzeugt hier nichts, was nicht sowieso schon da wäre. Wenn es im Verlauf dieses Kapitels um →*Fähigkeiten* geht und dass es sich dabei um →*Tätigkeitsmuster* handelt, die als *verinnerlicht* markiert werden, dann ist das nicht mehr oder weniger als die Abbildung eines realen Prozesses: Jemand macht etwas (geht einer bestimmten Tätigkeit nach), merkt, dass er oder sie das kann und geht anschließend der Tätigkeit immer wieder nach – so lange eben, bis er oder sie nicht mehr nachlesen muss, wie es funktioniert. Die Person *verinnerlicht* die Tätigkeit, wird sich darüber *bewusst* und gibt die Information entsprechend an die Software weiter. Hierdurch werden ihr zukünftig Tätigkeiten vermittelt, welche die *verinnerlichte Tätigkeit* voraussetzt. Was wir also machen ist, diesen realen Prozess in der Software abzubilden, damit die entsprechende Person mehr Möglichkeiten hat sich mit ihrer gewonnen Erfahrung in anderen Prozessen zu integrieren und sich somit entwickeln zu können.

Oder ein anderes Beispiel: Im späteren Verlauf der Textreihe wird es um *Softwarefunktionen zur Stärkung interpersonaler Beziehungen* gehen. Was ist die reale Situation? Über die →*Konfiguration* wird ersichtlich, welche konkreten Personen, die sich unabhängig voneinander Tätigkeitsmustern zugeordnet haben, im realen Prozess schließlich miteinander kooperieren. Das heißt, zwei oder mehr Personen kommen an einem Ort zusammen, um ein bestimmtes Projekt anzugehen und arbeiten einige Tage gemeinsam daran. Womöglich entstehen dabei Freundschaften oder zumindest Bekanntschaften – man kennt sich und würde ein ähnliches Projekt vielleicht auch zu einem späteren Zeitpunkt wieder zusammen angehen. Was über die Software jetzt möglich wird, ist dass diese Beziehung zwischen den beiden Personen und den Tätigkeiten, die sie ausgeführt haben, gespeichert werden kann und falls das gleiche Problem wieder auftritt, das sie bereits gemeinsam gelöst haben, können wieder diese konkreten Personen *gemeinsam* automatisch angefragt werden. Das ist nichts anderes, als wenn eine von beiden gehört hätte, das es irgendwo etwas zu tun gibt und dann die andere anschreibt, ob sie nicht Lust hat, wieder mit anzupacken – nur eben vereinfacht bzw. durch die Software *unterstützt*.

Aber genug davon und so weit sind wir noch lange nicht. In diesem Teil geht es überhaupt erst einmal um das *Tätigkeitsmuster*, um *Konfigurationen*, um *Fähigkeiten* und schließlich um die Notwendigkeit, *Qualifikationen* zu definieren.

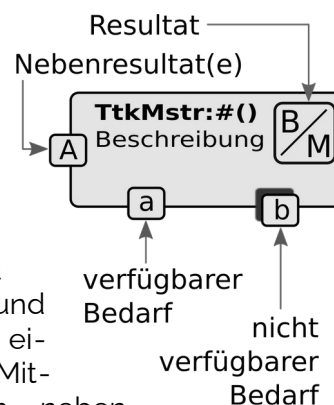


Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz (CC-BY-NC-SA)

Das Tätigkeitsmuster

Die Software selbst unterstützt Prozesse zur allgemeinen Bedürfnisbefriedigung. Das heißt, es gibt Bedürfnisse die befriedigt bzw. - von einem anderen Standpunkt betrachtet – Leiden die gelindert werden wollen. Oder noch ein wenig allgemeiner formuliert: Es gibt Probleme, die gelöst werden müssen. Und um diese Probleme zu lösen muss etwas getan werden. Und wenn wir etwas machen, dann brauchen wir oft Dinge dafür und wenn wir diese Dinge nicht haben, dann ist das ein weiteres Problem, das gelöst werden muss. Ein *Tätigkeitsmuster* beschreibt dabei, was getan werden *kann*, um ein bestimmtes Problem zu lösen und welche Dinge bei dieser *Möglichkeit einer Lösung* benötigt werden.

Ein Tätigkeitsmuster ist dabei eine *individuelle Erfahrung*, welche *gesellschaftlich geteilt* und damit *frei verfügbar* gemacht wird. Auf diese Weise kann auch seine Entstehung verstanden werden: Jemand vermittelt den Bedarf nach einem bestimmten Mittel und eine andere Person weiß, wie sie dieses Mittel verfügbar machen kann. Diese Person besorgt sich, was dafür notwendig ist, geht der Tätigkeit nach und deckt schließlich den Bedarf zur Zufriedenheit der Person, welche ihn vermittelt hat. Die Person, welche die Tätigkeit ausgeführt hat, beschreibt schließlich wie sie bei der Tätigkeit vorgegangen ist und was sie dafür benötigt hat in einer genormten Form und speist sie in einer Datenbank ein. Wenn anschließend der Bedarf nach demselben Mittel noch einmal vermittelt wird, kann dieses Muster abgerufen werden – neben all den anderen Mustern mit demselben Resultat – und alle anderen Anwender*innen der Software können auf diese individuell Erfahrung zurückgreifen und – die benötigte Qualifikation vorausgesetzt – denselben Bedarf ebenfalls decken, falls ihnen die notwendigen Mittel zur Verfügung stehen.



Tätigkeitsmuster sind dabei Kernelemente des ununterbrochenen Commonings. Wir bauen auf dem Gedanken auf, dass sehr viele Tätigkeiten zur Lösung bestimmter Probleme im gesellschaftlichen Kooperationsprozess nicht einmalig sind, sondern sich in gleicher Form wiederholen. Der *Rahmen* eines Tätigkeitsmusters besteht aus dem *Resultat* einer Tätigkeit – also dem, was nach der Tätigkeit verfügbar ist oder durch die Tätigkeit passiert – und dem *Bedarf* einer Tätigkeit. Der Bedarf sind alle Mittel, die zur Ausführung der Tätigkeit verfügbar sein müssen. Das Tätigkeitsmuster *enthält* schließlich eine möglichst genaue Beschreibung, wie der Bedarf angewendet werden muss, um das Resultat verfügbar zu machen.¹

Das Resultat vieler Tätigkeiten (insofern es keine direkte Bedürfnisbefriedigung ist) und der Bedarf jeder Tätigkeit, ist dabei immer ein Mittel, welches auf Softwareebene durch ein *Mittelmuster* bzw. *Muster eines Mittels* dargestellt wird und von dort auf konkret vorhandene Mittel verweist. Der Zusammenhang des Mittelmusters mit dem konkreten Mittel wird im sechsten Teil der Textreihe, *→Muster von Mitteln*, näher ausgearbeitet.

Neben Resultat, Bedarf und der Tätigkeitsbeschreibung ist jedes Muster noch mit weiteren Eigenschaften beschrieben, auf welche im Verlauf der Textreihe noch näher eingegangen werden wird. Für den *Konfigurationsprozess* wichtig ist dabei der *Aufwand* einer Tätigkeit (*→ Aufwandsbestimmung*), welcher zusammen mit den zur

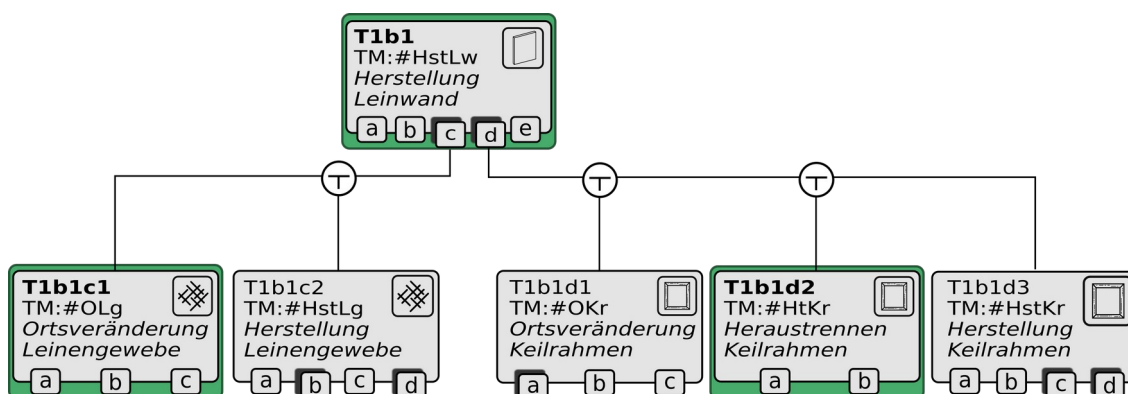
¹ Welche Form diese Beschreibung hat – ob Text, Audio, Video, etc. - ist dabei selbstverständlich nicht vorgegeben. Tätigkeitsmuster, wenn auch nicht in verarbeitbarer Form, finden sich heute bereits auf <https://de.wikihow.com/>

hat (→*Vermittlung von Bedürfnissen*), durch die Verwendung (Tätigkeit *T1*) von ‚Pin-seln‘ (M1a), einer ‚Leinwand‘ (M1b) und ‚Farbe‘ (M1c) befriedigt werden kann.

In diesem Beispiel wird die Person, welche diese erste Tätigkeit durchführt, wie selbstverständlich auch die Person sein, welche auch das Bedürfnis vermittelt hat. Selbst wenn diese Person das Bedürfnis vermittelt hätte, über ein Kunstwerk zu reflektieren, würde sie *die Tätigkeit des Reflektierens* selbst durchführen und das Kunstwerk wäre als Mittel ein Bedarf dieser Tätigkeit. Anders dagegen verhält es sich bei Tätigkeiten, die wir heute als „Dienstleistungen“ bezeichnen oder sich oft auch im Bereich der gegenseitigen Fürsorge finden lassen. Das Bedürfnis nach ‚körperlicher Hygiene‘ einer pflegebedürftigen Person könnte so etwa nicht durch diese Person selbst befriedigt werden. Falls es sich daher nicht um eine →*emphatische Bedürfnisvermittlung* handelt, und das bezieht sich wieder auf die konkrete Entwicklung der Software, steht die erste Tätigkeit einer Konfiguration nicht zur Selbstzuordnung frei, sondern wird automatisch durch die Person besetzt, welche das Bedürfnis auch vermittelt hat.

Der Bedarf zur Ausführung der im Tätigkeitsmuster beschriebenen Tätigkeit kann entweder *verfügbar* oder *nicht verfügbar* sein. Ob ein bestimmtes Mittel verfügbar ist, ist dabei sowohl abhängig von der *lokalen Umgebung* in welcher der Bedarf vermittelt wird und welche Mittel *dort* den entsprechenden Nutzungsbedingungen nach für die Tätigkeit verwendet werden dürfen und weiter abhängig von der konkreten Person, welche sich der Tätigkeit zuordnet und welche Mittel *ihr persönlich* zur Verfügung stehen. Erst nach einer Selbstzuordnung zu einer Tätigkeit kann also festgestellt werden, welcher Bedarf vorhanden ist und welcher Bedarf noch zur Verfügung gestellt und daher vermittelt werden muss.

Jeder Bedarf kann dabei grundsätzlich durch verschiedene Tätigkeiten gedeckt werden. Hierfür sehen wir uns das Tätigkeit zur ‚Herstellung einer Leinwand‘ näher an, welche auf das Tätigkeitsmuster #HstLw verweist. Selbstverständlich ist das auch nur eine von vielen Möglichkeiten, eine ‚Leinwand‘ verfügbar zu machen. Im Tätigkeitsmuster angegeben sind dabei fünf Bedarfe: (a) ein Hammer, (b) Nägel, (c) Leinengewebe, (d) Keilrahmen und (e) Spannvorrichtung.³ Wir nehmen an, dass die Mittel ‚Hammer‘, ‚Nägel‘ und ‚Spannvorrichtung‘ der Person, welche sich der Tätigkeit annimmt, zur Verfügung stehen und die Mittel ‚Leinengewebe‘ und ‚Keilrahmen‘ nicht.



³ Ein notwendiges Mittel für die meisten Tätigkeiten ist ein entsprechender Raum. Um diesen nicht ständig neu aufzuführen und somit den Lesefluss zu behindern, wird in der gesamten Textreihe davon abstrahiert. Eine Ausnahme ist das Kapitel →*Momente des Gesamtaufwandes*

Den Bedarf nach den beiden Mitteln kann die Person selbstverständlich auch außerhalb der Software an andere Personen oder andere Strukturen vermitteln. Wichtig wäre es daher, dass es innerhalb der Software die Funktion gibt anzugeben, wann mit der Verfügbarkeit der Mittel zu rechnen ist, um eine Transparenz des Kooperationsprozesses sicherzustellen. Hier nehmen wir jedoch weiter an, der Bedarf nach ‚Leinengewebe‘ und ‚Keilrahmen‘ wird über die Software vermittelt.

Das ‚Leinengewebe‘ kann zum Beispiel über eine Ortsveränderung von bestehender Leinwand verfügbar gemacht werden (Tätigkeitsmuster #OLg, Tätigkeit T1b1c1). Das entsprechende Muster wäre „Ortsveränderung von Leinengewebe durch (a) Leinengewebe, (b) Fahrzeug und (c) Treibstoff“⁴. Eine andere Möglichkeit ‚Leinengewebe‘ verfügbar zu machen, wäre selbstverständlich die Herstellung davon. Das in der Grafik verwendete Tätigkeitsmuster #HstLg (T1b1c2) heißt entsprechend: „Herstellung von Leinengewebe durch (a) unbespannten Webstuhl, (b) Scherbaum, (c) Leinengarn und (d) Leinwand-Bindungspatrone“.

Von hier ab weiter zu dem Bedarf an ‚Keilrahmen‘ (M2c): Eine Möglichkeit den Bedarf zu decken kann wieder die Ortsveränderung eines bereits vorhandenen Mittels sein (#OKr, T1b1d1). Eine andere Möglichkeit wäre den ‚Keilrahmen‘ aus einer zur Verfügung stehenden (a) ‚benutzten auf Keilrahmen gespannten Leinwand‘ mit einem (b) ‚Messer‘ her auszutrennen (#HtKr, T1b1d2). Schließlich kann der Bedarf nach Keilrahmen natürlich auch über deren Herstellung gedeckt werden. Das Tätigkeitsmuster #HstKr (T1b1d3) hieße entsprechend: „Herstellung eines Keilrahmens durch (a) Winkelsäge, (b) Holzleisten, (c) Nägel und (d) Werkbank.“

Wenn im Beispiel jeder Bedarf der ausgewählten Tätigkeiten (T1b1c1 und T1b1d2) verfügbar ist, sich zu diesen Tätigkeiten jemand zugeordnet hat und auch sonst kein offener Bedarf innerhalb der Konfiguration ansteht, kann die Konfiguration – die Anordnung und Auswahl der Strukturelemente – festgesetzt werden (→*Festsetzung von Konfigurationen*) und der Kooperationsprozess kann von den jeweils letzten Tätigkeiten her beginnen. Damit sich dabei Konfigurationen durch den Prozess der Selbstzuordnung herausstellen können, die im jeweiligen lokalen Umfeld sinnvoll und effektiv sind, ist einer der wesentlichen Funktionen der Software. Dieser →*Konfigurationsprozess* ist Schwerpunkt des dritten Teils der Textreihe.

⁴ Die „Ortsveränderung“ ist dabei ein besonderes Muster, da das Resultat der Tätigkeit gleich einem Bedarf der Tätigkeit ist und sich nur die Lokalität des Mittels verändert.

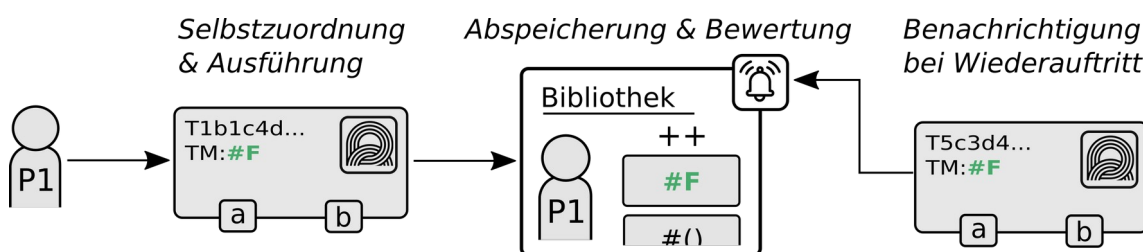
Der individuelle Musterspeicher

Der ‚individuelle Musterspeicher‘ ist selbst keine Funktion, sondern weist auf zwei verschiedene Ebenen der Softwarevermittlung hin: Einmal den (*Tätigkeits-*)*Musterspeicher* einer konkreten Person innerhalb der Software, was nachfolgend als ‚*Bibliothek*‘ bezeichnet wird. Und einmal auf das *persönliche Wissen* einer konkreten Person außerhalb der Software, das heißt die *Erfahrungen* zur Ausführung von Tätigkeiten, die sie im Laufe ihres Lebens gemacht hat und welche sie einerseits *allgemein verfügbar* machen und andererseits als *Fähigkeit* innerhalb der Software angeben kann.

Im Sinne der Übersichtlichkeit werden die

Bibliothek

Die ‚Bibliothek‘ ist ein Werkzeug, um Beteiligte darin zu unterstützen, sich Tätigkeiten anzunehmen, die in ihrem eigenen Interesse sind. Die Bibliothek ist dabei eine persönliche Sammlung von Tätigkeitsmustern. Ein Tätigkeitsmuster kann dabei *manuell* gesucht und schließlich in der eigenen Bibliothek gespeichert werden oder das Tätigkeitsmuster wird *automatisch* in die Bibliothek übertragen, nachdem die Person sich bereits mindestens einmal einer entsprechenden Tätigkeit zugeordnet hat. Die Tätigkeitsmuster können in der Bibliothek entsprechend markiert werden, ob sich der Tätigkeit *gerne* oder *nicht gerne* angenommen wird. Je nachdem kann die Person durch die Software *benachrichtigt* werden, sobald eine Tätigkeit im lokalen Umfeld *vorgeschlagen* wird (→ *Konfigurationsprozess*), an welcher Interesse besteht oder die Person wird, wenn überhaupt, erst benachrichtigt, wenn die Tätigkeit eine hohe →*Wichtigkeit* hat.



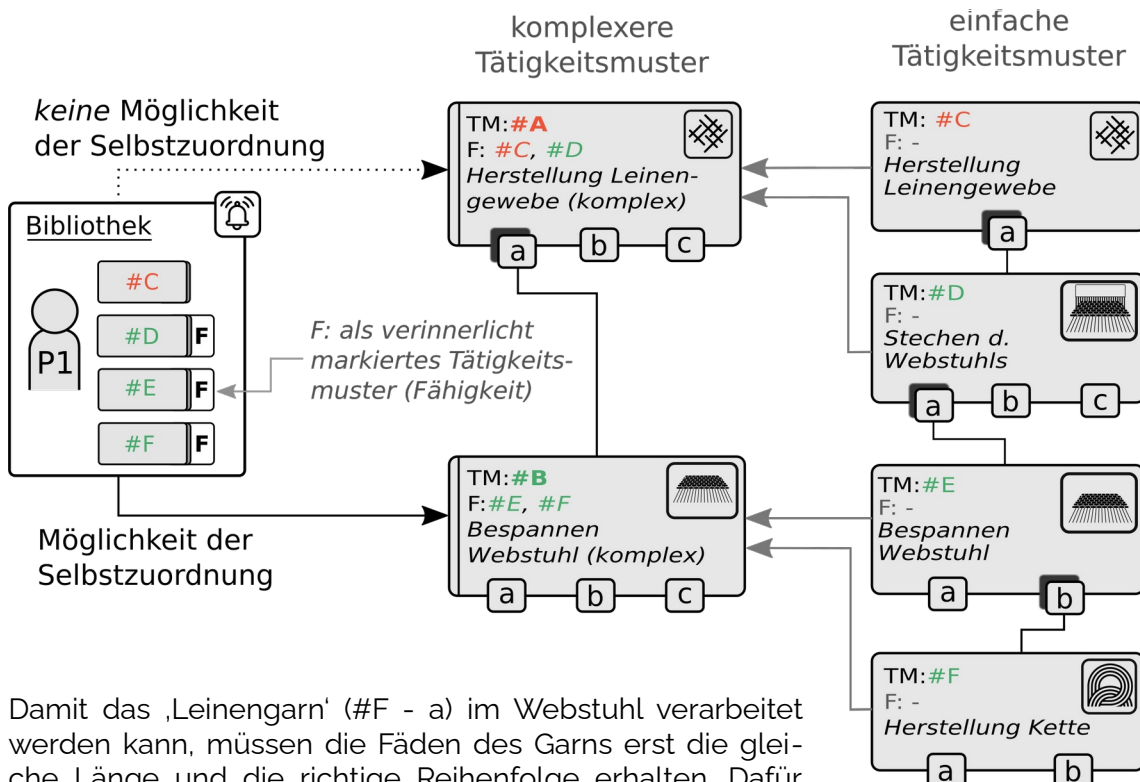
Fähigkeiten

Kurze Anmerkung: Fähigkeiten sind zwar als Funktion der Software notwendig, aber nicht unbedingt zum Verständnis der grundlegenden Struktur des Softwarekonzeptes. Wird die Textreihe chronologisch gelesen, kann dieses Kapitel übersprungen werden.

In der ‚Bibliothek‘ gibt es weiter die Möglichkeit, einzelne Tätigkeitsmuster als eigene *Fähigkeiten* zu definieren. Im Gegensatz zu den im nächsten Kapitel beschriebenen Qualifikationen, sind Fähigkeiten dabei auf Softwareebene keine Bedingungen, um bestimmte Tätigkeiten ausführen zu können. Die Definition von Fähigkeiten hilft Beteiligten vielmehr dabei, den Gesamtprozess übersichtlicher zu gestalten und die Selbstzuordnung zu vereinfachen. Fähigkeiten werden dabei als Tätigkeiten verstanden, deren Ablauf *verinnerlicht* ist. Es sind also Handlungen, die Beteiligte prinzipiell ohne die Beschreibung im jeweiligen Tätigkeitsmuster durchführen können bzw. nach eigener Einschätzung wissen, was bei den entsprechenden Problemen getan

werden muss. Die entsprechenden Tätigkeitsmuster können daher durch Beteiligte ihrem eigenen Ermessen nach in der Bibliothek als Fähigkeiten markiert werden.

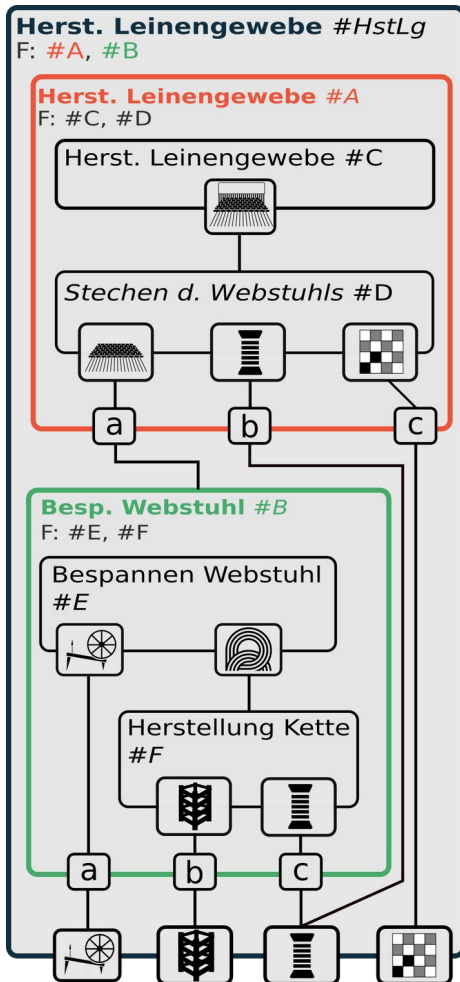
Durch diese Angabe, dass bestimmte Tätigkeiten *verinnerlicht* sind, können sich Beteiligte schließlich *komplexeren Tätigkeitsmustern* annehmen. Um das verständlich zu machen, wird weiter auf die Produktion von Leinengewebe zurückgegriffen, welche auch Marx in seiner Wertformanalyse verwendet hat. Die nachfolgende Darstellung der Herstellung von Leinengewebe ist dabei stark vereinfacht und im Sinne der Übersichtlichkeit werden die Tätigkeitsmuster von ihrem Inhalt unabhängig als #A-F bezeichnet.



Damit das ‚Leinengarn‘ (#F - a) im Webstuhl verarbeitet werden kann, müssen die Fäden des Garns erst die gleiche Länge und die richtige Reihenfolge erhalten. Dafür wird es auf den ‚Scherbaum‘ (#F - b) aufgespannt. Das so bearbeitete Leinengarn wird schließlich als ‚Kette‘ bezeichnet. Diese ‚Kette‘ (#E - b) wird schließlich auf den bisher ‚unbespannten Webstuhl‘ (#E - a) gespannt. Im Webprozess werden aus dieser Kette die vertikal verlaufenden Fäden des Gewebes entstehen. Ein weiteres ‚Leinengarn‘ (#D - b) wird schließlich gemäß einer ‚Leinwand-Bindungspatrone‘ (#D - c) durch die Litzen und Blätter des ‚mit Kette bespannten Webstuhls‘ (#D - a) gestochen. Die Leinwand-Bindungspatrone gibt dabei vor, wie die Fäden durch den Webstuhl verlaufen müssen, damit am Ende das gewünschte Gewebemuster herauskommt. Durch den damit vollständig ‚bespannten Webstuhl‘ (#C - a) kann das ‚Leinengewebe‘ hergestellt werden.

In der Grafik ist die Bibliothek einer Person zu sehen, welche sich sämtliche gerade aufgelisteten Tätigkeiten in ihrer Bibliothek gespeichert hat und diese Person könnte bei jeder einzelnen Tätigkeit durch die Software entsprechend benachrichtigt werden, sobald diese vorgeschlagen wird. Nur drei der vier Tätigkeitsmuster, auf welche die vorgeschlagenen Tätigkeiten verweisen, sind dabei auch als *verinnerlicht*, also als eigene ‚Fähigkeiten‘ (F) markiert. Weiter sind in der Grafik zwei ‚komplexere Tätigkeitsmuster‘ zu sehen, welche jeweils zwei der einfachen Tätigkeitsmuster als Fähigkeiten voraussetzen. Statt sich jetzt den beiden, durch die Tätigkeitsmuster #E und

#F beschriebenen Tätigkeiten einzeln zuzuordnen, könnte sich die entsprechende Person auch einer Tätigkeit zuordnen, die durch das komplexe Tätigkeitsmuster #B beschrieben wird. Dagegen könnte sie sich *nicht* einer Tätigkeit zuordnen, die durch das komplexe Tätigkeitsmuster #A beschrieben wird, da sie eine der beiden als Fähigkeiten vorausgesetzten Tätigkeitsmuster – das Tätigkeitsmuster #C – nicht in ihrer Bibliothek als verinnerlicht markiert hat.



Ein komplexeres Tätigkeitsmuster besteht dabei *ausschließlich* aus der Kombination einfacherer Tätigkeitsmuster. Diese einfacheren Tätigkeitsmuster selbst – die Bezeichnungen sind relativ zueinander – könnten dabei aber selbst wiederum komplexere Tätigkeitsmuster sein, mit denen einfache Tätigkeitsmuster zusammengefasst werden. Wesentlich ist, dass bei komplexeren Tätigkeitsmustern keine neuen Informationen hinzukommen und auf „unterster Ebene“ jeder Schritt genau beschrieben ist. Falls eine neue Information doch notwendig sein sollte, dann muss diese zuerst in Form eines *Tätigkeitsmusters* angelegt und schließlich ebenfalls als *Fähigkeit* des komplexeren Tätigkeitsmusters definiert werden.

Da das im Kapitel →*Konfigurationen* vorgestellte Tätigkeitsmuster ‚Herstellung von Leinengewebe durch (a) unbespannten Webstuhl, (b) Scherbaum, (c) Leinengarn und (d) Leinwand-Bindungspatrone‘ (#HstLg) denselben Bedarf hat wie die beiden zusammenhängenden komplexeren Tätigkeitsmuster #A und #B zusammen, kann #HstLg selbst als komplexeres Tätigkeitsmuster gefasst werden, welches die beiden anderen komplexeren Tätigkeitsmuster als Fähigkeiten voraussetzt. Über *Fähigkeiten* werden also Tätigkeiten auf Softwareebene in kleinere Schritte geteilt bzw. werden einzelne Tätigkeiten zusammengefasst. Da prinzipiell jedes Tätigkeitsmuster als Fähigkeit markiert werden kann, kann sich prinzipiell auch jedem komplexen Tätigkeitsmuster zugeordnet werden. Vorausgesetzt ist hier jeweils, dass die definierten *Qualifikationen* der darin enthaltenen Tätigkeitsmuster gegeben sind. Die Definition von Fähigkeiten hilft dabei den Beteiligten den Gesamtprozess einerseits schneller erfassen und sich leichter einbringen zu können, andererseits den organisatorischen Aufwand zu minimieren, welcher mit der Selbstzuordnung zu vielen kleineren Tätigkeiten unbedingt zusammenhängt.

Hierzu eine kurze Anmerkung zu den →*Details von Tätigkeitsmustern*: Dasselbe Tätigkeitsmuster hat im besten Fall eine Vielzahl von unterschiedlichen Beschreibungen für denselben Prozess. Die Sprache ist dabei natürlich ein wesentlicher Punkt, aber genauso die Form der Darstellung (als Textbeschreibung, Video, etc.) oder etwa der Detailreichtum der Beschreibung. Gibt es etwa eine zusätzliche sehr knappe Beschreibung eines Musters und ist diese entsprechend definiert, kann die Beschreibung eines komplexen Tätigkeitsmusters *automatisch generiert* werden, indem diese knappen Beschreibungen aneinander gehängt werden.

Da ein komplexes Tätigkeitsmuster lediglich eine Verschachtelung von einfacheren Tätigkeitsmustern ist, ist sowohl der Bedarf als auch Aufwand (→*Der Aufwand und die Einheit des Konfigurationsprozesses*) genau gleich. Der Aufwand eines komplexen Tätigkeitsmusters ist daher auch die Summe des Aufwandes der darin enthaltenen einfachen Tätigkeitsmuster. Und da ein Tätigkeitsmuster immer die eindeutige Beschreibung eines Prozesses ist, sind weiter auch die Tätigkeitsmuster, welche als dessen Fähigkeiten definiert sind, immer eindeutig. Einzelne Tätigkeitsmuster können dabei als Fähigkeiten in unterschiedlichen anderen (komplexeren) Tätigkeitsmustern definiert sein.

Qualifikation

Während Fähigkeiten frei gesetzt werden können, braucht es bei der *Qualifikation* eine äußere Instanz, welche die Erlaubnis erteilt, sich bestimmten Tätigkeiten überhaupt annehmen zu dürfen. Notwendig kann das in unterschiedlichen Situationen sein: Eine Tätigkeit kann den Umgang mit sensibler Technik beinhalten, deren Beschädigung einen großen Aufwand zur Wiederinstandsetzung nach sich ziehen würde. Oder die Tätigkeit ist an sich nicht schwierig, aber beinhaltet gefährliche Chemikalien, welche bei fehlerhafter Handhabung zu erheblichen Gesundheitsproblemen führen können. Genauso sollte sichergestellt werden können, dass sich bei Tätigkeiten an Menschen – wie etwa in der Chirurgie oder der Erziehung – nur Personen mit dem notwendigen medizinischen oder pädagogischen Verständnis zuordnen können. Aber auch im Bereich der Produktion symbolischer Mittel kann es problematisch sein, wenn Personen sich Tätigkeiten zuordnen, welche ihre eigene Kompetenz überschreiten: Der ganze Prozess einer Bedürfnisbefriedigung kann aufgehoben werden, wenn etwa ein Programmierer sich selbst überschätzt und die Tätigkeit anderer Personen auf dessen Code aufbaut.

Die Definition von notwendigen Qualifikationen für Tätigkeitsmuster scheint daher unerlässlich, wirft aber Fragen auf, welche Autorität und welche Strukturen die Qualifikationen innerhalb einer Gesellschaftlichkeit von Gleichrangigen möglichst allgemeingültig zertifizieren können. Innerhalb dieser Softwarekonzept-Reihe sollen diese Fragen nicht beantwortet werden. Was auf Vermittlungsebene nur gesagt werden kann: Es ist möglich – und notwendig – bestimmte Tätigkeitsmuster mit einer *allgemeingültigen* Qualifikation als Voraussetzung dafür zu beschreiben. Auf Seite der Beteiligten muss es in diesem Fall eine *Bibliothek von Qualifikationen* geben, in welcher die eigenen Qualifikationen festgehalten sind. Auf Ebene von Gleichrangigen ist außerdem möglich, dass Beteiligte Anfragen zum Erlernen bestimmter Qualifikationen stellen und diese Anfragen durch Personen ausgelesen werden können, welche die Qualifikation besitzen und autorisiert sind – in welcher Weise auch immer – diese Qualifikation weiterzugeben.

Stand: 8. April 2021

Alle Icons wurden unter einer Creative-Commons-Lizenz (CC-BY) durch das ‚Noun-Project‘ (thenounproject.com) zur Verfügung gestellt. In diesem Textteil neu verwendet wurden: „Linen“ von Rineesh, „wooden frame“ von Tomek Woloszyn, PL, „alarm“ von Larea, „Thread“ von Anton, „weave“ von Claire Jones, „yarn“ von Janira Keana und „spinning Wheel“ von Jessie White. Vielen Dank.