

Entscheidungsunterstützung im Operativen Prozessstörungenmanagement

Jürgen Kuster

Institut für Angewandte Informatik, Universität Klagenfurt
jkuster@ifit.uni-klu.ac.at

Abstract: Zentrales Ziel des operativen Prozessstörungenmanagements ist es, im Falle unerwarteter Zwischenfälle angemessene Reparaturmaßnahmen zu identifizieren und so die negativen Auswirkungen auftretender Störungen zu minimieren. Da es sich dabei um eine äußerst komplexe Aufgabe handeln kann, sollen entscheidungsbefugte Prozessmanager durch intelligente Systeme unterstützt werden. In der hier zusammengefassten Dissertation wird daher ein Rahmenwerk zur Bereitstellung von Entscheidungsunterstützung im operativen Prozessstörungenmanagement vorgeschlagen. Durch die Erweiterung des etablierten Resource Constrained Project Scheduling Problems (RCPSP) um das Konzept alternativer Aktivitäten wird es möglich, praktisch relevante Probleme und potentielle Reparaturmaßnahmen umfassend zu modellieren. Für die Lösung des so genannten x -RCPSP wird ein äußerst effizienter genetischer Algorithmus vorgeschlagen. Durch den Einsatz von Konzepten der lokalen Optimierung können damit innerhalb nur weniger Sekunden sehr gute bis optimale Lösungen für Störungsmanagementprobleme realistischer Größe identifiziert werden. Für die Integration der so entwickelten unterstützenden Systeme in bestehende Entscheidungsprozesse ist vor allem die Wahl eines human-centered approach von zentraler Bedeutung.

1 Einführung

Operatives Prozessstörungenmanagement (Disruption Management, DM) befasst sich mit der Bewältigung von Zwischenfällen, die im Zuge der Abwicklung eines Zeitplans unerwartet auftreten. Zentrales Ziel dabei ist es, angemessene Reparaturmaßnahmen zu identifizieren und so allfällige negative Auswirkungen von Störungen zu minimieren. Etliche Abhängigkeiten, Zeitkritikalität und geringe Datenqualität machen DM im realistischen Umfeld häufig zu einer äußerst komplexen Aufgabe. Es ist daher wünschenswert, jene Personen, die für die Behandlung von Prozessstörungen verantwortlich sind, informativonstechnisch und mithilfe intelligenter Verfahren zu unterstützen.

Neben der Verbesserung von Informationsverfügbarkeit und -qualität stellt insbesondere die Anwendung von Konzepten der Künstlichen Intelligenz (Artificial Intelligence, AI) eine vielversprechende Möglichkeit solcher Unterstützung dar. Dabei können automatisch die verfügbaren Informationen analysiert, mögliche Interventionen bewertet und so angemessene Handlungsvorschläge generiert werden. Obwohl die technischen Voraussetzungen für entsprechende Lösungen vielerorts gegeben sind, wurden bislang kaum solche Entscheidungsunterstützungssysteme (Decision Support Systems, DSS) realisiert. Die folgenden zentralen praktischen Probleme können zusammengefasst werden:

- Bestehende Ansätze des Prozessstörungsmanagements (siehe [YQ04, CHLL01]) konzentrieren sich auf die bloße Anpassung von Zeit- und Ressourcenzuweisungen. Komplexere Interventionen wie der Wechsel eines zuvor gewählten Prozessausführungspfades (im Zuge von Parallelisierung/Serialisierung, Einfügen/Entfernen, Austausch von Prozessschritten, etc.) werden nicht berücksichtigt.
- Es existiert kein konzeptuelles Framework für die umfassende Modellierung von Störungsmanagementproblemen. Ansätze des Disruption Managements beruhen häufig auf klassischem Planen und Scheduling. Die entsprechenden Modelle sind zwar weit verbreitet und ausgereift, für die Abbildung realistischer Probleme jedoch häufig nicht ausdrucksstark genug.
- Ein hoher Grad kombinatorischer Komplexität beschränkt die Größe jener Probleme, die in vernünftiger Zeit bewältigt werden können. In der Forschung zur Lösung von Planungs- und Schedulingproblemen werden üblicherweise deutlich weniger Prozessschritte und Aktivitäten betrachtet als im realistischen Umfeld notwendig.

Ziel der in diesem Dokument zusammengefassten Dissertation “Providing Decision Support in the Operative Management of Process Disruptions” ist es, geeignete Werkzeuge zur Bewältigung dieser Probleme zur Verfügung zu stellen. Zu diesem Zweck wird ein *Rahmenwerk für die Bereitstellung von Entscheidungsunterstützung im operativen Prozessstörungsmanagement* entworfen. Es bietet umfassende Möglichkeiten, praktisch relevante Probleme realistischer Größe zu modellieren und dafür in Echtzeit gute bis optimale Lösungen zu identifizieren. Kritische Erfolgsfaktoren für die Einführung intelligenter Systeme in bestehende Entscheidungsprozesse werden ebenfalls erörtert.

Das vorliegende Dokument ist wie folgt strukturiert. In Abschnitt 2 wird das in der Dissertation eingeführte Konzept des *x-RCPSP* präsentiert. Die darin entwickelte Erweiterung des etablierten Resource Constrained Project Scheduling Problems (RCPSP) ermöglicht die formale Beschreibung von Störungsmanagementproblemen. Abschnitt 3 beschreibt, wie das *x-RCPSP* zur automatisierten Bereitstellung von Entscheidungsunterstützung eingesetzt werden kann. Insbesondere wird ein genetischer Algorithmus präsentiert und aufgezeigt, wie das Konzept lokaler Optimierung zu signifikanten Performancesteigerungen beitragen kann. In Abschnitt 4 werden kritische Erfolgsfaktoren für die Einführung intelligenter Systeme in bestehende DM Entscheidungsprozesse skizziert bevor Abschnitt 5 die wesentlichen Beiträge der Dissertation zusammenfasst.

2 Modellierung von Störungsmanagementproblemen

Zur Modellierung von Störungsmanagementproblemen werden typischerweise bestehende Frameworks aus den Bereichen automatisierter Planung und Scheduling verwendet. Von den unzähligen verfügbaren Konzepten wurde hier das so genannte Resource Constrained Project Scheduling Problem (siehe [BLK83, BDM⁺99]) als Ausgangspunkt gewählt. Es stellt ein generisches Rahmenwerk für die Formalisierung von Scheduling Problemen zur Verfügung, das insbesondere die folgenden Vorteile aufweist:

- Anders als in produktionsspezifischen Problemklassen (wie Job Shop, Flow Shop Problemen, etc.) sind im RCPSP weder die Struktur der modellierten Prozesse noch Art und Anzahl beschriebener Ressourcenanforderungen beschränkt.
- Es existieren hocheffiziente Algorithmen für die Erstellung von Zeitplänen (Schedules) auf Grundlage des RCPSP (siehe zum Beispiel [KH06] für einen Überblick).
- Aktivitäten, Vorgängerbeziehungen, Ressourcen und zugehörige Anforderungen stellen die zentralen Modellierungskonstrukte des RCPSP dar. Damit können Probleme relativ einfach auf konzeptueller Ebene formalisiert werden.

Zwar ist das RCPSP damit sehr gut zur Modellierung von klassischen Scheduling Problemen geeignet, weiterreichende strukturelle Flexibilität kann darin jedoch nicht beschrieben werden. So ist es etwa nicht möglich, Optionen wie das dynamische Einfügen/Entfernen, die Parallelisierung/Serialisierung von Prozessschritten oder Änderungen der Ausführungsreihenfolge zu berücksichtigen. Aus diesem Grund wird vorgeschlagen, das RCPSP um das Konzept *alternativer Aktivitäten* zu erweitern. Im Extended RCPSP (*x-RCPSP*, siehe [KJF07]) werden dementsprechend aktive und inaktive Aktivitäten unterschieden. Durch die Aktivierung und Deaktivierung von optionalen Prozessschritten (anhand vordefinierter Regeln und Einschränkungen) kann im Zuge der Optimierung ein gewählter Prozessausführungspfad dynamisch variiert werden.

Formal kann das *x-RCPSP* wie folgt zusammengefasst werden. Ein Prozess besteht aus einer Menge möglicher Aktivitäten $\mathcal{A}^+ = \{0, 1, \dots, a, a + 1\}$. Erstes und letztes Element stellen abstrakte Start- und Endaktivitäten dar, die eine Dauer von 0 und keine Ressourcenanforderungen aufweisen. Alle übrigen $i \in \mathcal{A}^+$ haben eine beliebige nicht-negative Dauer d_i . Die Menge $\mathcal{A} \subseteq \mathcal{A}^+$ gruppiert alle aktiven Aktivitäten, die Differenz $\mathcal{A}^+ \setminus \mathcal{A}$ dementsprechend alle inaktiven Prozessschritte. $\mathcal{R} = \{1, \dots, r\}$ stellt die Menge wieder verwendbarer Ressourcenklassen dar. Für jeden Typ $k \in \mathcal{R}$ ist eine konstante Kapazität c_k für die Ausführung der vorgesehenen Aktivitäten verfügbar. Die Modellierung von Aktivitätsbeziehungen beruht auf den folgenden Konstrukten:

- Die Elemente in \mathcal{A}^+ werden anhand von Vorgängerbeziehungen geordnet. \mathcal{P}^+ enthält die entsprechenden Constraints, wobei $p_{i,j} \in \mathcal{P}^+$ definiert dass Aktivität $i \in \mathcal{A}^+$ spätestens zu Beginn der Aktivität $j \in \mathcal{A}^+$ beendet sein muss.
- Die Beziehung zwischen Aktivitäten und Ressourcen wird durch Ressourcenanforderungen beschrieben. Dazu werden in der Matrix Q^+ sämtliche Elemente $i \in \mathcal{A}^+$ mit den Elementen $k \in \mathcal{R}$ kombiniert. $q_{i,k} \in Q^+$ beschreibt wie viele Einheiten von Ressource k zur Ausführung von i benötigt werden.
- Ersetzungsregeln beschreiben zulässige Aktivierungen und Deaktivierungen. Das Vorkommen eines Elements $x_{i,j}$ in der Menge erlaubter Substitutionen \mathcal{X}^+ besagt, dass die Aktivierung von $j \in \mathcal{A}^+ \setminus \mathcal{A}$ im Gegenzug für die Deaktivierung von $i \in \mathcal{A}$ eine gültige Austauschoperation darstellt: i kann durch j ersetzt werden.
- Abhängigkeiten zwischen den Aktivierungszuständen einzelner Aktivitäten werden in der Menge \mathcal{M}^+ beschrieben. So kann ausgedrückt werden, dass $i \in \mathcal{A}^+$ bei Aktivierung oder Deaktivierung von $j \in \mathcal{A}^+$ aktiviert oder deaktiviert werden muss.

Jede gültige Kombination von Aktivierungszustand und zugehörigen Aktivitätsstartzeiten stellt eine mögliche Lösung des x -RCPSP dar. Eine Schedule S kombiniert demnach die Startzeiten β_i für alle aktiven Aktivitäten $i \in \mathcal{A}$ in einem Vektor $(\beta_1, \beta_2, \dots, \beta_n)$.

Im betrachteten Kontext kann ein operatives Störungsmanagementproblem damit als Kombination der folgenden Elemente beschrieben werden.

- Eine x -RCPSP Instanz definiert die Struktur der betrachteten Prozesse sowie die gegebenen Interventionsmöglichkeiten. Eine bestehende Schedule beschreibt jene zuvor identifizierte Lösung des x -RCPSP, die zur Ausführung vorgesehen ist.
- Eine unvorhergesehen auftretende Störung wird von ihrem Typ und einer Menge zugehöriger Parameter definiert. Mögliche Formen von Störungen (so genannte Rescheduling Factors) sind zum Beispiel in [VHL03, ZBY05] zusammengefasst.
- Eine Kostenfunktion $\varphi : S \rightarrow \mathbb{R}$ ermöglicht die Bewertung eines Zeitplans S . Damit können Schedules verglichen und so das Ziel der Optimierung definiert werden.

3 Automatisierte Lösung von Störungsmanagementproblemen

In diesem Abschnitt wird beschrieben, wie auf Grundlage des x -RCPSP modellierte Störungsmanagementprobleme automatisiert gelöst werden können. Zunächst wird ein Überblick über das allgemeine Vorgehen gegeben bevor ein grundlegender genetischer Algorithmus zur Optimierung skizziert wird. Abschließend wird aufgezeigt, wie das Konzept lokaler Optimierung zu signifikanten Performancesteigerungen eingesetzt werden kann.

3.1 Überblick

Vorrangiges Ziel des operativen Prozessstörungsmanagements ist es, im Falle unvorhergesehener Zwischenfälle eine optimale Kombination von Reparaturmaßnahmen zu finden. Für das zuvor definierte DM Problem soll jene Menge an Zeit-, Ressourcen- und Strukturanpassungen identifiziert werden, die für die gegebene x -RCPSP Instanz – unter Berücksichtigung der auftretenden Störung und sämtlicher Nebenbedingungen – machbar ist und gleichzeitig φ minimiert. Zur automatisierten Lösung dieses Problems in einem entscheidungsunterstützenden System kann wie folgt vorgegangen werden:

1. Der gegebene Zeitplan wird gemäß der auftretenden Störung (und der im x -RCPSP definierten Beziehungen) aktualisiert. Dazu werden die Auswirkungen des Zwischenfalls in den zukünftigen Teil der vorgesehenen Schedule integriert. Der neue Zeitplan beschreibt was passieren würde, wenn keine Intervention erfolgt.
2. Der zukünftige Teil des aktualisierten Zeitplans wird optimiert. Durch Verschieben von Aktivitäten und Ressourcen sowie die Evaluierung alternativer Ausführungspfade wird versucht, φ soweit wie möglich zu minimieren. Im praktischen Kontext ist dieser Schritt der Optimierung üblicherweise zeitlich limitiert.

3. Resultat der Optimierung ist ein modifizierter Zeitplan, der die beste bekannte Möglichkeit zur Ausführung der noch anstehenden Prozessschritte beschreibt. Durch den Vergleich mit der ursprünglichen Schedule gilt es, jene konkreten Interventionen abzuleiten, die den aktuellen Zeitplan auf den optimalen Zeitplan überführen.

In weiterer Folge wird vor allem der zweite, zentrale Schritt – die Optimierung des Zeitplans – betrachtet. Es wird dabei zunächst aufgezeigt, wie ein bestehender RCPSp-spezifischer Algorithmus zu diesem Zweck erweitert werden kann. Danach wird ein neues Verfahren lokaler Optimierung zur Steigerung der Performance vorgeschlagen.

3.2 Evolutionäre Optimierung in einem Genetischen Algorithmus

Die automatisierte Lösung von operativen DM Problemen ist vor allem deswegen komplex, weil naturgemäß nur sehr wenig Zeit zur Identifikation geeigneter Lösungen zur Verfügung steht, die Identifikation eines exakten Optimums für das NP-harte x -RCPSp jedoch schwierig und zeitaufwändig ist. Zum Umgang mit diesem Problem wird der Einsatz eines heuristischen Verfahrens vorgeschlagen: Damit kann innerhalb von wenig verfügbarer Zeit eine *möglichst gute Lösung* gefunden werden. Weil die Nähe zwischen ursprünglicher und optimierter Schedule im DM von besonderer Bedeutung ist, und weil genetische Algorithmen (GA) für das RCPSp besonders gute Resultate liefern, wird hier skizziert wie ein RCPSp-spezifischer GA für das x -RCPSp erweitert werden kann.

In einem genetischen Algorithmus beruht Optimierung auf der kontinuierlichen Weiterentwicklung einer Population von Lösungen. Jede Generation besteht dabei aus den besten Lösungen der vorangegangenen Generationen sowie deren Kindern, die im Wesentlichen das Resultat von Kreuzung und Mutation sind. Ein genetischer Algorithmus wird dadurch definiert, wie Lösungen repräsentiert, erstellt, bewertet, selektiert, gekreuzt und mutiert werden. Der vorgeschlagene GA kann damit wie folgt zusammengefasst werden:

- *Repräsentation und Initialisierung.* Es ist üblich, für die Zeitplanoptimierung eine abstrakte Repräsentation der Lösungen heranzuziehen. Der hier vorgeschlagene GA verwendet einfache Aktivitätslisten, die jederzeit eindeutig in zugehörige Schedules überführt werden können. Dazu werden die einzelnen Aktivitäten in der durch die Liste vorgegebenen Reihenfolge in den korrespondierenden Zeitplan eingefügt.

Der ursprünglich gültige Zeitplan wird zur Initialisierung des GA herangezogen: Die entsprechende Aktivitätsliste kann durch die simple Sortierung der enthaltenen Aktivitäten nach ihrem Startzeitpunkt erstellt werden. Neben dieser ersten Lösung besteht die erste Generation aus etlichen Mutationen der Ausgangsliste.

- *Bewertung und Selektion.* Die in einer Generation enthaltenen Aktivitätslisten werden zur Bewertung in die korrespondierenden Zeitpläne überführt und dann anhand von φ evaluiert. Die besten Lösungen werden unverändert übernommen, zusätzliche neue Lösungen werden durch Kreuzung und Mutation guter Vorgänger erstellt.

- *Kreuzung*. Die zentrale Schwierigkeit einer Kreuzung von Aktivitätslisten im Kontext des x -RCPSP resultiert aus der Tatsache, dass die beiden Ausgangslisten unterschiedliche Aktivierungszustände repräsentieren und damit unterschiedliche optionale Aktivitäten enthalten können. In diesem Fall soll ein Elternteil den Aktivierungszustand, der andere Elternteil die bevorzugte Reihenfolge der enthaltenen Aktivitäten bestimmen. Zum Umgang mit den unterschiedlichen Elementen wird ein Übergangssatz \mathcal{T} identifiziert, das im Wesentlichen beschreibt, welche Substitutionen notwendig sind, um die Listeninhalte anzugleichen. Immer dann wenn die Listen von vornherein die exakt gleichen Elemente enthalten, kann ein existierendes RCPSP-spezifisches Verfahren direkt angewandt werden.
- *Mutation*. Unter Mutation versteht man das leichte, zufällige Verändern eines Teils der überlebenden Lösungen, um damit die frühzeitige Konvergenz zu lediglich lokalen Optima zu vermeiden. Im Kontext des x -RCPSP kann entweder direkt ein RCPSP-spezifisches Verfahren angewandt oder aber eine beliebige (x -RCPSP-spezifische) Substitution gemäß \mathcal{X}^+ durchgeführt werden.

Zur Messung der Performance wurde der vorgeschlagene Algorithmus in einem Java-basierten Scheduling Framework implementiert und auf 320 automatisiert generierte Testfälle (in 32 verschiedenen komplexen Problemklassen) angewandt. Dabei konnte insbesondere beobachtet werden, dass für Probleme mit 100 Aktivitäten etwa 75% des insgesamt bekannten Optimierungspotentials bereits nach 15 Sekunden der Optimierung ausgeschöpft waren. Der vorgeschlagene GA konvergiert damit rasch zu hochqualitativen Lösungen.

3.3 Lokale Optimierung und Partielles Rescheduling

Eine große Anzahl an Prozessschritten, Nebenbedingungen und Optionen macht die automatisierte Lösung von operativen Störungsmanagementproblemen im praktischen Kontext häufig komplex. Es ist daher wünschenswert, die Größe des betrachteten Problems weitestgehend zu reduzieren. Dies kann zum Beispiel durch die Anwendung partieller Reschedulingverfahren erreicht werden. Leider ist es jedoch so, dass bestehende Ansätze wie Affected Operations Rescheduling (AOR, siehe [LSA93]) und Matchup Scheduling (MUP, siehe [BBMN91]) für produktionsspezifische Problemklassen konzipiert und daher nicht ohne weiteres für allgemeinere Probleme (wie das RCPSP oder das x -RCPSP) anwendbar sind. Um dieses Defizit zu beheben, wird ein neues Verfahren namens Local Rescheduling (LRS) vorgeschlagen.

LRS beruht auf der Idee, Störungen direkt dort zu behandeln wo sie sich auswirken. Anstatt unmittelbar sämtliche zukünftigen Aktivitäten zu betrachten, wird zunächst versucht Störungsmanagementprobleme *lokal* zu lösen. Im LRS Verfahren wird daher zuerst lediglich innerhalb eines relativ engen Zeitfensters optimiert. Dieses Zeitfenster wird schrittweise erweitert bis schließlich der gesamte zukünftige Teil des Zeitplans betrachtet wird. Die Anwendung von LRS kann immer dann empfohlen werden wenn (1) die Ausgangsschedule (beinahe) optimal ist und (2) Möglichkeiten zur lokalen Behandlung von Störungen

gen wahrscheinlich sind. Diese Voraussetzungen sind jedoch in den meisten praktischen Anwendungen gegeben. Zum einen steht meist ausreichend Zeit zur umfassenden Optimierung eines anfänglichen Zeitplans zur Verfügung. Zum anderen wird auch die Arbeit von menschlichen Entscheidungsträgern durch lokale Reparaturmöglichkeiten unterstützt. Schließlich konzentrieren sich auch diese zunächst auf enge Zeitfenster und betrachten weiterreichende Optionen nur wenn dafür ausreichend Zeit zur Verfügung steht.

Im Wesentlichen sind die folgenden drei Schritte elementar für eine konkrete Ausgestaltung des LRS Verfahrens:

1. *Initialisierung des Zeitfensters.* Vor der ersten Iteration wird zunächst das initiale Zeitfenster bestimmt. Dazu wird die mit Art und Ausgestaltung der auftretenden Störung verbundene Information verwendet, um zu analysieren in welchem Zeitraum mit Auswirkungen zu rechnen ist.
2. *Optimierung innerhalb des Zeitfensters.* Die partielle Optimierung des Zeitplans beruht im Wesentlichen auf drei Schritten. Zunächst wird ein partielles Optimierungsproblem für das gewählte Zeitfenster extrahiert. Dieses wird mithilfe eines beliebigen Optimierungsverfahrens gelöst. Schließlich wird die so erlangte Teillösung mit der Information zu sämtlichen nicht betrachteten Aktivitäten integriert.
3. *Erweiterung des Zeitfensters.* In jeder Iteration des LRS Verfahrens wird das betrachtete Zeitfenster erweitert. Die entsprechende Vergrößerung kann gemäß verschiedenster Muster erfolgen. Zum Beispiel kann die Größe der Erweiterung in jeder Iteration konstant bleiben, exponentiell anwachsen oder aber logarithmisch kleiner werden. Die optimale Strategie hängt im Wesentlichen von der konkret betrachteten Problemklasse ab.

Zur Messung der Performancesteigerungen, die durch den Einsatz von LRS möglich sind, wurde der zuvor skizzierte GA mit einem (auf exponentiellen Erweiterungen des Zeitfensters basierenden) LRS Verfahren integriert. Damit konnte für Probleme mit 100 Aktivitäten der Anteil des innerhalb von 15 Sekunden ausgeschöpften Optimierungspotentials von rund 75% (siehe Abschnitt 3.2) auf rund 85% erhöht werden. Noch aussagekräftiger ist die Tatsache, dass für große Probleme mit 1000 Aktivitäten der Anteil des innerhalb von 5 Sekunden ausgeschöpften Potentials von rund 15% auf rund 70% anstieg. Da LRS besonders signifikante Performancesteigerungen liefert, wenn die betrachteten Probleme groß sind und nur sehr wenig Zeit für die Optimierung zur Verfügung steht, kann im Allgemeinen der Einsatz dieses Verfahrens im praktischen Kontext empfohlen werden.

4 Integration in Bestehende Entscheidungsprozesse

Die mangelhafte Integration von DSS in bestehende Entscheidungsprozesse hat mitunter negative Auswirkungen auf die Organisation, das Berufsbild der Entscheidungsträger und die einzelnen Prozessmanager selbst. Insbesondere sind Veränderungen von Kommunikations- und Machtstrukturen, die Wahrnehmung zusätzlicher Kontrollmechanismen sowie

Verschiebungen der zugewiesenen Aufgabenbereiche möglich. Eng damit verbunden ist das Risiko der Ablehnung des Systems durch die dafür vorgesehenen Benutzer. Daneben sind bei einem falsch gewählten Grad der Automatisierung vor allem die folgenden Gefahren für das Erreichen einer beabsichtigten Effizienzsteigerung relevant (siehe [Par00]):

- *Mental Workload.* Durch die Einführung eines intelligenten Systems kann es zu einem Gleichbleiben oder gar Ansteigen des Arbeitsaufwandes kommen, wenn durch mangelhafte Integration zusätzliche Arbeitsschritte notwendig werden.
- *Reduced Situation Awareness.* In blindem Vertrauen auf das System kann es zu einem Abbau von Aufmerksamkeit kommen. So werden mitunter wesentliche (vom System übersehene oder nicht verarbeitbare) Zusammenhänge übersehen.
- *Automation Bias.* Bei gutem Funktionieren des Systems ist der Entscheidungsträger eventuell dazu geneigt, die gemachten Vorschläge unreflektiert zu übernehmen.
- *Skill Degradation.* Wenn komplexe dispositive Tätigkeiten im DSS automatisiert werden, besteht die Gefahr, dass entscheidungsbefugte Prozessmanager das entsprechende Wissen verlieren. Dies resultiert mitunter in zu starker Systemabhängigkeit.

Zum Umgang mit diesen Risiken kann allgemein empfohlen werden, einen *human-centered approach* für Umsetzung, Integration und Ausbau eines DSS zu wählen. Der Mensch soll bei Konzeption und Entwicklung der entsprechenden Systeme stets die zentrale Rolle spielen: Seine Stärken sind zu betonen, seine Aufgaben und Verantwortlichkeiten sind zu definieren und gegenüber dem intelligenten System klar abzugrenzen. Entscheidungsbeauftragte Disponenten sollen so in Bezug auf ihre Rolle und die damit verbundene Verantwortung sensibilisiert werden.

Durch das ausdrückliche Commitment des oberen Managements, klare und ehrliche Kommunikation soll eine projektbezogenen Vertrauensbasis geschaffen werden. Daneben ist der Grad der angestrebten Automatisierung sorgfältig zu wählen und zu planen. Das Risiko der Ablehnung des Systems kann minimiert werden, wenn (1) die organisatorischen Strukturen bereits mittelfristig auf die Einführung des DSS vorbereitet werden, (2) das DSS sowohl informationstechnisch als auch prozedural in die bestehenden Abläufe integriert wird und (3) zukünftige Systembenutzer frühestmöglich in den Entwicklungsprozess des intelligenten Systems eingebunden werden.

Was die Funktionalität des DSS selbst betrifft, so sei auf die zentrale Bedeutung einer Erklärungskomponente hingewiesen. Diese ist dafür verantwortlich, dem Entscheidungsträger sämtliche berücksichtigten Aspekte und Kriterien klar und in einfach verständlicher Weise offen zu legen. Dadurch werden Stärken und Schwächen des Systems kommuniziert und so die tatsächliche Kooperation von Mensch und Maschine gefördert.

Schließlich ist bei der Einführung des unterstützenden Systems auch darauf zu achten, die Entscheidungsträger umfassend zu schulen, Backuplösungen und Notfallstrategien vorzubereiten und insbesondere das organisatorische Wissen langfristig zu erhalten. Nur so kann gewährleistet werden, dass die Bewältigung der organisatorischen Aufgaben auch im Falle eines Ausfalls des intelligenten Systems möglich bleibt.

5 Zusammenfassung

In der hier zusammengefassten Dissertation wird ein generisches Rahmenwerk zur automatisierten Bereitstellung von Entscheidungsunterstützung im operativen Prozessstörungsmanagement präsentiert. Zur Modellierung entsprechender Probleme wird die Erweiterung des etablierten und weit verbreiteten RCPSP um das Konzept alternativer Aktivitäten vorgeschlagen. Das so geschaffene Extended RCPSP (*x-RCPSP*) zeichnet sich unter anderem durch die folgenden Stärken aus:

- Das *x-RCPSP* stellt ein einfaches und *intuitives Werkzeug* zur formalen Beschreibung alternativer Prozessausführungspfade dar. Aktivitäten, Ressourcen und Abhängigkeiten werden auf konzeptueller Ebene formalisiert, was die entsprechenden Modelle leicht verständlich und einfach wartbar macht.
- Das zugrundeliegende Konzept alternativer Aktivitäten ist *ausdrucksstark* genug, um einen Großteil praktisch relevanter Interventionen abzudecken. Damit können insbesondere Möglichkeiten struktureller Modifikationen (wie zum Beispiel das Einfügen/Entfernen und das Parallelisieren/Serialisieren von Prozessschritten, sowie mögliche Änderungen der Ausführungsreihenfolge) modelliert werden.
- Hocheffiziente Algorithmen, die für die Lösung des RCPSP vorgeschlagen wurden, können im Kontext des *x-RCPSP* *wiederverwendet* werden. Dies resultiert im Wesentlichen daraus, dass das *x-RCPSP* eine Generalisierung des RCPSP darstellt.

Die beschriebene Dissertation zeigt auf, wie das *x-RCPSP* zur automatisierten Bereitstellung von Entscheidungsunterstützung im DM eingesetzt werden kann. Es wird dazu ein effizienter RCPSP-spezifischer Algorithmus für das generische *x-RCPSP* ausgebaut. Zur weiteren Steigerung der Performance wird darüber hinaus ein neues Verfahren lokaler Optimierung vorgeschlagen: Local Rescheduling (LRS) verwendet die mit einer auftretenden Störung verbundene Information, um Zeitfenster zu identifizieren, in denen nach (möglichst lokalen) Lösungen für das auftretende Problem gesucht werden soll. Mithilfe dieses Verfahrens wird es möglich, auch Probleme realistischer Größenordnung in Echtzeit zu bewältigen.

Neben der technischen Realisierung werden auch jene organisatorischen Aspekte, die bei der Einführung eines entsprechenden entscheidungsunterstützenden Systems berücksichtigt werden müssen, behandelt. Von zentraler Bedeutung sind der Fokus auf die Menschen, sorgfältige Planung und Vorbereitung, die frühzeitige Einbindung von zukünftigen Systembenutzern sowie die langfristige Sicherung organisatorischen Wissens.

Literatur

- [BBMN91] James C. Bean, John R. Birge, John Mittenthal und Charles E. Noon. Matchup Scheduling with Multiple Resources, Release Dates and Disruptions. *Operations Research*, 39(3):470–483, 1991.

- [BDM⁺99] Peter Brucker, Andreas Drexl, Rolf Möhring, Klaus Neumann und Erwin Pesch. Resource-Constrained Project Scheduling: Notation, Classification, Models, and Methods. *European Journal of Operational Research*, 112(1):3–41, 1999.
- [BLK83] Jacek Błazewicz, Jan Karel Lenstra und Alexander Rinnooy Kan. Scheduling Projects to Resource Constraints: Classification and Complexity. *Discrete Applied Mathematics*, 5(1):11–24, 1983.
- [CHLL01] Jens Clausen, Jesper Hansen, Jesper Larsen und Allan Larsen. Disruption Management. *ORMS Today*, 28(5):40–43, 2001.
- [KH06] Rainer Kolisch und Sönke Hartmann. Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update. *European Journal of Operational Research*, 174(1):23–37, 2006.
- [KJF07] Jürgen Kuster, Dietmar Jannach und Gerhard Friedrich. Handling Alternative Activities in Resource-Constrained Project Scheduling Problems. In Manuela M. Veloso, Hrsg., *IJCAI-07, Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, Seiten 1960–1965, Hyderabad, India, 2007.
- [LSA93] Rong-Kwei Li, Yu-Tang Shyu und Sdash Adiga. A Heuristic Rescheduling Algorithm for Computer-based Production Scheduling Systems. *International Journal of Production Research*, 31(8):1815–1826, 1993.
- [Par00] Raja Parasuraman. Designing Automation for Human Use: Empirical Studies and Quantitative Models. *Ergonomics*, 43(7):931–951, 2000.
- [VHL03] Guilherme E. Vieira, Jeffrey W. Herrmann und Edward Lin. Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. *Journal of Scheduling*, 6(1):39–62, 2003.
- [YQ04] Gang Yu und Xiangtong Qi. *Disruption Management: Framework, Models and Applications*. World Scientific Publishing, Singapore, 2004.
- [ZBY05] Guidong Zhu, Jonathan F. Bard und Gang Yu. Disruption Management for Resource-Constrained Project Scheduling. *Journal of the Operational Research Society*, 56(4):365–381, 2005.



Jürgen Kuster wurde am 30. Juli 1980 in Feldkirch (Österreich) geboren. Nach Abschluss der dortigen AHS absolvierte er seinen Präsenzdienst beim Österreichischen Roten Kreuz. Im Jahr 2000 begann er das Studium der Wirtschaftsinformatik an der TU Wien. Seine Diplomarbeit beschäftigte sich mit geographischen Informationssystemen im Air Traffic Management und ist in Zusammenarbeit mit der FREQUENTIS AG entstanden. Nach Abschluss des Magisterstudiums im Jahr 2004 belegte Jürgen Kuster das Doktoratsstudium der Sozial- und Wirtschaftswissenschaften an der Universität Klagenfurt, das er 2007 mit Auszeichnung abschloss. Seine Dissertation trägt den Titel “Providing Decision Support in the Operative Management of Process Disruptions” und beschäftigt sich mit intelligenten Entscheidungsunterstützungssystemen im operativen Prozessstörungsmanagement. Aktuell ist Jürgen Kuster als freiberuflicher Consultant und Entwickler tätig. Er arbeitet unter anderem für die FREQUENTIS AG und am Online Operating System (OOS). Daneben studiert er Rechtswissenschaften an der Universität Linz.