

Die bedenkliche Dienlichkeit und Sicherheit von Softwaresystemen und die erlebte Verlässlichkeit

Cecile K. M. Crutzen¹, Hans-Werner Hein²

¹⁾ Open Universiteit Nederland, Heerlen, cecile.crutzen@ou.nl

²⁾ Verlässliche IT-Systeme, Köln, hwh@acm.org

Zusammenfassung:

Es wird eine Ontologie des Interagierens von Menschen mit Softwaremethoden und Softwaresystemen vorgestellt, in der Handlungen die semiotischen Elemente sind. Innerhalb dieser Ontologie werden die Begriffe Dienlichkeit, Sicherheit und Verlässlichkeit diskutiert. Sicherheit und Dienlichkeit können einen Teil der Verlässlichkeit verbergen, die Benutzer von Softwaremethoden und Softwaresystemen erleben sollten.

Schlüsselwörter: Dienlichkeit, Sicherheit, Verlässlichkeit, Objektorientierung, Common Criteria

"Denn sobald die sprechenden Menschen in grösseren Gruppen zusammenleben und sich nicht nur an Sprachhäuser, sondern auch an gebaute Häuser binden, geraten sie ins Kraftfeld der sesshaften Seinsweisen. Sie lassen sich nunmehr nicht mehr nur von ihren Sprachen bergen, sondern auch von ihren Behausungen zähmen. ... Wo Häuser stehen, dort muss entschieden werden, was aus den Menschen, die sie bewohnen, werden soll; es wird in der Tat und durch die Tat entschieden, welche Arten von Häuserbauern zur Vorherrschaft kommen." [1]

1. Menschen und Interaktion

Interaktion zwischen Menschen ist das gegenseitige Präsentieren von Handlungen. Die Präsentation von Handeln veranlaßt einen bedeutungskonstruierenden Prozeß bei den in der Interaktion beteiligten Aktoren. Der wahrnehmende Akteur interpretiert das angebotene Handeln und verwirklicht diese Interpretation durch eigenes Verhalten [Bild 1]. Der präsentierende Akteur interpretiert die Wirkung des eigenen Handelns. Jede Interpretation von Handeln wird für die anderen in der Interaktion beteiligten Aktoren wahrnehmbar durch sein Verhalten und die damit verbundenen Änderungen. Dieses zukunftsgerichtete Verhalten der Fürsorge und des Besorgens ist ein Umgang der Aktoren miteinander und enthält auch die "defizienten Modi des

Unterlassens" [2], denn diese gehören auch zu den möglichen Interpretationen.

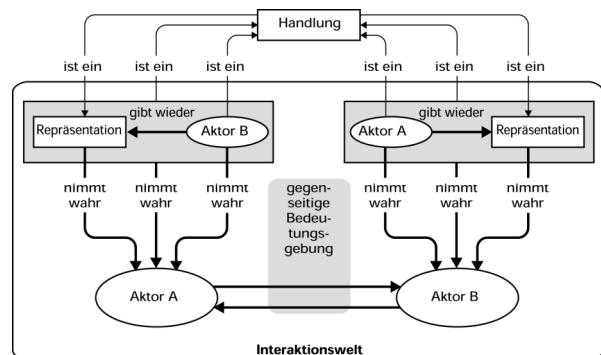


Bild 1: Interaktion: Präsentation und Interpretation von Handeln

Durch wiederholte gegenseitige Präsentation und Interpretation wird ein Raum möglichen Handelns eröffnet. Stuart Hall spricht davon, daß ein "meaningfull discourse" bereits entsteht, wenn es für den wahrnehmenden Akteur beim angebotenen Handeln überhaupt möglich ist, eine eigene Bedeutung zu konstruieren [3]. Ein solcher Raum ist nicht konfliktlos, denn angebotenes Handeln braucht bei der Interpretation nicht die Bedeutung zu bekommen, die der Produzent dieses Handelns intendiert hat. Die Aktoren werden im Moment ihres Handelns einerseits aus bisheriger Erfahrung, andererseits zukunftsgerichtet, mit einer bestimmten Wirkungsintention handeln.

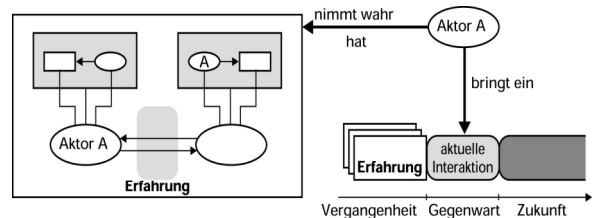


Bild 2: Einbringen von Erfahrung in die aktuelle Interaktion

Sie konstruieren zu gleicher Zeit sowohl für die aktuelle Interaktion eine neue Bedeutung, als auch für die früheren Interaktionserfahrungen, die sie in die aktuelle Interaktion einbringen [Bild 2]. Menschliche Akteure erfahren in einer Interaktion einen anderen Akteur als handelbar, wenn dessen Präsentation aus eigener Erfahrung interpretiert werden kann.

Dewey spricht davon, daß Menschen meistens aus Routine und Gewohnheit handeln. Gewohnheit entsteht, indem Menschen aus der Wirkung lernen, die Handlungen in bestimmten Situationen haben. Gewohnheiten haben aber die Gefahr, daß sie zu Routine werden, und daß Menschen dann solches Handeln nicht mehr ändern können: "*Fixity of habit may mean that something has a fixed hold upon us, instead of our having a free hold upon things. (...) Habits reduce themselves to routine ways of acting, or degenerate into ways of action to which we are enslaved just in the degree in which intelligence is disconnected from them. (...) Routine habits, and habits that possess us instead of our possessing them, are habits which put an end to plasticity. They mark the close of power to vary.*" [4]

Der Unterschied zwischen gewohntem Handeln und Routine ist, daß Gewohnheiten änderbar bleiben und losgelassen werden können. Gewohntes Handeln ist in diesem Sinn "verlässlich", weil dieses Handeln verhandelbar bleibt. In der Situation des aktuellen Handelns miteinander kann man erleben, daß man gewohntes Handeln loslassen kann, oder sich darauf verlassen kann. Wenn aber menschliche Akteure ihr Verhalten nicht zum Teil auf Routine und Gewohnheit bei sich selbst und anderen basieren könnten, dann würde ihnen ihr Leben nicht mehr erlebbar, weil sie jeden Moment gezwungen wären, so viel zu entscheiden, daß sie wegen der Fülle und Komplexität der Entscheidungen gar nicht mehr handeln könnten: "*Imagine breaking down the distinction between the producers and the consumers of knowledge: we all come to learn what we all need to know. Clearly such an ideal is unworkable in those terms as soon as we need to know more than the barest basics about ourselves and the world. It is impossible that we could all come to learn for ourselves what we would have to know for our cars to run, our bread to be baked, our illnesses to be cured, our roofs to keep the rain out, our currency to be stable, and our airplanes to fly.*" [5]

Die Wirklichkeit und die Wirkung des Handelns enthalten somit immer das Selbstverständliche und Stabilisierte, welches durch Wiederholung der Dienlichkeit entsteht: "*Die Dienlichkeit des Zeugens ist jedoch nur die Wesensfolge der Verlässlichkeit. Jene schwingt in dieser und wäre ohne sie nichts. Das einzelne Zeug wird*

abgenutzt und verbraucht; aber zugleich gerät damit auch das Gebrauchen selbst in die Vernutzung, schleift sich ab und wird gewöhnlich." [6]

Routine und Gewohnheit können aber in Interaktionswelten, durch Wiederholung und Imitation, zu Regeln und Traditionen führen, wobei der repräsentierende Akteur nicht mehr wahrgenommen wird und das Handeln dieses Akteurs als zuhänden aufgefaßt (gedankenlos benutzt) wird. Verlässlichkeit kann aber nur erhalten bleiben, wenn die Wirklichkeit und die Wirkung des Handelns auch das Verhandelbare enthält. Verlässlichkeit beruht auf dem Verhandeln, das ein menschlicher Akteur mit anderen Akteuren durchführen kann. Sie kann nur entstehen, wenn zwischen menschlichem Akteur und anderen Akteuren ein kritischer transformativer Raum existiert, worin gegenseitige Handelbarkeit entsteht. Mittels Handeln wird im Wesentlichen verhandelt über zukünftiges gegenseitiges Handeln. Weil jeder Akteur in diesem Raum sich selbst und anderen Akteuren ausgesetzt ist, existiert immer eine Geworfenheit, welcher der einzelne Akteur sich nicht entziehen kann und sehr oft auch nicht will. Innerhalb dieses Raums kann Verlässlichkeit nur dann erlebbar bleiben, wenn der Wiederholungsprozeß unterbrochen werden kann. Das gegenseitige Präsentieren und Interpretieren von Handeln sollte nicht immer ein reibungsloser Prozeß sein, bei dem Bedeutung nur durch Präsentieren konstruiert wird. Bedeutung sollte durch die Möglichkeit des Zweifels immer wieder aufs neue situiert konstruiert werden können. Die Entscheidung, wie ein Handeln interpretiert und welches Handeln präsentiert wird, gehört in den Bereich des Entwurfs, der Verwirklichung möglichen Verhaltens. Nach Heidegger gehört der Entwurf wesentlich zur Geworfenheit des Daseins: "*Und als geworfenes ist das Dasein in die Seinsart des Entwerfens geworfen. Das Entwerfen hat nichts zu tun mit einem Sichverhalten zu einem ausgedachten Plan, gemäß dem das Dasein sein Sein einrichtet, sondern als Dasein hat es sich je schon entworfen und ist, solange es ist, entwerfend. Dasein versteht sich immer schon und immer noch, solange es ist, aus Möglichkeiten.*" [7]

Entwurf bedeutet damit sowohl "Die Handlung des Änderns des Handelns" als auch "Die Präsentation des Handelns, das ändert". Ein Raum von Akteuren bleibt somit für einen bestimmten menschlichen Akteur verlässlich, wenn der Akteur aus der Erfahrung des Handelns selbst weiß, wann, wo und wie man sich auf das von anderen Akteuren angebotene Handeln verlassen kann (im Sinne von vertrauen), aber vor allem, wenn der Akteur weiß, wann das angebotene Handeln losgelassen und verlassen werden kann (im Sinne von allein lassen). Loslassen bedeutet, daß man das mögliche Angebot in

einer bestimmten Situation nicht zur Wirklichkeit kommen läßt, nicht mehr "wiederholt". Verlassen bedeutet, daß man sich auf den Weg macht, um innerhalb der Geworfenheit eigenes und anderes Handeln als Möglichkeiten zu erschließen. Dieses Erschließen setzt aber voraus, daß Zweifeln über Handeln eine Gewohnheit ist und daß eine Differenziertheit von Handeln bereitliegt [8].

2. Bedenkliche Dienlichkeit und Sicherheit

Ein Softwaresystem ist ein Akteur, der aus der Sicht des Benutzers Handlungen (re)präsentiert. Softwaresysteme werden softwaretechnisch so hergestellt, daß sie spezifizierten Dienlichkeits- und Sicherheitskriterien genügen. Sicherheit ist gerichtet auf das Erzwingen einer Vorbestimmtheit der Interpretation des bereitgelegtem Handelns. Dienlichkeit ist gerichtet auf die Präsentation desjenigen Handelns, das Benutzern in zweckbestimmter Weise nützlich sein könnte. Diese Dienlichkeit sollte bei der Interpretation nur durch die Präsentation des angebotenen Handelns selbstverständlich sein. Die Konzeptionen der Dienlichkeit und Sicherheit beruhen auf dem Zuhandenbleiben des Softwaresystems. Ein Softwaresystem wird als dienlich und sicher qualifiziert, wenn Benutzer die Unzuhandenheit der Software nicht erfahren brauchen. Wenn Software im Handeln zu oft unzuhanden ist, dann veranlaßt dies ein Nachdenken über das Zeug selbst und dies wird gesehen als eine nicht dienliche Ablenkung vom Nachdenken über das eigene Handeln. Diese Form von Sicherheit impliziert eine bedachte Dienlichkeit die nicht verhandelbar ist. In den zur Benutzung bereitgelegten Softwaresystemen wird dabei dem Aspekt der Verlässlichkeit nicht genügend Beachtung geschenkt, weil die Priorisierung der Dienlichkeit und Sicherheit die Verlässlichkeit verbirgt. Die Sicht der Hersteller ist es, daß Softwaresysteme Zeug sind, das so schnell wie möglich in Unauffälligkeit verschwinden sollte. Die Funktionalität wird basiert auf der Dienlichkeit einer Umhüllung, welche eine Suggestion von Vertrautheit und Selbstverständlichkeit erzeugen soll. Interfaces werden produziert als Simulation von Vertrautem. Solches Vertraute ist aber eine bedachte Sicherheit und Dienlichkeit, in der das mögliche Verhalten (Benehmen) des Softwaresystems hinter der Maske eines "benutzerfreundlichen Interface" verborgen wird. Das Interface verdeckt so die Möglichkeit des Loslassens und Verlassens. In diesem Sinn trägt die "bloße Dienlichkeit" des Zeugs, die durch die Wiederholung zur Routine wird, dazu bei, daß das Handeln "Loslassen und Verlassen" unmöglich wird, weil in der Routine das

Ändern des Handelns nicht mehr zum möglichen Handeln gehört. Die Interaktion von Mensch und bereitliegendem Handeln ist zum Bestand geworden, worin die gegenseitige Handelbarkeit erstarrt ist. So kann ein Softwaresystem unhandelbar werden. Bedachte Dienlichkeit und Sicherheit sind darauf gerichtet, daß man nicht mehr verhandelt, und werden dadurch bedenklich. Heidegger meint dazu, daß eine bloße Dienlichkeit die Verlässlichkeit verschwinden läßt, weil im alltäglichen Gebrauch das Interpretieren des angebotenen Handelns in der Gewöhnlichkeit verschwindet [6]. Diese bedachte Dienlichkeit und Sicherheit von Softwaresystemen wird dadurch bedenklich, daß sie die Möglichkeiten des Verhandeln beim Benutzer einschränkt. In der Erstarrung der Dienlichkeit ist das Loslassen nicht mehr möglich. In der Unauffälligkeit der Sicherheit bietet sich kein Anlaß mehr, Systeme zu verlassen. Es ist zur vorherrschenden Meinung geworden, daß gerade als Folge der Einschränkung der Möglichkeiten des Verhandeln Verlässlichkeit entstehe. Damit wird aber die Verlässlichkeit selber reduziert auf "ungebildetes" Vertrauen.

Verlässlichkeit ist jedoch keine Eigenschaft des Zeugs selbst, sondern ein Prozeß, der im Handeln des Benutzers situiert ist. Die Interpretation des Benutzers von bereitgelegtem Handeln entsteht erst und nur im Handeln mit dem Softwaresystem. In einer Interaktion zwischen einem Akteur und einer (vor-)bereit-liegenden Informatikrepräsentation findet bei dem menschlichen Akteur ein Prozeß von benutzendem und entwerfendem Handeln statt. In diesem Prozeß gibt der Akteur immer aufs neue Bedeutung an die Repräsentation. Die Übergänge in diesem Prozeß sind die Momente des Zweifels des Akteurs. Für einen Akteur ist die Bedeutung der Informatikrepräsentation nicht stabil, solange der Prozeß von Benutzen und Entwerfen andauern kann.

Entwerfen und Benutzen von Informatikrepräsentationen sind Handlungen, die einander nicht ausschließen, sondern einander abwechseln und ineinander enthalten sein sollten. So kann zwischen Benutzer und Entwerfer ein kritischer transformativer Raum entstehen, worin die Verlässlichkeit entstehen und als Prozeß andauern kann. Verlässlichkeit kann nicht entstehen, wenn nur die bedachte Sicherheit und Dienlichkeit des Herstellers angeboten wird. Sie gehört nicht zu der Erfahrung des Benutzers. Die Feststellung: *"Working with a rapidly developing technology, one always runs the risk of protecting the past rather than contributing to the future. (...) In the land of computers you will not find security by holding on to the past, but by throwing yourself over the edge of the future."* [9] impliziert aber, daß die Hersteller selber in der Gegenwart über das bereitgelegte Handeln ihrer Methoden der

Softwareentwicklung verhandeln sollten, um dessen Verlässlichkeit für sich erlebbar zu machen.

3. Diskussion der Methode OO

Objektorientierung ist das Beispiel einer Methode, bei der deutlich wird, daß der (Ver)Handlungsraum zwischen Präsentation und Interpretation für die Benutzer immer mehr eingeschränkt wird [8]. Die objektorientierte Methode ist eine Sprache für die Präsentation des Verhaltens von OBJEKTEn, welche durch ihre Realisierung in Software-OBJEKTEn zur Interaktion bereitgelegt werden. Durch die Festlegung einer endlichen Zahl von Zuständen und Zustandsübergängen, sowie der Bedingungen, wann bestimmte Zustandsübergänge stattfinden dürfen, ist es möglich, das Verhalten eines OBJEKTEs vollständig zu determinieren. Der Raum für die Interpretation des präsentierten Handelns durch das realisierte Software-OBJEKT ist leer. Die Bedeutung der Interpretation ist gleich der Bedeutung der Präsentation. Die Beschreibung möglichen Verhaltens wird in der Sprache Objektorientierung durch den Mechanismus der Vererbung vereinfacht; durch Generalisierung können Oberklassen und durch Spezialisierung Unterklassen gebildet werden (mit der is-a-Relation). Die Interaktion zwischen Software-OBJEKTEn ergibt sich aus ihrem festgelegten Verhalten in den OBJEKTEn, Interaktionsprozesse sind somit determiniert. OBJEKTE sind für menschliche Akteure handelbar, wenn sie ein Interface besitzen, mit dem sie direkt wahrnehmen können, wenn der menschliche Akteur die Bedingungen für einen Zustandsübergang erzeugt. Objektorientierte Software enthält in der Regel eine eher kleine Anzahl solcher Interface-OBJEKTE. Die meisten OBJEKTE sind nur indirekt erreichbar.

Einmal konstruierte OBJEKTE können ihr determiniertes Verhalten nicht mehr ändern. In diesem Sinn kann man OBJEKTE "sicher" nennen. Fraglich ist es, ob die objektorientierte Methode und ihre Produkte einen Prozeß von Verlässlichkeit möglich machen und ob die bereitgelegte Sicherheit ein Verhandeln erlaubt.

Die objektorientierte Methode wurde ursprünglich zur Implementation von Systemen dynamisch interagierender (Software-)OBJEKTE entwickelt (SIMULA). Sie hat sich aber in den letzten Jahren auf andere Bereiche ausgebreitet. Durch Produkte wie UML fand eine Kolonialisierung der Analysephase statt, verbunden mit einer "Wiederholung der Vergangenheit" durch die Fokussierung auf Möglichkeiten von "re-use": "(...) *object-oriented techniques allow us to reuse far more than code. We can reuse requirements, analysis,*

design, test plans, user interfaces and architecture. In fact, virtually every component of the software engineering life cycle can be encapsulated as a reusable object." [10]

Beim kritischen Anwenden von Objektorientierung für die Analyse kann man aber feststellen, daß viel des menschlichen Verhaltens nicht repräsentierbar ist, weil jeder Versuch der Determinierung der "Änderung des Änderns" einen Widerspruch erzeugt. Auch die Repräsentation von menschlichem Gruppenverhalten scheitert, weil die Interaktion zwischen den OBJEKTEn auf Konfliktlosigkeit basiert ist. Somit kann der kritische transformative Raum, der bei einer zwischenmenschlichen Interaktion existiert, nicht mittels OO repräsentiert werden. Im Erleben der Verlässlichkeit von OO wird man feststellen, daß es eine Illusion ist:

- daß alle dynamischen Aspekte menschlicher Handlungs-Domänen in einer Struktur von OBJEKTEn abgebildet werden könnten,
- daß alle Subjekte jedem Objekt eine gleiche Bedeutung geben, daß also die Repräsentation dieses Objekts, das OBJEKT, für alle Subjekte eindeutig sei [11][12].

Eine erlebte Verlässlichkeit zeigt, daß die Objektorientierung für die Analysephase der Softwareentwicklung nur sehr bedingt dienlich ist [13][14]. Handlungs-Domänen, deren hierarchische statische Strukturen man sichtbar machen will, können mit der Methode OO aus dieser Perspektive teils präsentiert werden, und es wird so möglich, die hierarchische Struktur zu verlassen oder zu ändern.

Mit bereitgelegten objektorientierten Produkten können immerhin mit Hilfe von Aggregation Verhandlungsprozesse stattfinden; menschliche Akteure können durch Aggregation vorhandener OBJEKTE neue OBJEKTE konstruieren (mit der part-of-Relation). Dies bedeutet, daß menschlichen Akteuren auf zwei Weisen die Interpretation von OBJEKTEn möglich ist:

- die Integration der OBJEKTE ins eigene Verhalten, so wie sie bereitgelegt sind,
- und das Zusammenstellen neuer OBJEKTE aus schon bereitgelegten, wodurch sie neues Handeln für sich selbst bereitlegen können. Deren Verhalten ist zwar teilweise determiniert im Rahmen des präsentierten Handelns der einzelnen Teil-OBJEKTE und ihrer möglichen Zustände. Aber Aggregation läßt zu, daß man die Interaktion zwischen OBJEKTEn neu gestalten und die eigene Interaktion mit den OBJEKTEn immer wieder neu konstruieren kann.

Brenda Laurel sieht Software als ein Theater-SKRIPT, worin Rollen gespielt werden. Objektorientierte Software kann als ein Spiel von OBJEKTEn gesehen werden, die Rollen präsentieren. Die

Aggregation solcher Rollen ist dann die semantische Abgrenzung, die ein menschlicher Akteur selbst machen kann. In einem so bereitgelegten Theaterspiel hat ein menschlicher Akteur vier mögliche Positionen: die eines Akteurs, eines Zuschauers, eines Regisseurs und eines Autors [Bild 3]. Brenda Laurel hat versucht, die Dienlichkeit von Software zu erhöhen, indem sie Benutzer als Akteure ins Spiel positioniert, statt als Zuschauer. Dadurch wird ihrer Meinung nach die gegenseitige Handelbarkeit besser [15]. Aber in der Akteursposition hat ein menschlicher Akteur keinen Verhandlungsraum, weil er immer noch wie ein OBJEKT determiniert handeln muß. Erst aus der Position des Regisseurs heraus kann ein kritischer transformativer Raum entstehen. Diese Position ist für Benutzer wohl möglich, weil in der Objektorientierung das Aggregieren von Handlungen möglich ist und so in der Interaktion selbst eine Differenziertheit von Handlungen entstehen kann. Notwendig ist aber, daß die bereitgelegten objektorientierten Rollen nicht zu viel geschlossenes determiniertes Handeln besitzen, weil man dann der Benutzer die beim Zusammenstellen entstehenden (Rollen-)Konflikte nicht mehr überschauen kann. Wenn aber ein Softwarehersteller in seiner Rolle als Autor eine Differenziertheit von Handeln anbietet (viele möglichst einfache Rollen), kann der Benutzer selbst bestimmen, welcher Verhandlungsraum während des Ausführung des Spieles dienlich ist.

4. Diskussion der Methode CC

Jeder einzelne Akteur (Privatperson, Verein, Firma, Institution, etc.) in einer vernetzten IT-Umgebung muß sich zu allen Handlungen, Objekten und Mitspielern seiner Interaktionsräume differenziert über seinen subjektiven Sicherheitsbedarf klarwerden und diesen in begrenzte aber ausreichende Sicherheitsmaßnahmen umsetzen – begrenzt vor allem schon wegen der Kosten von Implementation und Betrieb solcher Maßnahmen.

Spezifikation von Sicherheitsbedarf und Sicherheitsmaßnahmen fand nie in einer Aura von Unverbindlichkeit statt, welche z.B. die alle Verantwortung abweisenden Lizenztexte aktueller Betriebssysteme, Officesysteme und Internetbrowser ausstrahlen. Sicherheit war schon immer ein Politikum. Haftungsrecht und Staatsräson haben hier von Anfang an für empirisch abgesicherte formale Präzision zwischen den Programmierern und den Betreibern von "verhandelten Sicherheitseigenschaften genügenden" Softwaresystemen gesorgt. Das bisher letzte Glied in der Generationenfolge internationaler Sicherheitsstandards sind die Common Criteria 2.1 (ISO/IEC 15408:1999 "Common

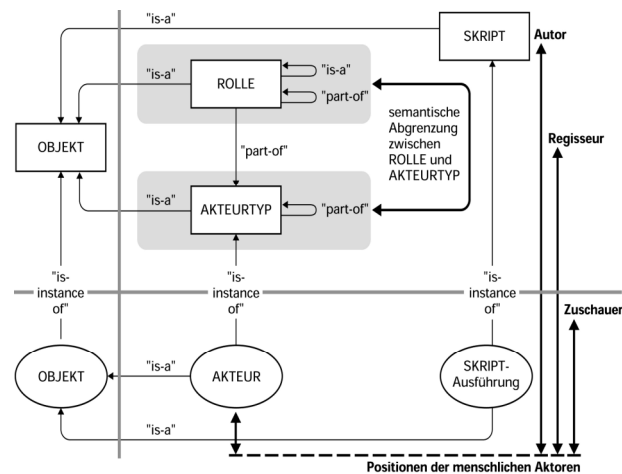


Bild 3: SKRIPT und SKRIPT-Ausführung

Criteria for Information Technology Security Evaluation"), kurz CC [16]. Sie zeichnen sich gegenüber den Vorgängerstandards dadurch aus, daß sie:

- von einer globalen Kooperation auf staatlicher Ebene getragen sind,
- in weiten Teilen generisch (offen) sind, also das weltweite Hinzufügen erweiterter, spezialisierter oder aktualisierter Bestandteile herausfordern,
- eine auf der Qualifikation von Handlungen basierte Ontologie zugrundelegen,
- technologie- und anwendungsunabhängig sind, also nicht nur staatlichen Organisationen und großen Providern dienlich sein könnten, sondern auch z.B. Verbraucherguppen, elektronischen Marktplätzen, Firmen und Privatpersonen.

Dadurch haben die CC, insbesondere mit ihrem Baukastensystem zur Spezifikation von "protection profiles" (PPs, Sicherheitsprofilen), einen transformativen kritischen Raum geöffnet, in dem Benutzer und Hersteller über ihre gegenläufigen Sicherheitsbedürfnisse bei Softwaresystemen verhandeln können. Gegenläufige Sicherheitsbedürfnisse zeigen auf, daß der Raum der CC keineswegs konfliktlos ist. Ein typischer Fall verschiedener Bedeutungsgebung an angebotenes Handeln ist z.B. der mögliche Verlust von Privatsphäre (z.B. aktorischer Anonymität) beim Einsatz biometrischer Verfahren zur Identifikation von Menschen [17].

Der kritische Raum, den die CC öffnen, ist dringend notwendig geworden, weil immer mehr menschliche Lebensbereiche durch Softwaresysteme unterstützt werden (Geldverkehr, staatliche Administration, Handel, Informations- und Unterhaltungsdienste, Medizin, zwischenmenschliche Kommunikation und Zusammenarbeit, Verkehrswesen, etc.) und diese Softwaresysteme

gemeinsam und gleichzeitig auf der gleichen Infrastruktur (global vernetzte Server, PCs, Mobiltelefone, SmartCards, etc.) agieren und interagieren.

Der kritische Raum der CC ist noch offen – aber der gesellschaftliche Prozeß der Ausgestaltung der Verlässlichkeit unseres gemeinsamen elektronischen Hauses ist in vollem Gang. Leider handelt es sich bei den CC um keinen leicht eingängigen Text, man muß sie erst eine Weile studieren, um mit ihnen etwas entwerfen zu können. Das gibt denen, die dafür Experten haben, einen (Ver)Handlungsvorsprung. Staatliche Dienste, IT-Hersteller und Service-Provider bringen derzeit in ihrem Interesse entworfene Sicherheitsprofile auf den Weg der internationalen Standardisierung und allgemeinen gesellschaftlichen Verbindlichkeit. Sind diese erst einmal legislativ implementiert, können wir sie kaum mehr "loslassen". Daher ist es zu wünschen, daß sich viel mehr verantwortungsbewußte Informatiker mit den CC befassen und hier eine gesellschaftliche Vermittlerrolle einnehmen. Dabei können sie die Verlässlichkeit der CC erleben, wenn sie mit den Benutzern entstehender IT-Infrastrukturen über ihre Wünsche und Zweifel verhandeln und ihnen so die Sicherheit dienlich machen. Nur wenn verschiedene gesellschaftliche Gruppen in die Lage kommen, in den CC mit- und gegenzuentwerfen, könnte das zu einer IT-Infrastruktur führen, die nicht als ein globaler Hochsicherheitstrakt angelegt ist, sondern die eine Vielzahl unterschiedlich "sicherer" Interaktionsräume anbietet, in denen eine Differenziert von Handeln bereitliegt. Eine Infrastruktur, in der Zweifeln eine Gewohnheit ist, in der auch auf Unsicherheit bedacht eingegangen werden kann, wo Verlässlichkeit erlebbar bleibt.

5. Schlußfrage

Man kann nach den Gründen für das unbeabsichtigte oder beabsichtigte Verbergen der Verlässlichkeit von Softwaresystemen und -methoden fragen. Betrachten Informatiker eigentlich ihre eigenen Methoden, Theorien und Auffassungen noch als Aktoren, mit denen sie einen entwerfenden Prozeß von Handeln angehen können? Sie finden es bei der Softwareentwicklung dienlich, den transformativen Raum zwischen Analysephase und Implementationsphase mit objektorientierten Methoden zu kolonialisieren, um ihn konfliktlos zu halten. Diese Konfliktlosigkeit ist aber eine Scheinsicherheit, weil sie damit den Benutzern den Prozeß der Verlässlichkeit nicht gestatten. Dementgegen erlaubt es Objektorientierung, eine große Differenziertheit von Handeln bereitzulegen, in der Benutzer mittels Aggregation verhandeln können. Informatiker benutzen

diese Möglichkeit, sich auf den Benutzer zu verlassen, selten. Stattdessen verläßt man sich auf die Dienlichkeit der Analyse und unterläßt hier das Loslassen der Methode. Bei den Common Criteria zeigt sich, daß ein angebotener konfliktvoller Raum, Benutzern, in Interaktion mit ihnen, eine dienliche Sicherheit zu vermitteln, nur zögerlich betreten wird.

Vielleicht haben Informatiker zu wenig Vertrauen in das Änderungspotential von Benutzern, weil sie selber nicht mehr entwerfen?

- [1] Sloterdijk, Peter, *Regeln für den Menschenpark*. Suhrkamp, Frankfurt am Main, 1999, pp.35,37
- [2] Figal, Günther (2000), Martin Heidegger, Phänomenologie der Freiheit, Beltz Athenäum, Weinheim, 2000, pp.81,144
- [3] Hall, Stuart, et al. (eds) *Culture, Media, Language*. Hutchinson, London, 1980
- [4] Dewey, John, *Democracy and Education*. The Macmillan Company, 1916, Chapter 4: Education as Growth
- [5] Scheman, Naomi (1993), *Engenderings. Constructions of Knowledge, Authority, and Privilege*. Routledge, New York, 1993, pp.208
- [6] Heidegger, Martin, *Der Ursprung des Kunstwerkes (1936)*. Philipp Reclam jun., Stuttgart, 1960, pp.28
- [7] Heidegger, Martin, *Sein und Zeit (1926)*, Max Niemeyer Verlag, Tübingen, 17.Auflage, 1993, §31 pp.145-146
- [8] Crutzen, Cecile K. M., *Interactie, een wereld van verschillen (Een visie op informatica vanuit genderstudies)*. Dissertation, Open universiteit, Heerlen, 2000
- [9] Dahlbohm, Bo, *The New Informatics, Scandinavian Journal of Information Systems*, Vol.8, No.2, 1997
- [10] Yourdon, Edward/Argila, Carl, *Case Studies in Object-Oriented Analysis and Design*. Yourdon Press, Upper Saddle River, 1996, pp.6
- [11] Coyne, Richard, *Designing Information Technology in the Postmodern Age (From Method to Metaphor)*. The MIT Press, Cambridge (MA), 1995, pp.183
- [12] Hirschheim, Rudy/Klein, H. K./Lyytinen, K., *Information Systems Development and Data Modeling (Conceptual and Philosophical Foundations)*. Cambridge University Press, Cambridge (UK), 1995, u.a. pp.62-63,85,109
- [13] Crutzen, Cecile K. M./Vosseberg, Karin, *Die Interaktion zwischen objektorientiertem Denken und feministischer Kritik – eine dynamische Verbindung. Dreher, B./et al. (eds.) (1999), Software Engineering im Unterricht der Hochschulen SEUH '99*, Teubner Verlag, Stuttgart, 1999, pp.149-165
- [14] Crutzen, Cecile K. M. /Gerrissen, Jack F., *Doubting the OBJECT World. Balka, Ellen/ Smith, Richard (eds), Women, Work and Computerization, Charting a Course to the Future*. Kluwer Academic, Boston, 2000, pp.127-136
- [15] Laurel, Brenda, *Computers as Theatre*. Addison-Wesley, Reading, Massachusetts, 1993, pp.17,20-21,44-45
- [16] "www.bsi.de/zertifiz/" , "csrc.ncsl.nist.gov/nistpubs/cc/"
- [17] Hein, Hans-Werner, *Big Brother is scanning you (Biometrie und Datenschutz), Spektrum der Wissenschaft*, März 1999, Heidelberg, pp.106-109