Development of a Decision Logic to Support Group Improvisation: An Application to Emergency Response

David Mendonça
New Jersey Institute of Technology
djm@njit.edu

W.A. Wallace
Rensselaer Polytechnic Institute
wallaw@rpi.edu

Abstract
This paper reviews recent progress in the development of a computer-based system for supporting improvised group decision making in risky, time-constrained situations. One goal of this research is to realize a system design which is consonant with cognitively-grounded research on improvisation and which contributes to performance on representative tasks. Some of the issues and challenges involved in supporting creative but time-constrained decision making are first outlined. Salient constructs from research on cognition in improvisation, particularly in jazz, are next reviewed in order to frame the discussion of system design principles. The operationalization of these principles is next discussed. The paper concludes with a brief discussion of current work in the evaluation of the system and also outlines areas for further work.

1. Introduction
Technological systems involving hazards are typically managed by experienced personnel guided by well-formulated, pre-determined procedures. These procedures are designed to ensure that operations proceed in a safe and cost-effective manner. Reasoning in the management of such systems normally consists of (1) monitoring operations during normal conditions, (2) selecting a planned-for procedure when an event occurs that could disrupt operations but is part of an emergency or contingency plan and (3) revising a planned-for procedure when needed due to unforeseen conditions such as unavailability of equipment. Unfortunately, these operations are exposed to unexpected contingencies that can disrupt normal operations and require execution of a fourth management task: real-time development and deployment of new procedures. Creative thinking in such situations is necessary in order to prevent degradation of operations, particularly when there is potential for personal injury, property damage or environmental contamination.

The approach taken in this work is to provide computer-based decision support to groups engaged in creative thinking in risky, time-constrained situations. This paper reviews the design and implementation of the latest version of Emergency Management imPROViser (EMPROV), a computer-based prototype system comprised of a decision logic and database designed to support improvised decision making (see [1] for details). To improvise, as defined here, is to rework knowledge to produce a novel action in time to meet the requirements of a given situation. A goal of the design process is to develop a group decision support system whose design is consonant with prior research on cognition in improvisation and which contributes to performance on representative tasks. Some of the issues and challenges involved in supporting creative but time-constrained decision making are first outlined in the context of prior research. Salient constructs from research on cognition in improvisation are then reviewed in order to frame the discussion of system design principles. The paper describes how these constructs have been operationalized and embedded in the architecture of the latest generation of the system; it concludes with a brief discussion of future work, including the evaluation of the system in experimental settings.

2. Background
Operational Risk Management (ORM) is a decision logic to support individual- [2] or group-level [3] reasoning processes in risky, time-constrained situations when the need for plan revision arises. ORM consists of the identification of a real-time event, an assessment of its consequences and a decision to change the planned course of action in real-time [4]. Real-time events are occurrences, such as weather changes and equipment failure, which may force a change in the present course of action. Real-time events are unpredictable and occur without warning; therefore, it is crucial that all such events be sensed by system managers. Re-evaluation of a course of action is necessary if at least one activity is threatened by the real-time event (e.g., a fire truck which is called-for by the plan is out of commission). To ensure safe and cost-effective operations, not only must the
appropriate course of action be selected but deadlines must be met. ORM is a logic for assessing the risk, cost and benefit of an activity and selecting activities to form a course of action.

In contrast to a real-time event, an unexpected contingency prevents the group from making further decisions using the current plan, represented in ORM as a graph of activities and decisions. To complete the response task following the incidence of an unexpected contingency, decision makers must re-build the graph, either by creating new activities and decisions or recombining existing ones in novel ways. Three principal activities following the occurrence of an unexpected contingency are recognizing the need to improvise, conceptualizing an improvised solution and, finally, executing the solution, all under time constraints. A situation such as this reduces the gap between planning and execution, thereby creating an improvisational situation [5], [6]. It is assumed that ORM can be used to diagnose the need to improvise and, if necessary, to assist in execution of plans. The current research therefore focuses on the second stage of the enterprise (i.e., development of an improvised a course of action).

Creative reasoning in operational decision making is likely to be directed towards returning to an appropriate plan. There can be pressures or real obligations associated with creative decision-making under risk, particularly for governmental or quasi-governmental organizations. Examples include legislated emergency response plans or an industry’s standard operating procedures. There is also considerable evidence to suggest that teams in decision settings like emergency management enact strategies based on recognizing characteristics of past problems in the current one [7]. Indeed, as noted by Weick [8] in his seminal study of the Mann Gulch fire, “What we do not expect under life-threatening pressure is creativity.” A sobering conclusion this study is that, under certain conditions, teams may force their conception of the emergency to fit one they know how to address [8].

While it is assumed that ORM will be used to diagnose the need to improvise and to evaluate existing candidate courses of action, there seem to be few tools to assist groups in making creative decisions in time-pressured, risky situations. The approach of the current research is to create decision support systems which are driven by theory concerning thinking during improvisation. Some of this research is reviewed in the following section in order to motivate the discussion of system design principles.

2.1. Cognition in Improvisation

A rich body of recent research in various domains seeks to make observations and develop theories on thinking processes of improvisers. Domains include organizational decision making (see [9] for a recent review), organizing (e.g., the recent special issue of Organization Science (v.5, n.9), particularly [10]) and music performance (e.g., [11], [12], [13], [14]). With some exceptions (e.g., [15]), a common vehicle or metaphor for theorizing about improvising in organizational contexts is jazz improvisation.

Jazz is fundamentally a music of improvisation [14]. According to Berliner [12], improvisation in jazz involves “reworking precomposed material and designs in relation to unanticipated ideas conceived, shaped, and transformed under the special conditions of performance, thereby adding unique features to every creation.” According to Sarath [16], improvisers seem to hold a referent for their improvisation: that is, an underlying format, such as what Pressing [13] has called “the harmonic-rhythmic framework of the composition they play.” A challenge for improvisers is “adhering to the constraints of the format while maintaining spontaneity and interactiveness” [16].

By analyzing transcriptions of various improvisations, researchers have suggested that different realizations may arise from the same referent. Berliner [12], for example, has analyzed different realizations, over the course of forty-six years, of one referent (pp. 576 ff.). Similar work was done by Lord [17] in the context of improvised singing of folk tales. Notwithstanding these observations, there is little empirical research on the thinking processes involved in developing a realization from a referent. By considering how the process has been conceptualized in computer science domains, the current research attempts to bridge this gap for the task of improvising in emergency response.

Research by Hodges [18] links a referent to a realization computationally through the use of a decision logic operating upon an ontology. The ontology is intended to capture the functional capabilities of various mechanical devices, so that alternative devices can be utilized when the device normally used for a task is unavailable. An example is the use of a can opener in place of a nutcracker to crack pecan nuts. The logic recognizes the functional capabilities of a non-standard device by reasoning abstractly about its functional relation to a standard device, all in the context of the current plan and goals [18]. The logic navigates through a hierarchical representation of devices, the situations in which they are usually used, and the functions which relate devices to each other through their properties and the situations in which they are utilized. This hierarchical representation of devices and functions is denoted a “Functional Ontology for Naive Mechanics” (FONM). Given some event, then, which blocks the accomplishment of the current goal (e.g., opening the car door), the logic reasons to find another device which could be used as a substitute.
2.2. Support for Creative Thinking in Groups

Support for creative thinking in groups, not necessarily in the context of operational decision making, has generally been in the form of training in divergent-thinking techniques, such as brainstorming [19], [20], which may or may not be embedded in software [21], [22]. Although following certain creativity heuristics seems to result in more ideas (see, e.g., [23]), it appears that nominal groups outperform actual groups in ideation ([24], [25], [26]). Few studies have attempted to gather empirical evidence concerning the impact of these techniques on decision quality of groups [27] (but see [28] for a consideration of quality in the context of individual creativity).

Other recent work has sought to develop techniques which direct or focus creative thinking. In general, these techniques employ heuristics related to convergent thinking (see [29] for an example using brainstorming and [30] for an example using groupware). One conclusion of this research is that “face-to-face or electronic sharing of ideas is mutual stimulation of associations,” [26] and that this stimulation or priming can occur at multiple levels. For example, groups may naturally generate ideas in certain categories then find themselves being stimulated by emergent associations between ideas within categories. Moreover, “when one is exposed to an idea from a particular category, it will tend to stimulate ideas within this category since associations tend to follow the rule of similarity” [31]. Hilmer and Dennis [30] detected improvements in decision quality when participants were required to integrate information (i.e., to categorize it). However, Paulus’ [26] comment remains true: that “[m]ost studies of work teams have focused on manufacturing or service settings and only a few of these have used objective measures of performance” (p.249, citing [32]). Clearly, additional research is needed in appraising idea quality and group performance on creative decision making, both with and without the aid of software or heuristics.

3. System Design

EMPROV employs a decision logic which acts upon an ontology to provide recommendations to a group making creative, operational-level decisions in a simulated emergency response situation. The system is intended to be used in finding alternative ways to attain emergency response goals given that the need to improvise has been identified. The system is also intended to be collaborative, since its recommendations are driven by the preferred actions of group members.

The two main components of the system are an ontology and a decision logic. The ontology maps functions to goals and to physical resources. This is accomplished by identifying physical resources drawn from the same categories as those which the decision makers intended to utilize but which they may not have considered for use. The logic navigates through the ontology and provides, as output, recommendations concerning which resources might substitute for other, possibly unavailable resources. The domain in which the system operates is first discussed, followed by a description of the structure of the ontology and the design of the logic.

3.1. Problem Domain

EMPROV is being evaluated in experimental sessions with both novice and experienced groups of participants. Each group of five participants convenes to work on two separate emergency response-related cases, both involving shipboard fires with threatened chemical releases. The essence of the group’s task is to decide which resources should be mobilized to the incident location in order best to meet the goals of the response. Each participant acts as the representative of a particular emergency service, with the exception of the group coordinator, who acts as a facilitator and principal communicator with EMPROV. The layout of a typical experimental session is shown in Figure 1:

![Figure 1. Layout of experimental session](image-url)
constraint, it is essential that participants account simultaneously for planning and execution times.

### 3.2. Ontology Structure

The ontology serves as the database upon which the logic acts. Although there is no agreed-upon definition of an ontology, the term has been used to refer to “the shared understanding of some domain of interest which may be used as a unifying framework” to solve problems that “obstruct communication between or among people, organizations and/or software systems” [33]. Guidelines for construction of the ontology were drawn from [34] and [35]. The structure is based on FONM principles; the contents are based on emergency response guidelines; the contents are based on emergency response guidelines in prior experimental sessions. In this section, the ontology is described by providing extended descriptions of its elements (i.e., goals, functions, object groups, objects and properties) and of the relations between these elements. A description begins with a statement of an element’s relation to the element immediately beneath it in the hierarchy. Elements of the ontology are summarized in Table 1.

- **Goal has associated Function(s)**
  
  A goal, G, is a desired situation [37] or state of the world. A given goal (such as treatment of injured persons) has associated with it various functions (such as provision of shelter and transport of personnel) that contribute to its level of accomplishment or attainment. The goal of removing trapped persons from danger would be attained if all such persons were in fact removed. Functions are actions typically undertaken in the emergency response cases used in the experimental sessions described above. They are analogous to functional constraints in the FONM model [18]. A given function, F, may of course be undertaken for different goals.

- **Function has Object Group(s)**

  An object group, OG, is a prototype, or base-class, object. Most individual objects (i.e., material resources) in the ontology are instances of an object group, the exceptions being alternative resources, as described below. In the current implementation of the ontology, objects in an object group are equivalent instances of the object group; that is, all objects within an object group are regarded as equally capable of contributing to the execution of a function with which their base-class object group is associated. A function’s level of attainment is the degree to which it is executable. For example, the function of “Patient Treatment” has numerous object groups associated with it, including the object group “Treatment Center.” Certain resources may contribute to greater degrees than others to a function’s level of attainment.

- **Object Group has Object(s)**

  A number of objects are specified as members of each object group. Objects are physical entities about which decisions (literally, allocations of resources) can be made. In the experimental cases, objects are the resources listed on the computer screen (e.g., ambulances, ladder trucks and medical personnel). Since a course of action (denoted CAi) specified by the group involves objects, a single object is denoted CAij. Objects occupy the middle part of the ontology’s hierarchy (i.e., the same level as devices in FONM) [33]. Based on informal inspection of the prior session transcripts, middle-level elements of the ontology (i.e., objects) appear to be used more than upper- or lower-level elements during decision making. An object may be classified as a member of more than one object group. Alternative resources (i.e., the additional resources presented during phase two) are not directly associated with object groups. An instance of the object group “Treatment Center” is the object “Hospital.”

- **Object has Properties**

  The properties of an object are atomic structures in the ontology. They occupy the lowest level in the hierarchy [33]. Consistent with FONM, properties are used in finding possible resource substitutes for use in accomplishing a function when objects associated with an object group within a function are not available. All objects have at least one property.

**Table 1. Summary of ontology elements**

<table>
<thead>
<tr>
<th>Element</th>
<th>Symbol</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>G</td>
<td>Control Access to Incident Location</td>
</tr>
<tr>
<td>Function</td>
<td>F</td>
<td>Erect Barriers</td>
</tr>
<tr>
<td>Object Group</td>
<td>OG</td>
<td>Barrier</td>
</tr>
<tr>
<td>Object</td>
<td>CAij</td>
<td>Patrol Car</td>
</tr>
<tr>
<td>Properties</td>
<td></td>
<td>length, height</td>
</tr>
</tbody>
</table>

The foregoing ontology is operated upon by the logic described in the following section.

### 3.3. Decision Cycle and Logic Description

The purpose of the decision logic of EMPROV is to provide alternative solution components to members of the group. There are three main steps to the decision cycle: initialization, inference of referent and provision of recommendations to the coordinator, as summarized in the Table 2:
Table 2. Overview of decision cycle

1. Initialization
Participants are presented with problem definitions, including resources available and goals to be met. Participants develop courses of action and specify corresponding goals, all of which are communicated to the coordinator. Coordinator communicates courses of action and goals to system.

2. Inference of Referent
System employs logic, acting upon ontology, to develop recommended solution components.

3. Provision of Recommendations
Solution components are made available to coordinator for consideration by group. Participants specify revised courses of action and corresponding goal(s).

These steps are now considered in greater detail.

1. Initialization
During phase one of each case, group members develop recommended courses of action and the goals which they intend these courses of action to achieve. The group coordinator records courses of action; service representatives corroborate these decisions by authorizing the use of their resources. The ith course of action for a group is denoted CAi, where

\[ \{CA_{ij}, \ldots, CA_{in}\} \]

and i = 1, ..., n. The elements of CAi are the individual resources, listed in the order in which they are used. An example course of action is to send resource "b" (say, a police cruiser with driver) from site "C" directly to the incident location (site "Z") for the goal of controlling access to the incident location. The coordinator indicates the resource(s) to be used in a course of action, the sequence in which they are to be used and the goals the group intends to achieve in utilizing them. For the above example, CA1 = \{Cb, Z\}.

The screen printout in Figure 2 shows the interface used by the coordinator for entering courses of action. On the left side is a map showing the area over which resources are distributed and also the location of the incident. Sites where resources are located are labeled with capital letters; the icon adjacent to this label indicates which service operates the site (i.e., has control over its resources). To the right are boxes in which the coordinator enters courses of action and corresponding goals.

![Figure 2. Case one interface for coordinator](image)

Since there may be more than one goal which the group intends to accomplish with a given course of action, the goals associated with CAi are denoted as the elements of Gi, where

\[ \{G_{ij}, \ldots, G_{im}\} \]

The group’s specification of courses of action and corresponding goals completes the initialization phase. Courses of action and corresponding goals are stored by the system for processing during step two.

2. Inference of Referent
The next step is to infer the referent from the phase one courses of action and to make recommendations on resources which might serve as substitutes for those submitted in phase one. Two routines are associated with this step. The first routine, InferReferent, proceeds as follows. For each CAi, determine the contribution of CAij to each element of Gi. That is, determine the contribution towards the goals associated with each phase one courses of action. The contribution is computed as follows. For each goal Gij in Gi, determine the functions

\[ F_i = \{F_{ij}, \ldots, F_{ip}\} \]

associated with Gij. Next determine the object groups \( OG_{ij} = \{OG_{ij}, \ldots, OG_{ijq}\} \) with which CAij is associated, then for all \OG_{ij}, k = \{1, ..., r_i\}, repeat the GoalScore routine with all the object groups in \OG_{ij}, as follows.

- GoalScore Routine
If the number of elements in \OG_{ij} is greater than zero, then determine if OG_{ij} is used in accomplishing any function(s) in Fii in \( F_i \), where t = 1, ..., p. If so, then increment the attainment level for the function Fii with respect to Gij, the current goal. If not, then evaluate CAij as an alternative resource using the routine AREval.
• AREval Routine

Denote the current goal as $G_{ij}$. If a resource is passed to the routine, denote it $AR_k$ and set $AR = \{AR_1\}$. Otherwise, find all candidate alternative resources $AR = \{AR_1, \ldots, AR_s\}$ near $CA_3$ (i.e., those within a threshold distance from $CA_3$). Repeat the following sub-routine with all $AR_k \in AR$, $k = 1, \ldots, s$:

- Initialize with $F_k$, $Q_{ij}$ and $AR_k$.
- Determine $OG_{pi} = \{OG_{p1,i}, \ldots, OG_{pm,i}\}$, the set of all object groups associated with $F_k$.
- For all $OG_{p,m}$, $m = 1, \ldots, s$, determine the proportion of matching properties between $AR_k$ and $OG_{p,m}$. Store the value which is largest (this is the greatest potential contribution of $AR_k$ to $OG_{p,m}$ via the function $F_k$).
- Once all $AR_k$ have been evaluated over all elements of $OG_{p,i}$, compute the potential overall contribution of each $AR_k$ to $G_{ij}$ as the sum of the contributions to the individual functions in $G_{ij}$ divided by the total number of functions in $G_{ij}$. If the resulting value is positive or if $AR_k$ is a resource submitted by the group during phase one, then retain $AR_k$ in $AR$; otherwise, delete it.

3. Provision of Recommendations

Provide the coordinator with each alternative resource for each element of each $CA_i$ (i.e., with the various sets of $AR$ stored at the conclusion of step 2, “Inference of Referent”).

4. Discussion and Conclusions

Two concepts which emerge from the literature on cognition during improvisation are those of the referent and the realization. The respective terms denote a rough guide to creative behavior and its particular effectuation. The cognitive processes by which a realization emerges from a referent, however, are not well-understood. In creating improvising computer programs, researchers have speculated that hierarchical memory structures may be drawn upon in developing creative actions. To integrate this work with results from creativity support systems, the current research attempts to link human improvisation for the purpose of more seamlessly integrating human-computer collaboration on an improvisatory task. The second is integration of the decision logic with simulations of emergency response situations. Doing so should create a more realistic experience for participants and increase opportunities for gathering data on how creative decision making strategies evolve in the context of changes in system state.

Acknowledgements: This research has been supported through National Science Foundation Grant CMS-9872699.

5. References