Prozessmanagement im Katastropheneinsatz: agil, strikt, tolerant und robust?

Gerhard Chroust[1], Georg Aumayr[2], Gudrun Haider[3], Rudolf Randus[4] und Alexander Thür[5]

Abstract: Ausgehend von der Mechanisierung der Fertigungsindustrie ab dem 19. Jahrhundert verbreitete sich das Konzept eines definierten und aufgezeichneten Modells für einen Prozess (ein ‚Vorgehensmodell') und dessen (mehr oder minder) automatische Abarbeitung durch einen Prozess-Interpretierer auf immer weitere Anwendungsgebiete. Der Interpretierer kann ein Mensch, ein mechanisches Gerät (z.B. Jacquard-Webstuhl) oder ein Software-Produkt ('Process Engine') sein. Jede weitere Bereichsausweitung bedingte eine weitere Aufweichung des strikten 'automatischen' Ablaufes. Dabei zeigt es sich, dass neben der Erweiterung des Vorgehensmodells noch stärker die Erweiterung der Funktionalität des Prozess-Interpretierers notwendig ist. In diesem Beitrag identifizieren wir historische Anforderungen an das Paradigma 'Vorgehensmodell/Prozess-Interpretierer'. Wir betrachten Katastrophenmanagement als eine neue Herausforderung und wagen einen Blick in die Zukunft. Dabei sind zwei Herausforderungen zu unterscheiden: die Definition des Vorgehensmodells und die Implementierung des Prozess-Interpretierers. Letzteres ist von besonderem, leider aber zu wenig geschätztem Interesse, wird aber in diesem Beitrag in den Vordergrund gestellt.

Keywords: Vorgehensmodell, Navigation, Interpretierer, Instanziierung, Process Engine, Software-Entwicklung, Feldversuch, INSARAG, Can Do, Johanniter, makeit GmbH, Katastrophenmanagement

1 Motivation

Ein wesentlicher Beitrag zu den Erfolgen der Fertigungsindustrie ab dem 19. Jahrhundert war das Konzept der 'automatischen Fabrik', basierend auf dem Konzept des definierten und aufgezeichneten Modells eines Prozesses (eines ‚Vorgehensmodells') und der (mehr oder minder) automatischen Abarbeitung durch einen Abarbeitungsmechanismus, "Prozess-Interpretierer", "Process Mechanism" [Ch89], "Process Engine" etc. genannt. Ein Prozess-Interpretieren leitet aus den Vorgaben des Vorgehensmodells eine Sequenz von Aktivitäten ab und koordiniert und steuert die Durchführung der Aktivitäten durch Menschen, Automaten oder Software (Abb. 1). Die ursprünglichen Vorgehensmodelle im Software-Engineering (z.B. [Ro70]) erfuhren eine große Erweiterung durch Detaillierung der beschriebenen Komponenten, Erhöhung der Granularität und Ergänzung mit parallelen Submodelle, die neben der eigentlichen Entwicklung auch begleitende Tätigkeiten wie

[1] Johannes Kepler Univ. Linz, Institut für Telekooperation, Altenberger Straße 69, A-4040 Linz, Austria, gerhard.chroust@jku.at

[2] Johanniter Österreich Ausbildung und Forschung gemeinnützige GmbH, Wien, Ignaz-Köck Straße 22, 1210 Wien, georg.aumayr@johanniter.at

[3] Johanniter Österreich Ausbildung und Forschung gemeinnützige GmbH, Wien, Ignaz-Köck Straße 22, 1210 Wien, gudrun.haider@johanniter.at

[4] makeit information systems GmbH, Wien, Mooslackengasse 17, 1190 Wien, rudolf.randus@makeit.at

[5] makeit information systems GmbH, Wien, Mooslackengasse 17, 1190 Wien, alexander.thuer@makeit.at

Qualitätsmanagement, Projektmanagement [CGMS88, BR05, HH08] etc. modellierten. Der Erfolg von Assessment-Methoden (ISO/IEC 15504 SPICE, CMMI [Kn06] und ISO/IEC 32000 [IS12]) veranlasste die Anwendung des Paradigmas von Vorgehensmodell/Interpretierer auf weitere Bereiche, wobei die Auto-Industrie einen der Treiber darstellt [Sc12].

Für die verschiedenen Anwendungsfelder genügt prinzipiell dasselbe Metamodell, das im wesentlichen aus den in Abb. 2 dargestellten Komponenten besteht [Ch00]. Obwohl des Vorgehensmodell einen linearen Ablauf suggeriert, ist eine strikt sequentielle Abarbeitung eines Vorgehensmodells in Anlehnung an das Fließband-Paradigma der Fertigungsindustrie [Ro70] und [BBL76] nicht adäquat. Eine Aufweichung des strikten automatischen Ablaufs war erforderlich und daraus folgte eine starke Erweiterungen der Funktionalität des Prozess-Interpretierers.

Globale und internationale Koordination und Kooperation bei Interventionen im Katastropheneinsatz, wie sie auch von der Europäischen Union gefordert wird, ist nur auf Basis von international akzeptieren und kodifizierten Vorgehensmodellen möglich. Im Bereich des Katastrophenmanagement gibt es ebenfalls eine Reihe von Vorgehensmodellen z.B. [IN12, IF07b], doch sind sie in Prosa-Text geschrieben, bestenfalls mit gewissen standardisierten Formatierungen. Sie sind aber weit von einer formalen Darstellung entfernt. Damit ist aber auch die Unterstützung durch einen Prozess-Interpretierer [IS11, IS12] nicht möglich. Aber selbst wenn das Vorgehensmodell in eine computer-lesbare und damit interpretierbare Form umgeschrieben wird, ergeben sich noch immer wesentliche Unterschiede bei der Interpretation (dem 'enactment') dieser Modelle. Wesentliche diesbezügliche Erkenntnisse brachte eine Studie über die Computer-Unterstützung im Katastrophen-Management [Ha14], die auch wesentliche Unterschiede herausarbeitete (vgl. Kapitel 5).

Der Beitrag ist wie folgt strukturiert: Kapitel 2 diskutiert das Zusammenspiels von Vorgehensmodell und Prozess-Interpretierer, Kapitel 3 die Unterschiede zwischen Software-Engineering und Katastrophenmanagement. Im Kapitel 4 werden die konkreten Anforderungen an einen Prozess-Interpretierer (‚Navigation'), herausgearbeitet. Ein Feldversuch und daraus gezogene Lehren werden Kapitel 5 zusammengefasst.
Wichtige Aussagen zum Katastrophenmanagement sind mit " KM ⇒ ...Text..." markiert.

2 Das Vorgehensmodell/Prozess-Interpretierer-Paradigma

Das Basiskonzept ist in Abb. 1 skizziert: Ein Vorgehensmodell wird durch einen entsprechenden Interpretierer Schritt für Schritt abgearbeitet, wobei der Benützer die Steuerung und Reihenfolge ("Navigation") in Abhängigkeit von äußeren Gegebenheiten und Notwendigkeiten durchführt.

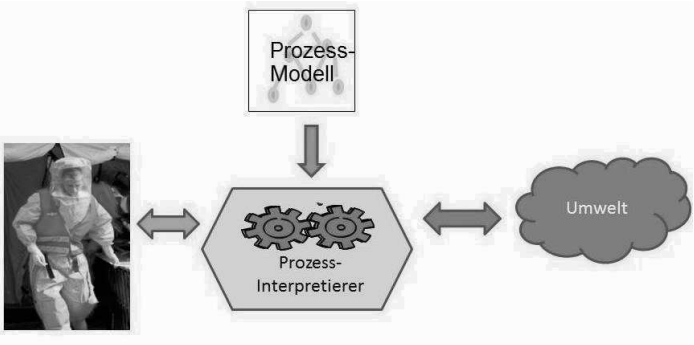

Abb. 1: Vorgehensmodell und Prozess-Interpretierer

2.1 Software-Engineering

Das Vorgehensmodell/Prozess-Interpretierer-Paradigma wurde im Wesentlichen von der Fertigungsindustrie geschaffen, wobei die 'Interpretierer' anfänglich Menschen waren ('arbeitsteilige Fertigung'), die erst allmählich durch entsprechende mechanische Interpretierer unterstützt wurden. In der Software-Industrie hat sich das Paradigma der Vorgehensmodelle sehr schnell durchgesetzt, angeregt durch die Ähnlichkeit zwischen dem Software-Entwicklungsprozess und den durch die Software gesteuerten Prozessen: Man hörte *"Software Processes are Software too"* [Os87]. Die Idee der Steuerung der Software-Entwicklung durch Interpretation von Software-Entwicklungsmodellen kam etwas später; sehr früh gab es in Deutschland die sogenannte „DV-Verfahrenstechnik" [IB78]. Software-Engineering Environments, die Vorgehensmodelle, Interpretierer und Werkzeuge integrierten, waren bald große Renner [Hu80].

Im Laufe der Zeit hat das software-basierte Vorgehensmodell/Prozess-Interpretierer-Paradigma praktisch alle Bereiche der Industrie erobert: Software Engineering, Büroautomation, Mobile und Soziale Medien, e-Business, e-Health und mit etwas Verspätung auch Katastrophenmanagement im Interventionsfall.

Dabei sind zwei Herausforderungen zu unterscheiden: die Definition des Vorgehensmodells und die Implementierung des Prozess-Interpretierers. Letzteres ist von besonderem, leider aber zu wenig geschätztem, Interesse wird aber in diesem Beitrag in den Vordergrund gestellt. War ursprünglich nur der eigentliche Entwicklungspfad Thema eines Vorgehensmodell, so wurden bald Bereiche, die mit der eigentlichen Entwicklung korreliert waren, ebenfalls dargestellt (vgl. PM: Projektmanagement, QS: Qualitätssicherung, KM: Konfigurationsmanagement, PA: Problem- und Änderungsmanagement im V-Model [BR05, HH08]). Ein reichhaltiges Metamodell zeigt Abb. 2 [CKS10]. Von den einzelnen Typen des Modells werden beim Betrieb mehr oder minder viele Instanzen erzeugt. Seine einzelnen Elemente (zusammen mit Beispielen aus dem Katastrophenmanagement) sind:

Resultattyp: Produkte, die "WAS zu erzeugen ist" beschreiben *(z.B. "Einsatzplan")*

Resultatabhängigkeiten: logische Abhängigkeiten zwischen (Teil-)Produkte voneinander *(z.B. "Hubschrauber-Anforderung benötigt ärztliche Diagnose")*

Resultatstruktur: hierarchische Struktur der Produkte zwecks Strukturierung und Aggregierung *(z.B. "Schlauch ist Teil des Löschfahrzeuges")*

Aktivitätstyp: auszuführende Aktivitäten *(z.B. "Verschüttete orten", "Gebäude sichern")*

Eingangsbeziehung: Identifizierung der Eingaben zu einer Aktivität *(z.B. "Lageplan" ist eine Eingabe zu "Einsatzplan erstellen")*

Ausgangsbeziehung: Identifizierung der Ausgaben (Resultate) einer Aktivität *(z.B. "gepölztes Haus" ist Resultat von "Gebäude sichern")*

Aktivitätsstruktur: Über- und Unterordnung von Aktivitäten *(z.B. "Lageplan erstellen" ist eine Aktivität in "Phase Orientierung")*

Aktivitätsfluss: Abstraktion und Idealisierung der Reihenfolge der Ausführung von Aktivitäten. Die wesentlichen Prioritäten werden beschrieben *(z.B. "zuerst Verschüttete finden" dann erst "Verletzte versorgen")*.

Akteur-Rolle: Bei manchen Modellen werden auch Rollen der Akteure festgelegt *(z.B."Sanitäter") [IN12]*

Werkzeug: Verwendung von Methoden und/oder Werkzeugen zwecks Uniformität beim Einsatz *(z.B. "Suchhund", "schwerer Bagger") [IN12]*

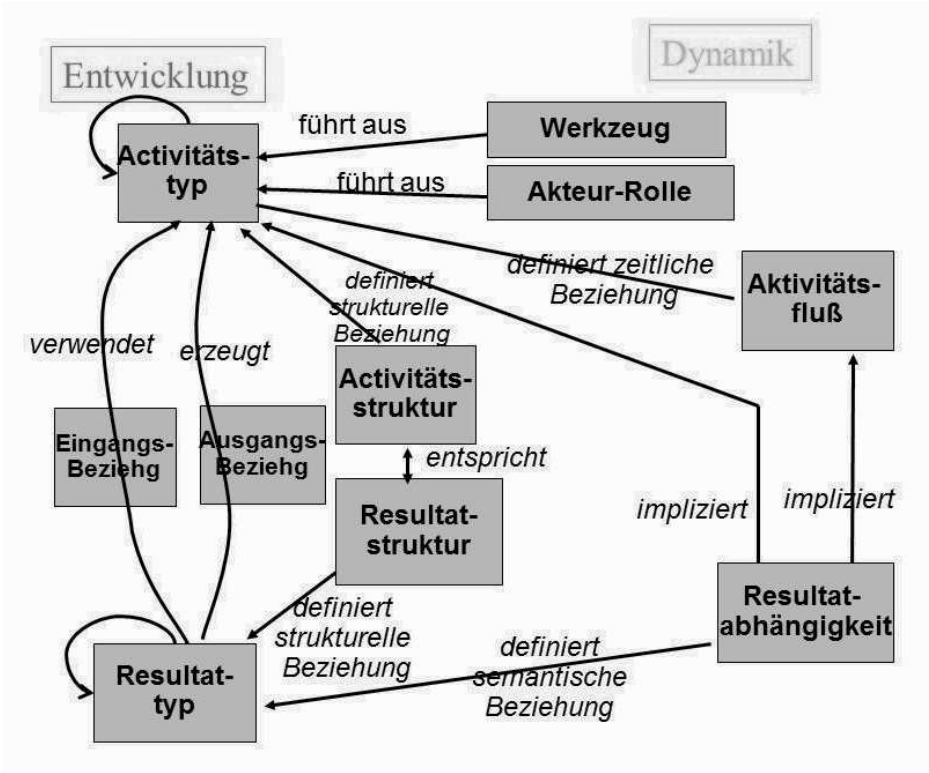

Abb. 2: Allgemeines Meta-Vorgehensmodell

Im Vergleich zu anderen Anwendungsgebieten können wir für das Katastrophenmanagement feststellen:

- Die Modelle in den einzelnen Anwendungsgebiete unterscheiden sich im wesentlichen durch die Relevanz oder auch durch Nichtvorhandensein mancher Komponenten (z.B. Werkzeugen) und durch die andersartigen Namen für die einzelnen Komponenten.
- Die Akteur-Rollen und die notwendigen Werkzeuge sind oft, z.B. in [IN12], sehr detailliert beschrieben. *(z.B. 'Datenbank entwerfen', 'Pass ausstellen', 'Verletzte retten')*.
- Unterschiede gibt es bei Zahl und Untergliederung der Komponenten, ebenso ist die Zahl der Instanzen sehr verschieden, vgl. [Ch95].
- Die Toleranz bei Detaillierung des Aktivitätsflusses ist ebenfalls sehr unterschiedlich,
- *KM ⇒ Büroautomation und e-Business neigen zu einer strikteren Festlegung des Aktivitätsflusses, auch aus rechtlichen Gründen (Audit, Compliance), während Katastrophenmanagement eher zur agilen Abarbeitung der Aktivitäten tendiert.*

2.2 Der Prozess-Interpretierer

Ein Prozess-Interpretierer, vgl. Abb. 1, hat mehrere unterschiedliche Aufgaben:

Instanziierung und Verwaltung der Aktivitäten: Für fast alle Aktivitätstypen werden mehrere, oft sehr viele Instanzen geschaffen. Diese müssen für die Akteure angezeigt und von ihnen ausführbar gemacht werden, zusammen mit einem Zugang zu den benötigten Resultaten (von anderen Aktivitäten) und Ablagemöglichkeiten für zu erzeugende Resultate. Die entsprechenden Verknüpfungen zwischen benötigten und zu erzeugenden Resultaten, deren Status und allfälligen Einschränkungen sind ebenfalls zu berücksichtigen.
KM ⇒ Im Katastrophenfall entsteht eine große Zahl von parallel zu erzeugenden Resultaten und durchzuführenden Aktivitäten. Es müssen viele oft neue Akteure im System registriert und verwaltet werden, ebenso Geräte, die ad-hoc eingebracht werden.

Instanziierung und Verwaltung der Zwischen- und Endresultate: Auf Grund der Größe heutiger Anwendungen entsteht eine Vielzahl von Instanzen der einzelnen Resultattypen. Sie müssen verwaltet, bezüglich ihres Status und Inhalt aktualisiert werden und den einzelnen Aktivitäten zur Verfügung gestellt werden.
KM ⇒ Je weiter sich die Anwendungsgebiete von der eigentlichen Software-Entwicklung entfernen, desto mehr Resultattypen sind reale Objekte der Umwelt (Verletzte, Gebäude, ...) und somit nur durch 'Surrogate' im Vorgehensmodell bzw. Interpretierer abbildbar.

Navigation: Wesentlich ist die sogenannte 'Navigation', d.h. die Entscheidung in welcher Reihenfolge die einzelnen Aktivitäten (Instanzen!) durchzuführen sind. Das Vorgehensmodell (als ein Konstrukt auf der Typen-Ebene) enthält nur wenige Informationen über die Abarbeitungsreihenfolge, da viele der Vereinbarungen nur auf der *Instanzenebene* fest-

gelegt werden können. Die Navigation berücksichtigt neben den logischen Abhängigkeiten wie sie z.T. durch das Vorgehensmodell beschrieben sind, noch weitere Nebenbedingungen und Einschränkungen nichttechnischer Art wie sie aus dem Projektmanagement kommen, z.B. Prioritäten (z.B. Lebensrettung), verfügbares Personal, Materialbedarf, Zeitgrenzen und Termine, strategische Aufgabenteilung, politische Entscheidungen, usw.

Protokollierung, Ressourcen-Erfassung: Die computer-gesteuerte Navigation ermöglicht auch eingesetzte Ressourcen (Einsatzkräfte, auch Helfer, Betriebsmittel, etc.) als auch den Betriebsmitteleinsatz zu erfassen und am Ende abzurechnen.

Ex-post Analyse: Eine genaue Protokollierung der Reihenfolge der einzelnen Aktivitäten ist für Audits, Schadenserfassung und Abrechnung wertvoll. Man kann auch wichtige Sub-Prozesse identifizieren und analysieren („Prozess-Mining" [KK16]). Eine Reifegrad-Messung wäre (z.B. CMMI [Kn06] oder ISO/IEC 32000 [IS12]) möglich.

3 Katastrophenmanagement

3.1 Prozess-Sicht

Die Prozess-Sicht ist aus vielerlei Gründen für das Katastrophenmanagement von großer Bedeutung [Ch12, Kap. 5] [Au15a, CA14]. Globale Koordination und Kooperation bei Interventionen im Katastropheneinsatz, wie sie auch von der Europäischen Union gefordert werden, sind nur auf Basis von international akzeptierten und kodifizierten Prozess-Modellen möglich. Ein typisches Beispiele ist der von der ISO entwickelte Standard ISO22320 [La13, IS11]. Die ISO 22300-Standard Familie definiert globale Best Practices um Command und Control Strukturen und Prozeduren, Entscheidungsunterstützung, Nachvollziehbarkeit (traceability) und Information Management sicherzustellen.
Ähnlich Dokumente sind die Guidelines wie sie von der International Federation of Red Cross and Red Crescent Societies (IFRC) [IF07b, IF07a] und von der International Search and Rescue Advisory Group (INSARAG) von OCHA [IN12] für interoperative Interventionen und Aktionen publiziert wurden. Einsatzkräfte im internationalen Katastrophendienst müssen diese Richtlinien befolgen, besonders um eine reibungslose Kommunikation und Zusammenarbeit mit anderen nationalen Einsatzkräften zu gewährleisten. Nationale Organisationen erstellen ebenfalls Handbücher und Anleitungen für Notfallsituationen. Die Unterstützung durch Prozess-Interpretierer ist noch kaum angedacht. Ein Grund ist, dass die existierenden Vorgehensmodelle wenig für eine Automatisierung geeignet sind, wie der Feldversuch (Abschnitt 5) zeigte.

3.2 Herausforderungen im Katastropheneinsatz

Im Vergleich zur klassischen Software-Entwicklung verschieben sich im Katastropheneinsatz die Schwerpunkte. Wesentliche Herausforderungen sind:

Zeit- und Erfolgsdruck: Katastropheneinsätze stehen praktisch immer unter Zeitdruck: weitere drohende Gefahren müssen abgewendet werden, die Rettung von Menschenleben ist durch oft sehr enge Zeitfenster eingeschränkt.

wechselnde Prioritäten: An einen 'geordneten' Ablauf im Sinne des Vorgehensmodells ist nur bedingt zu denken. *KM ⇒ Manche Aktivitäten müssen kurzfristig zugunsten anderen Aktivitäten (z.B. " Verschüttete bergen", oder "Einsatzkräfte in Sicherheit bringen") unterbrochen werden.*

Hohes Stress und psychische Belastung*: KM ⇒ Sowohl Einsatzkräfte als auch Opfer stehen vielfach unter Stress (auch durch Sorge um ihre eigenen Anverwandten)*

Kooperation mit Unbekannten*: KM ⇒ Internationale Einsätze erfolgen oft in fremden, unbekannten Gebieten mit unbekannten Menschen, sowohl als Opfer als auch als lokale Einsatzkräfte. Auch die Einsatzkräfte selbst sind meist international zusammengewürfelt und kennen sich a-priori nicht, was auch zu kulturellen Missverständnissen führen kann.*

Externe Sichtbarkeit und mediale Aufmerksamkeit*: KM ⇒ Katastrophen erregen im allgemeinen große Aufmerksamkeit der Bevölkerung weltweit (auch von Nicht-Betroffenen). Dadurch ergibt sich auch großes Interesse von Medien und Politik, die sehr oft als Störfaktor auftreten.*

Politischer Widerstand, kulturelle Hemmnisse, etc.*: KM ⇒ Nicht immer wollen Behörden, andere Interessenvertreter oder Stakeholder, dass die Wahrheit unkontrolliert an die Öffentlichkeit dringt. Ebenso können im Zug der Rettungsmaßnahmen kulturelle Tabus verletzt werden.*

Systemische Probleme*: KM ⇒ Katastrophen tendieren dazu, stark vernetzte Ursachen und Auswirkungen zu haben, wobei auch neue Auswirkungen (Emergenz!) auftreten können. Auch unerwartete dominoeffekt-artig auftretende Folge-Katastrophen sind zu berücksichtigen*

Nichtprofessionelle Hilfskräfte: *KM ⇒ Von freiwilligen Helfern, die meist in großer Zahl erwartet werden, kann man keine professionelle Ausbildung, und schon gar nicht ein computergesteuertes Verhalten in Bezug auf ein Vorgehensmodell erwarten. Dies ist besonders bei der Auswahl von Teams und bei der Aufgabenverteilung zu berücksichtigen.*

Aufwand-Erfassung*: KM ⇒ Es ist notwendig zeitlichen und materiellen Aufwand genau zu dokumentieren, um nach der Intervention entsprechende Vergütungen durchzuführen.*

Widriges Umfeld*: KM ⇒ mangelnde Infrastruktur (z.B. Strom- oder Internet-Anbindung), mangelhafte und schlechte Straßen, etc.*

Beschädigte Support-Strukturen*: Die für Katastropheneinsätze vorgesehenen Ressourcen (Personen und Material) können durch die Katastrophe selbst zu Schaden gekommen und nur teilweise einsatzfähig sein.*

Betrachtet man die oben angeführten Herausforderungen so kann man mehrere (sich teilweise widersprechende) Eigenschaften für die Vorgehensmodell/Interpretierer-Anwendung identifizieren. Es ergeben sich vier wesentliche Forderungen:

strikt: Aktivitäten und Resultate müssen präzise definiert und allgemein-verständlich sein, um eine genaue Kontrolle über abgeschlossene und nicht abgeschlossene Aktivitäten zu ermöglichen. Das impliziert eine relativ strikte Abbildung des Geschehens auf das Vorgehensmodell und allfälliger Navigationsvorschriften sowie eine strikte Protokollierung. Auch soll ein Audit über Zeit- und Ressourcenverbrauch möglich sein.

agil: Die Reihenfolge der Ausführung von Aktivitäten muss flexibel sein, sowohl bezüglich Reihenfolge als auch Unterbrechungsmöglichkeiten. Unterstützung bei der Wiederaufnahme von unterbrochenen Aktivitäten ist nötig. Die Navigation muss mehr 'judgement-based' und nicht 'model-based' sein [BH11]. Möglichst geringe (und wenig aufwändige) Dokumentation ist sehr wichtig, um die eigentlichen Rettungskräfte zu entlasten (Agilität! [Be02]).

tolerant: Sowohl der Einsatz von nicht-professionellen Helfern als auch die mangelnde Verfügbarkeit von Ressourcen müssen vom System toleriert werden, bzw. man muss Umgehungsmöglichkeiten erlauben.

robust: Das System muss im Einsatz robust sein, sowohl gegenüber physischen Störungen als auch Problemen der Infrastruktur (intermittierende Stromversorgung und Internet, nicht geschulte Operatoren, kulturelle Differenzen, etc.).

4 Prozess-Interpretierer im Katastropheneinsatz

In der Software-Entwicklung können für die Abarbeitung des Vorgehensmodells verschiedene Strategien verwendet werden, die auch wesentliche Entwicklungsmethoden abbilden (z.B. Wasserfall, Spiralmodell, Evolutionäre Entwicklung, etc.), vgl. Abb. 3.

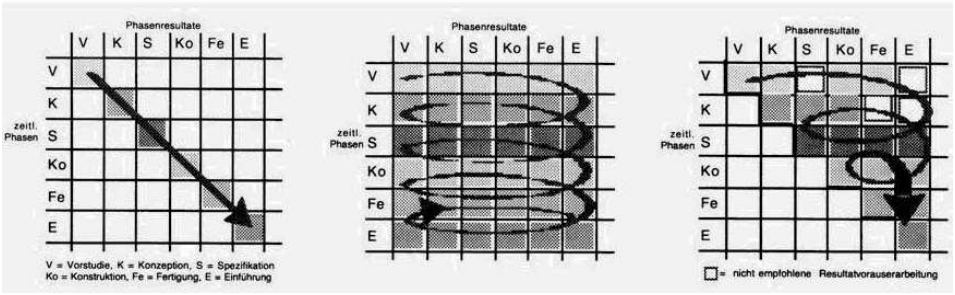

Abb. 3: Verschiedene Abarbeitungsstrategien

Grundlegende Differenzierungen besteht zwischen

der Bevorzugung der phasenweisen Abarbeitung, d.h. alle Aktivitäten einer Phase (='Entwicklungsstufe') werden *vor* den Aktivitäten der nächsten Phase erledigt (Wasserfall-Modell [Ro70]). Ein 'Vorpreschen' soll dadurch vermieden wird.

der Bevorzugung der Fertigstellung von Teilprodukten ('clusterweise' Abarbeitung) bevor ein anderes Teilprodukt in Angriff genommen wird. Dies erlaubt ein früheres Testen und erhöht die Sichtbarkeit, was aber im Katastrophenmanagement wenig Rolle spielt.

Dem verfeinernden Vorgehen im Sinne des Spiralmodells, das die Auswahl unter Risiko-Abschätzung vornimmt [Bo86].

Diese Strategie-Fragen sind analog zu den verschiedenen 'tree-walking'-Strategien im Software-Compilerbau: 'depth-first' versus 'breadth-first'[Pa10, Chapter 5].
KM ⇒ Katastrophenmanagement benötigt ad-hoc Entscheidungen basierend auf der Dringlichkeit, und auch oft 'schnelle Teillösungen', was die clusterweise Abarbeitung oder auch die risiko-orientierte Abarbeitung bevorzugt.

Parallele Wege: Wegen Zeitproblemen und Personalverfügbarkeit werden viele Aktivitäten parallel ausgeführt. Die Koordination (eventuell auch die Zusammenführung der Wege) durch den Prozess-Interpretierer ist von entscheidender Bedeutung.
KM ⇒ Es werden i.a. sehr viele Helfer gleichzeitig an diversen Aktivitäten teilnehmen. Über einen Prozess-Interpretierer ist es möglich, der Einsatzleitung einen besseren (und auch aktuelleren) Überblick zu geben.

Unterbrechung/Preeemption: I.a. kämpfen mehrere Aufgaben um dieselben Ressourcen. Es ist daher manchmal notwendig, eine Aktivität mit oder ohne Vorwarnung [SRC15] zugunsten einer anderen Aktivität zu unterbrechen um Ressourcen zu re-allozieren. Die Art des Abbruchs (Abbruch unter Verlust der gemachten Änderungen, Speichern der bisherigen Änderungen, Bewahrung eines früheren Checkpoint, etc.) ist vom Prozess-Interpretierer zu administrieren.
KM ⇒ Unterbrechungen kommen sehr häufig vor, wobei in vielen Fällen "Speichern des erreichten Zustandes" sinnvoll ist, aber auch dokumentiert werden muss, da vielleicht andere Akteure die Aufgabe später wieder aufnehmen.

Auswahl der nächsten Aktivität: Im allgemeinen ist zu erwarten, dass mehrere Aktivitäten (Instanzen!) begonnen werden können. Welche ist als nächste zu nehmen? Im Prinzip fällt diese Auswahl dem Prozessverantwortlichen (Einsatzleiter, Entwicklungsingenieur, Bürochef, ...) zu. Der Interpretierer kann jedoch wertvolle Entscheidungsgrundlagen anbieten, basierend auf technischen, administrativen, projektinternen und projektextern Daten (vgl. [SRC15]). Dazu gehören auch weitere Informationsquellen wie Simulationen, Projektionen, Data Mining, die bereits getroffenen Entscheidungen berücksichtigen [CH12].
KM ⇒ Im Katastrophenmanagement ist die Entscheidung oft viel kritischer als in der Software-Entwicklung, da diese Entscheidungen oft menschliche Aspekte berücksichtigen müssen, unter Zeitdruck stehen und große Auswirkung auf Leben und Gesundheit der Betroffenen haben.

Im Prinzip stehen folgende Alternativen zur Verfügung:

Instanziierung einer neuen Aktivität: Durch Instanziierung eines Aktivitätstypen kann eine neue Aktivität begonnen werden
KM ⇒ Dies bedeutet sehr oft, viel häufiger als im Software-Engineering, die Unterbrechung einer anderen Aktivität, um Ressourcen zu gewinnen.

Wiederaufnahme einer unterbrochenen Aktivität/Verfeinerung: Eine bis zu einem gewissen Grad abgeschlossene Aktivität wird zum Verfeinern wieder aufgenommen. Im Spiralmodell [Bo86, BT15] wird die Wiederaufnahme zum Prinzip erhoben.

KM ⇒ Die Dringlichkeit vieler Aktivitäten macht es sehr oft notwendig, z.B. kann es sich um die Weiterbehandlung eines erstversorgten Opfers handeln. Die Grenze zwischen Wiederaufnahme einer unterbrochenen Aktivität und der Verfeinerung ist schwer zu ziehen.

Work-ahead: Falls eine Aktivität B auf die Resultate einer verspäteten Aktivität A warten muss, kann B unter gewissen Bedingungen und Vorsichtsmaßnahmen gestartet werden. Bei Fertigstellung von A muss die Konformität und Korrektheit von B sichergestellt bzw. es müssen Korrekturen eingeleitet werden, vgl. [Ph89].
KM ⇒ Es wird sehr oft vorkommen, dass 'auf Verdacht' als Vorsorge gewisse Aktivitäten durchgeführt werden ohne alle Voraussetzungen zu kennen.

Wiederaufrollung (Änderung, Fehler): Bereits das Wasserfall-Modell [Bo76] enthält, basierend auf einer Verifizierungs/Validierungs-Aktivität, einen Rücksprung in die vorhergehende Phase. Die Gründe können Fehler, externe Änderungen (Fakten oder 'politische' Entscheidungen) sein. Aus der Sicht des Prozess-Interpretierer ist kaum ein Unterschied zu einer *Wiederaufnahme* von bereits abgeschlossenen Aktivitäten zu finden. Derartige Rücksprünge werden im Vorgehensmodelle höchstens angedeutet, finden aber auf der Instanzenebene statt. Außerdem müssen weitere von der Änderung betroffene Resultate identifiziert werden und deren Änderung veranlasst werden („Domino-Effekt").
KM ⇒ Im Katastrophenmanagement müssen Entscheidungen unter großer Unsicherheit und basierend auf unsicheren Informationen gefällt werden, wodurch Korrekturen immer wieder notwendig sind.

Verkettete Wiederaufnahmen (Domino-Effekt): Diese Wiederaufnahmen zwecks Änderung betrifft oft mehr als eine Aktivität. Auch hier bedarf es einer strategischen Entscheidung, in welcher Reihenfolge vorgegangen werden soll (Vorwärtsschreiten von der ersten ausgeführten Aktivität, Rückwärtsschreiten von der letzten Aktivität, oder 'middle out').
KM ⇒ Da die 'Produkte' bereits extern vorhanden und sichtbar sind, erfordert dies in vielen Fällen eigene 'Rückabwicklungs-Aktivitäten'. Dies ist bei Software-Entwicklung nur bei ausgelieferten Produkten notwendig (Kundenbenachrichtigungen, eventuell Rückruf, etc.).

5 Feldversuch

5.1 Projekt-Management

Viele der oben aufgezählten Aufgaben sind de-facto Aufgaben des Projektmanagements. Das bedeutet, dass hier eine Überlappung/Ergänzung zwischen Prozess-Interpretation und Projektmanagement besteht und diese Aufgaben in dem einen oder anderen Software-Produkt durchgeführt werden können. Wesentliche Aufgaben sind: Verwaltung/Verwendung der Ressourcen (Personal, Material, Infrastruktur, Logistik) gemeinsam für mehrere Projekte inklusive der notwendigen Aufzeichnungen, Zeitaufzeichnungen (Terminplanung und -überwachung, Dokumentation), Infrastruktur (Sicherstellung der Verfügbarkeit und Ausfallsicherheit), Back-up, etc.

KM ⇒ Der Ausfall kritischer Ressourcen (z.B. des Prozess-Interpretierers) ist für ein Projekt gefährlich. Besonders im Katastropheneinsatz muss die Ausfallsicherheit ohne Einschränkungen gegeben sein.

5.2 Versuchsannahme und Anordnung

Computerunterstützung im Katastrophenfall steht noch im Anfangsstadium [Ch12, Au15b, Ha14]. Um die Möglichkeiten des Einsatzes eines computerbasierten Projektmanagement-Werkzeuges zu untersuchen wurde in Zusammenarbeit zwischen einer Blaulicht-Organisation (Johanniter Österreich - Ausbildung und Forschung gemeinnützige GmbH) und einem Projektmanagement-Anbieter (Fa. makeit information systems GmbH, Wien, mit der Projekt-Management-Software "Can Do"[Ma15]) unter Zugrundelegung des internationalen Standards "INSARAG Guidelines and Methodology Technical Report, 2012" [IN12] ein Versuch und Feldtest durchgeführt und dokumentiert [Ha14]. Ziel des Feldversuches war es, einen Großschadensfall abzuarbeiten und die notwendigen Planungen durchzuführen. Die Übungsannahme war ein schweres Zugunglück im urbanen Bereich mit einer hohen Anzahl von verletzten Personen in einer ungesicherten Gefahrenzone (Verschubbahnhof), siehe Abb. 4.

Abb. 4: Versorgung der Verletzten

Als vorbereitender Schritt wurde unter Benützung des PDF-nach-XML Übersetzungsprogramm von Adobe Acrobat aus dem INSARAG-Manual (in PDF-Format [IN12]) ein Rohdokument in XML-Format erzeugt. Daraus kann relativ leicht ein interpretierbares Vorgehensmodell, abgeleitet werden, das den Import-Anforderungen der Projektsoftware CanDo entspricht (dieser Schritt wurde nur händisch ausgeführt). Die im INSARAG-Manual meist nur implizit vorkommenden Resultate wurden auch händisch hinzugefügt.

Der Feldversuch zeigte, dass der Ansatz ‚Vorgehensmodell/Prozess-Interpretierer' in vielen Fällen gangbar ist und eine Verbesserung der Planungsarbeit bringt. Ein weitere Vorteil ist die Möglichkeit, nach Ende des Einsatzes den gesamten Vorgang nachzuvollziehen und zu analysieren.

5.3 Erfahrungen und offene Herausforderungen

Bei der Durchführung wurden einige Probleme identifiziert:

- Die kritische Abhängigkeit von einer funktionierenden Infrastruktur (Internet!) wurde deutlich. Eine Zeiterfassung durch die Helfer wie in der klassischen Arbeitswelt, ist wegen Stress und des schwierigen Umfeldes im Einsatz kaum möglich.
- Für den realen Einsatz ist Unterstützung für die genaue Abrechnung von Personen und Material erforderlich, obwohl wegen des Zeitdrucks wenig Zeit für Aufzeichnungen bleibt.
- Bei Softwareprodukten wird meist in Halbtagen oder Stunden abgerechnet, hier braucht man wegen der Kürze mancher Aktivitäten und für Synchronisierung und post-mortem Analysen minutengenaue Aufzeichnungen.
- Das Projektmanagement-Werkzeug kann eigentlich nur für Einsatzleitungen, nicht für die Einsatzkräfte selbst verwendet werden

Als weiterzuverfolgende Fragen, die sich teilweise auch im Software-Engineering stellen sind, wurde u.a. identifiziert:

- Welche Informationen kann der Prozess-Interpretierer verwenden, um gezielte Empfehlungen abzugeben?
- Kann der Interpretierer die Richtigkeit/Plausibilität von Status-Attributen (Aktivitäten und Resultate) beurteilen?
- Wieviel Unterstützung kann ein Prozess-Interpretierer geben, besonders in Hinblick auf logische oder semantische Abhängigkeiten zwischen Resultaten. Dies gilt gleichermaßen für Korrekturen während der Prozessabwicklung.
- In wieweit kann ein Prozess-Interpretierer mittels der Methoden der Artificial Intelligence verbesserte Selektionskriterien für Aktivitäten vorschlagen?
- Welche anderen Gebiete sind auch prädestiniert für den Einsatz des Paradigmas 'Vorgehensmodell/Prozess-Interpretierer', z.B. e-Health, Service-Industrie, Crowd-Tasking, etc.?

6 Zusammenfassung

Katastropheneinsätze sind und werden immer wesentlich auf dem Einsatz von Menschen (‘First Responder’) beruhen. Trotzdem zeigt sich auch in vielen Gebieten, die im wesentlichen menschlichen Einsatz und Intelligenz erfordern, dass Computer-Unterstützung hilfreich sein kann.

In diesem Paper wurden grundlegende Fragen und Anforderungen der Unterstützung von Projekten durch Prozess-Interpretierer im Rahmen des Paradigmas ’Vorgehensmodell/Prozess-Interpretierer’ diskutiert. Die Betonung lag auf notwendigen/wünschenswerten Funktionalitäten des Prozess-Interpretierers.

Besonderes Augenmerk wurde auf das Katastrophenmanagement gelegt. In einem Feldversuch wurde der Einsatz eines spezifischen Vorgehensmodells im Katastrophenmanagement getestet und prinzipiell als positiv beurteilt. Es zeigte sich, dass Katastropheneinsatz und konventionelle IKT-Anwendungen vieles gemeinsam haben, dass jedoch besonders in der Interventionsphase einer Katastrophe zusätzliche Erschwernisse und Probleme gelöst werden müssen.

Danksagung

Der Feldversuch wurde durch die Österreichische Forschungsförderungsgesellschaft (FFG) gefördert (Innovationsscheck "CanDo im Krisen- und Katastrophenmanagement " no. 844636, 2014 [Ha14]).

Literaturverzeichnis

[Au15a] Aumayr, G.; Chroust, G.; Haider, G.; Randus, R.; Thür, A.: Disaster-Management: Challenges for Computer-supported Process and Project Management (Abstract 2508). In (Ison, R.L., ed..): Governing the Anthropocene: The greatest challenge for systems thinking in practice? - Program and Abstracts. ISSS, International Society for Systems Sciences, 2015, S. 107–109, 2015.

[Au15b] Aumayr, G.; Chroust, G.; Haider, G.; Randus, R.; Thür, A.: Disaster-Management: Challenges for Computer-supported Process and Project Management (Präsentation). unpublished, 2015.

[BBL76] Boehm, B.W.; Brown, J.R.; Lipow, M.: Quantitative Evaluation of Software Quality. Proc. 2nd ICSE, San Francisco, S. 592–605, 1976.

[Be02] Beck, K. et al.: Manifesto for Agile Software Development. http://agilemanifesto.org/, 2001, 2002.

[BH11] Benestad, Hans Christian; Hannay, Jo E.: A Comparison of Model-based and Judgment-based Release Planning in Incremental Software Projects. In: Proceedings of the 33rd International Conference on Software Engineering. ICSE '11, ACM, New York, NY, USA, S. 766–775, 2011. Benestad:2011:CMJ:1985793.1985901.

[Bo76] Boehm, B.W.: Software Engineering. IEEE Trans on Computers vol. C-25, no. 12, S. 1226–1241, 1976.

[Bo86] Boehm, B.: A Spiral Model of Software Development and Enhancement. ACM SIGSOFT - Software Engineering Notes, 11:4,:22–42, 1986.

[BR05] Broy, M.; Rausch, A.: Das neue V-Modell(R) XT - Ein anpassbares Modell für Software und System Engineering. Informatik-Spektrum, vol. 28 (2005), no. 3, S. 220–229, 2005.

[BT15] Boehm, Barry; Turner, Richard: The Incremental Commitment Spiral Model (ICSM): Principles and Practices for Successful Systems and Software. In: Proceedings of the 2015 International Conference on Software and System Process. ICSSP 2015, ACM, New York, NY, USA, S. 175–176, 2015. :2015:ICS:2785592.2785619.

[CA14] Chroust, G.; Aumayr, G.: Process Models for Disaster Management - Standardization and Assessment. In (Hofkirchner, W.; Blachfellner, Stefan, ed.): EMCSR 2014 - book of Abstracts. BCSSS 2014, S. 5, 2014.

[CGMS88] Chroust, G.; Gschwandtner, O.; Mutschmann-Sanchez, D.: Das Entwicklungssystem ADPS der IBM. Gutzwiller T., Österle H. (eds.): Anleitung zu einer praxisorientierten Software-Entwicklungsumgebung, Band 2, AIT Verlag München, S. 123–148, 1988.

[Ch89] Chroust, G.: Application Development Project Support (ADPS) - An Environment for Industrial Application Development. ACM Software Engineering Notes vol 14 (1989, no. 5), S. 83–104, 1989.

[Ch95] Chroust, G.: Interpretable Process Models for Software Development and Workflow. Schaefer W. (ed.): Software Process Technology - 4th European Workshop EWSPT'95, Noordwijkerhout, Springer Lecture Notes No. 913, Springer Berlin-Heidelberg, S. 144–153, 1995.

[Ch00] Chroust, G.: Software Process Models: Structure and Challenges. In: Feng, Y. and Notkin, D. and Gaudel, M.C.(ed.): Software: Theory and Practice - Proceedings, IFIP Congress 2000, Aug. 2000, Beijing. Kluwer, S. 279–286, 2000.

[Ch12] Chroust, G.: ICT Support for Disaster Management. In (Doucek, P.; Chroust, G.; Oskrdal, V., Hrsg.): IDIMT 2012 ICT-Support for Complex Systems, vol.38 Sept 2012. Trauner Verlag Linz, 2012, S. 13–23, 2012.

[CKS10] Chroust, G.; Kuhrmann, M.; Schoitsch, E.: Modeling Software Development Processes. In (Cruz-Cunha, M. M., Hrsg.): Social, Managerial and Organizational Dimensions of Enterprise Information Systems. IGI-Publishing, Hershey, USA 2010, S. 31–62, 2010.

[Ha14] Haider, G.; Chroust, G.; Kaundert, M.; Thür, A.; Randus, R.; G., Aumayr.: CanDo im Krisen- und Katastrophenmanagement - Lastenheft. Bericht, Johanniter Österreich Ausbildung und Forschung gem. GmbH, Okt. 2014.

[HH08] Höhn, R.; Höppner, S: Das V-Modell XT. Springer Lehrbuch, 2008

[Hu80] Huenke, H., Hrsg. Software Engineering Environments. Proceedings, Lahnstein, BRD, North Holland, 1980.

[IB78] IBM Corp.: DV-Verfahrenstechnik - Eine methodische Vorgehensweise zur Entwicklung von DV-Anwendungen. Schriftenreihe Management- und Methoden-Institut, IBM Deutschland, Form No. SR12-1657-0, 1978.

[IF07a] IFRC (ed.): Contingency planning guide. Bericht, International Federation of Red Cross and Red Crescent Societies, 2007.

[IF07b] IFRC (ed.): Disaster response and contingency planning guide. Bericht, International Federation of Red Cross and Red Crescent Societies, 2007.

[IN12] INSARAG (ed.): INSARAG Guidelines and Methodology Technical Report. Bericht, International Search and Rescue Advisory Group United Nations Office for the Coordination of Humanitarian Affaires, 2012.

[IS11] ISO: ISO 22320:2011, Societal security – Emergency management – Requirements for incident response. Bericht, International Organization for Standardization, 2011.

[IS12] ISO/IEC: ISO/IEC CD 33001.5 Information Technology - Process Assessment - Concepts and terminology. Bericht, International Organization for Standardization (2012-09-09), 2012.

[KK16] Kneuper, R.; Kneuper, R.: Process Mining be Software-Prozessen. Softwaretechnik-Trends vol 36 (2016), no. 1, S. 16–19, 2016.

[Kn06] Kneuper, R.: CMMI - Verbesserung von Softwareprozessen mit Capability Maturity Model Integration. 2. verb. Auflage, dpunkt.verlag 2006.

[La13] Lazarte, M.: , New ISO standard for emergency management. http://www.iso.org/iso/home/news_index/ news_archive/news.htm?refid=Ref1496, [Feb. 2014], 2013.

[Ma15] Makeit GmbH: , Can Do: Projektmanagement. http://www.makeit.at/ web/angebot/werkzeuge/cando [2015-08-22], 2015.

[Os87] Osterweil, J.L.: Software Processes are Software Too. In: 9th International Conference on Software Engineering (ICSE 1987). 1987.

[Pa10] Parr, T.: Language Implementation Patterns: Techniques for Implementing Domain-Specific Languages. O'Reilly, 2010.

[Ph89] Phillips, R.W.: State Change Architecture Protocols for Process Models. IEEE Proceedings of the Hawaii International Conference on System Sciences (HICSS-22), Kona, Hawaii, January 3-6, 1989.

[Ro70] Royce, W.W.: Managing the Development of Large Software Systems. Proc. IEEE WESCON, Aug. 1970, S. 1–9, 1970.

[Sc12] Schoitsch, E.: Cyber-Physical Systems (CPS) - What can we learn from Disasters with respect to Assessment, Evaluation and Certification/Qualification of 'Systems-of-Systems'? In (Doucek, P.; Chroust, G.; Oskrdal, V., Hrsg.): IDIMT 2012 ICT-Support for Complex Systems, vol.38 Sept 2012. Trauner Verlag Linz, 2012, S. 69–81, 2012.

[SRC15] Schryer, G.; Rauchecker, G.; Comes, T.: Resource Planning in Disaster Response - Decision Support Models and Methodologies. Business & Information Systems Engineering, vol. 57 (2015), no. 2, S. 243–259, 2015.