Message classification as a basis for studying command and control communication: an evaluation of machine learning approaches

Ola Leifler and Henrik Eriksson

Ola Leifler and Henrik Eriksson, Message classification as a basis for studying command and control communication: an evaluation of machine learning approaches, 2011, Journal of Intelligent Information Systems.
http://dx.doi.org/10.1007/s10844-011-0156-5

Journal of Intelligent Information Systems manuscript No.
(will be inserted by the editor)

Message Classification as a basis for studying command and control communications - An evaluation of machine learning approaches

Ola Leifler · Henrik Eriksson

the date of receipt and acceptance should be inserted later

Abstract In military command and control, success relies on being able to perform key functions such as communicating intent. Most staff functions are carried out using standard means of text communication. Exactly how members of staff perform their duties, who they communicate with and how, and how they could perform better, is an area of active research. In command and control research, there is not yet a single model which explains all actions undertaken by members of staff well enough to prescribe a set of procedures for how to perform functions in command and control. In this context, we have studied whether automated classification approaches can be applied to textual communication to assist researchers who study command teams and analyze their actions.

Specifically, we report the results from evaluating machine leaning with respect to two metrics of classification performance: (1) the precision of finding a known transition between two activities in a work process, and (2) the precision of classifying messages similarly to human researchers that search for critical episodes in a workflow.

The results indicate that classification based on text only provides higher precision results with respect to both metrics when compared to other machine learning approaches, and that the precision of classifying messages using text-based classification in already classified datasets was approximately 50%. We present the implications that these results have for the design of support systems based on machine learning, and outline how to practically use text classification for analyzing team communications by demonstrating a specific prototype support tool for workflow analysis.

Keywords Command and control; classification; exploratory sequential data analysis; workflow mining; random indexing; text clustering

O. Leifler · H. Eriksson
Dept. of Computer and Information Science
Linköping University
SE-581 83 Linköping, Sweden
Tel. +46 13 281000
E-mail: {ola.leifler,henrik.eriksson}@liu.se

1 Introduction

Although successful command and control is essential to the success of crisis management and military operations, our understanding of how command and control is performed is still limited (Brehmer 2007). Studying command teams present commanders and researchers with great challenges. First, commanders need to accomodate shifting circumstances and uncertain information about the environment in their work process which makes the work process inherently dynamic (Klein et al. 1993). Second, the use of electronic communications and new media in command teams yields large amounts of data (e.g. text communications, audio, video, computer logs) that are difficult for researchers to process.

An important aspect of analyzing command and control is to find critical episodes in the workflow that warrant further study. Currently, most analyses of electronic communication in both situated and distributed teamwork are conducted manually through the use of *classification schemes* (Silverman 2006). A consequence of the significant effort required by manually classifying communications is that only a part of teams' communication patterns can be explored. The prospect of using automatic support for finding relations in command and control communications is therefore appealing.

This paper presents an evaluation of automated approaches for classifying text messages in the workflows of command and control teams by comparing a selection of classifiers with respect to their precision of classifying messages similarly to human experts. Our selection of classification approaches to compare was justified by the requirements of a widely used method for studying command and control, Exploratory Sequential Data Analysis (ESDA) (Sanderson and Fisher 1994). The results from our evaluation are twofold: first, we identify a classification approach which is suitable for use in an ESDA application, and second, based on the precision results attained, we outline how the classification approach could be used to support the study of command and control workflows.

In the following sections, we describe command and control research and the rationale for investigating machine learning approaches for supporting it in Section 2. Section 3 presents research on extracting patterns related to workflows and related concepts from texts. In Section 4 we present the specific data sets we have applied our classification approaches on. We present our classification approaches in Section 5 and the results of classifying messages in Section 6. Based on these results, we discuss their implications on the design of support tools for analyzing command and control communications and present an implementation that uses automatic classification of text messages in Section 7, and Section 8 concludes this paper.

2 Background

Command and control researchers investigate *how groups and group members perform their tasks*, identify *performance measures* for the group and study how they could *improve their performance* (Brehmer 2007). There are several frameworks for understanding teams and teamwork (e.g. (Argyle 1972; Salas et al. 2008)). A common representation of team workflows is to use graphs, where nodes represent tasks and arcs denote transitions between tasks. Such graph-based workflow models have been suggested for the analysis and support the coordination of work in various professional settings (Medina-Mora et al. 1992; van der Aalst and van Hee 2002).

One example of a workflow model that aims to describe how members of command teams perform their tasks is the Dynamic Observe-Orient-Decide-Act model in Figure 1.

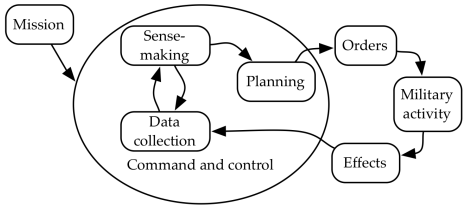

Fig. 1 The Dynamic Observe-Orient-Decide-Act loop by Brehmer as an abstract model of a workflow in command and control with tasks and transitions between them (Brehmer 2005).

DOODA describes a set of tasks with transitions from one task to another. These tasks can be overlapping or iterating, such as the tasks of *sensemaking* (Weick 1995) and data collection in DOODA. At one point, however, there is a transition from sensemaking to planning, when the commander's intent is formulated and communicated to subordinate units. Irrespective of whether this model accurately describes command and control at a sufficient level of detail for correlating the activities in the model to the observable activities in a command staff, the model could be used as a *hypothesis* for analyzing staff work. If we believe, according to a model such as DOODA, that the staff should begin with *data collection*, and we know that messages of certain types denote a transition to the *sensemaking* step in the DOODA process, then the absence or presence of such types of messages would be part of a researcher's work of establishing performance measures for a command staff.

In general, we can interpret the task of understanding command and control as three separate tasks. First, understanding *how command teams and team members perform their tasks* means constructing a general workflow model such as DOODA from command and control scenarios. Second, establishing direct *performance measures* is synonymous to relating the workflow model to the estimated outcome of scenarios as defined by indirect measurements of scenario outcome (for example, performance scores in computer simulations (Johansson et al. 2003) or evaluations by human experts of team performance in role-playing exercises (Jensen 2009)). Third, *improving performance* is equal to, in each particular scenario, using those performance measures to relate staff actions to the proposed workflow. The process by which researchers establish a workflow and relate staff actions to it from recorded scenario data is based on two principal activities: (1) labeling communication acts with a categorization scheme (Thorstensson et al. 2001), and (2) looking for higher level patterns of episodes (tasks) with the labelled communication acts to focus the search for critical points that have affected the outcome of the scenario (Sanderson and Fisher 1994; Albinsson et al. 2004).

The work of labeling messages according to message categories is the most time-consuming step, with vast quantities of communication data to sift through iteratively, first searching for commonalities that can lead to classification schemes, and later by applying classification schemes to all utterances and reducing the amount of data to a set of episodes based on the classification. This is also the activity for which we evaluate the use of machine learning techniques.

3 Related work

The problem of inferring activities from text-based communications has been studied previously by Kushmerick and Lau (Kushmerick and Lau 2005). Their approach was based on searching for specific syntactic patterns originating from the use of computer software (e-

commerce systems). Those patterns were in turn used to organize messages into workflows. Patterns originating from the use of computer systems has also been studied by the workflow management community (van der Aalst et al. 2003) where workflows have been elicited from interactions with workflow management systems or other software systems. Both these approaches concern the mining of machine-generated patterns, not patterns originating from human activities.

Regarding the recognition of human activities from text, Scerri et al. (2008) have proposed a model for human workflow management in a semantic desktop environment that relies on the detection or tagging of speech acts in e-mail. Their approach is based on Speech Act recognition performed by a speech act extraction web service which uses grammar patterns for detecting speech acts. Their stated application is to support individuals by monitoring unresolved issues in e-mail conversations such as unanswered questions. Other researchers have described an approach to workflow mining from unstructured data which relies on the existence of a fixed, known number of activity types or named entities in messages for determining which activity a message pertains to (Wen et al. 2009; Geng et al. 2009). Mainly, however, the problem of extracting patterns from e-mail has been studied for the purpose of filtering spam (Sahami et al. 1998) which is essentially equivalent to considering whether a message is at all related to any kind of activity the user is engaged in.

Several projects have attempted to elicit patterns of a domain-specific discourse, mainly from questions and responses sent between customers and company support lines for the purpose of helping customer support identify previous, relevant answers to new questions (e.g. (Larsson and Jönsson 2009; Chalamalla et al. 2008)).

In document management, researchers have studied approaches to relate specific domain knowledge in the form of concepts, objects and relations to textual documents (McDowell and Cafarella 2006; Eriksson 2007) and based on such semantic documents, some projects have studied how to create support for information management in team workflows by using domain-specific document features (Franz et al. 2007; Leifler and Eriksson 2009).

4 Material

We used three data sets to establish how well machine learning approaches would classify messages compared to human classification. Our data sets came from three command and control scenarios (Labeled *ALFA -05*, *C3Fire -05* and *LKS* from the projects they originate from) in which crisis management teams had used free text-based means of communication for coordinating their work (fending off forest fires in *ALFA -05* and *C3Fire -05*, and defending against information warfare in *LKS*). In all settings, the participants engaged in activities they were likely to encounter in their profession and the settings used had authentic chains of command and scenario descriptions. The tasks in each scenario were conducted as simulated exercises where the participants collaborated in teams to solve a task. Their performance had been assessed by the staff leading the exercises, which in all cases consisted of researchers studying team performances.

In Table 1 we list the attributes made available to the classifiers we studied. Some of the attributes were derived from other attributes and reflected what we believed was relevant for human classification of the messages in each dataset. The *Message direction* attribute is calculated using the algorithm in Figure 2, which implements the `compareTo` method available in Java and other programming languages.

Table 1 Non-text attributes used for message classification.

Attribute	Description	*ALFA -05*	*C3Fire -05*	*LKS*
Sender	Text	×	×	×
Recipient	Text	×	×	×
Sender level[1]	$\{0 \dots 4\}$, low values represent high rank in the organization	×	×	
Recipient level[1]	$\{0 \dots 4\}$, low values represent high rank in the organization	×	×	
Question marks present[1]	$\{true, false\}$	×	×	
Message direction[1]	$\{-1, 0, 1\}$, calculated as the normalized difference between the sender level and recipient level	×		
Message text	Text	×	×	×
Message time	Date	×	×	×
Message type	Nominal decision attribute	×	×	×

```
def get_message_direction(instance):
    direction = instance.sender_level - instance.recipient_level
    if direction > 0:
       return 1
    elif direction < 0:
       return -1
    else:
       return 0
```

Fig. 2 Algorithm for calculating the *Message direction* dataset attribute.

4.1 ALFA -05

The *ALFA -05* dataset consisted of 849 text messages exchanged between seven commanders in a simulated crisis response scenario (Trnka et al. 2006). During the scenario, commanders operated at three levels of command, in two administrative areas (approximately county-sized areas) and played a role-playing simulation exercise (Trnka and Jenvald 2006) in which there was initially a forest fire but subsequently also an evacuation from a zoo as well as a search and rescue operation. Participants communicated with one another through a text-based messaging system designed for use in micro-world simulations with the C^{3}Fire simulation environment (Johansson et al. 2003), although it shared the basic features of an e-mail messaging system without the use of subject lines or other auxiliary e-mail headers. The scenario was played over the course of one day.

Each message had been assigned one of 19 different classes by hand. These classes fall into four speech-act-related categories tied to the functions of command and control (Trnka et al. 2006). The four categories were *questions*, *information*, *commands* and *other messages* (the *Message type* in Table 1). When researchers had looked for patterns in the ALFA -05 dataset, they had studied both the general proportions of messages of each class sent to and from the participants in the scenario, but they had also studied specific sequences of speech acts, such whether as a set of *information* and *question*-labelled message exchanges had preceded a *command*.

4.2 C3Fire -05

The C3Fire -05 dataset was similar to ALFA -05 with regard to the scenario played and the categorization used. It consisted of 619 messages. One of the main differences was that it was categorized by two independent researchers with a 77.86% agreement between the two on which category to assign each message (the agreement was 87.02% when considering only the four main categories described in the section above). Only those messages which had been classified similarly by the two researchers were selected for classifier comparison. The other main difference compared to ALFA -05 messages was the participants of the study, who were domain experts in the ALFA -05 scenario and students in the C3Fire -05 scenario.

4.3 LKS

The *LKS* dataset consisted primarily of 113 e-mail messages exchanged during a training exercise concerning information warfare at the Swedish Defense Research Institute. All participants were experts in the domain and the exercise served the dual purpose being of exercise for them as well as a study of performance indicators in command and control. The scenario was role-played over the course of two days and the participants received instructions from their higher command to engage in intelligence operations for the first day to find information about, locate, and monitor potential terrorists, and repel threats during an evacuation of a VIP during the second day of their operation.

Due to these instructions, we categorized the e-mail exchanges pertaining to the first day as *intelligence* and those from the second as *evacuation*, which was consistent with the expected outcome of the exercise. The manual classifications of both datasets were used as validation of the automatic classification approaches we report in this paper.

5 Method

To verify that the information in messages could be used for distinguishing contextually significant classes of messages from one another[1] consistent with how command and control researchers would classify messages, we added meta-data to our datasets that we believed to be relevant to classification. With these datasets, we conducted a comparison between several classification approaches by using standard methods for evaluating Machine Learning algorithms.

Messages in a military command and control workflow usually contain domain-specific attributes such as the rank and role of participants. Also, researchers may classify according to the appearance of question marks and the grammatical structure of messages. To understand how these attributes affect automated classification, we compared the impact on classification results of encoding these attributes as part of the message instances. The appearance of question marks became a binary attribute available to non-text classifiers while the grammatical structure was made available to a String Subsequence Kernel-based classifier (see Section 5.2). We also evaluated the relative significance of non-text attributes in relation to the text by using a combined classifier that would use a text-based classifier and a non-text classifier in combination for classification. The combined classifier would

[1] such as identifying the two tasks in the LKS dataset or the message classes related to speech acts in the ALFA -05 and C3Fire -05 datasets

Table 2 Frequency of messages in each of the four message categories of the ALFA -05 dataset.

Category	Proportion
Questions	23%
Information	39%
Orders	17%
Other messages	24%

also provide information on the relative contributions of a non-text classifier compared to a text-based one.

Apart from domain-specific message attributes which are likely to influence human classifications of messages, we considered the influence of a numerical attribute with statistically significant differences of attribute values across the categories of messages: message length. To establish whether a significant difference in message lengths would be used by a classifier when building a classifier model, we studied whether a standard discretization approach (Fayyad and Irani 1992) (as required by the classifiers we evaluated) would generate meaningful nominal interval values and if so, what precision results the classifiers would attain.

We also considered the precision of a random classifier and used that as a baseline for comparing the results of using our selected classification algorithms. If our classifier would not find a meaningful distance measure for the purpose of classifying with respect to message categories (in the ALFA -05 dataset) or belonging to different stages in the scenario workflow (in the LKS dataset), the classifier would basically choose a class at random. The precision it could attain for each decision class could then be described as a function of the proportion of instances of each decision class in the training data.

The precision of the algorithm is expressed as the number of times the algorithm answers correctly, divided by the total number of questions asked. Thus, it is the sum of the number of correct classifications with respect to each of the classes. A completely random classifier, given a dataset U and a function d for mapping messages to the domain of decision classes $\{c_1, c_2, \ldots, c_l\}$ where the sizes of each class is $|c_i| = |\{x \in U : d(x) = c_i\}|$ would attain precision of Equation 1.

$$\Sigma_{i=1}^{l} \left(\frac{|c_i|}{|U|} \right)^2 \tag{1}$$

The LKS dataset consisted of two classes, evenly distributed with 61 messages from day one and 52 from day 2. The random precision would be $(61/113)^2 + (52/113)^2 = 0.5032$, close to 50%. Given the distribution of decision classes in Table 2, random precision attainable in the ALFA -05 dataset was $0.23^2 + 0.39^2 + 0.17^2 + 0.24^2 = 0.29$. Classification results of approximately 29% in ALFA -05 would therefore be attributed to the distribution of messages and not to the message contents. For the C3Fire -05 dataset, the distributions of classes was more even for both sets of classifications from the two researchers, resulting in random precision of 24.04% and 25.07% respectively.

When evaluating the different approaches to classify messages, we used a stratified cross-validation (Witten and Frank 2005) on each dataset. To accomodate the execution times of text-based classification, we decided to use a 3-times 3-fold stratified cross-validation on our datasets for evaluation. The results were stable when confirmed with a train-and-test procedure on each dataset.

Table 3 Message lengths in all categories

	Information	**Commands**	**Questions**	**Other**
Mean	110 ± 78.94	76 ± 93.18	90 ± 68.56	55 ± 60.18
Median	92	58	68	32

5.1 Message lengths

The messages from the four main categories of the ALFA -05 dataset were compared with one another with respect to the lengths of the messages in each category. Since the different message categories contained a different number of messages and the message lengths could not be assumed to be normally distributed, we compared the differences with a non-parametric Mann-Whitney U-test. All categories of messages were compared to one another pairwise. All pairs of categories displayed significant differences in message lengths ($p < 0.002$) and the mean and median values differed as outlined in Table 3, along with standard deviations from the means.

5.2 Classifier selection

The classification schemes we used for both text classification and non-text classification on our datasets were selected based on two primary criteria:

1. the models built as part of learning patterns in data should be *accessible to human inspection*, and
2. they should be computationally *tractable for interactive use* in both scenarios.

The first criterion, accessibility, was considered important because of the prospect of using the resulting classifier model as a basis for a support tool for command and control researchers. In ESDA analysis, exploration means using various data sources in combination to detect patterns of team activity. For a computer-based support tool in this process, establishing trust is critical, and understanding the basis for making classifications could even be more important than high precision for classification, depending on the role of a classifier. The second criterion, computational tractability for interactive use, was considered important for the practical use of automatic classification. In data exploration tools such as MacSHAPA (Sanderson et al. 1994) and MIND (Thorstensson et al. 2001), researchers navigate scenario data looking for critical episodes by scanning a timeline according to which all scenario data is logged to find incidents that are important for further study. When using such tools, researchers expect interaction with data to be smooth and allow fast manipulations due to the labor-intensive task of finding critical episodes. For an automatic classifier to contribute in such exploration, it would have to build a classifier model fast enough not to interrupt the closer study of data.

Based on these criteria, we selected a text-based classification scheme that would connect important terms as well as the relationships between terms during the process of classification, with the intention of using those terms as part of a workflow analysis tool. Also, it would have to handle the datasets we had with little computational overhead. Based on these criteria, we decided to use the Random Indexing (RI) (Kanerva et al. 2000) vector space model as the primary method of text classification. RI assigns random vectors of a fixed dimensionality to words and texts to create the vector model for measuring similarity between texts (Kanerva et al. 2000). Prior to building the RI model, we filtered the messages

```
Questionmark present = true: 1
Questionmark present = false
|   Sender = RL Nkpng: 2
|   Sender = LKC E-ln
|   |   Recipient_level <= 3
|   |   |   Time <= 1133433043000
|   |   |   |   Time <= 1133432780000: 2
|   |   |   |   Time > 1133432780000: 3
|   |   |   Time > 1133433043000
|   |   |   |   Recipient = Ambu E-ln: 2
|   |   |   |   Recipient = Patruller E-ln: 2
|   |   |   |   Recipient = RL Nkpng: 2
|   |   |   |   Recipient = LKC D-ln
|   |   |   |   |   Time <= 1133436580000: 4
|   |   |   |   |   Time > 1133436580000: 2
```

Fig. 3 Part of the decision tree generated by the J48 classifier on the ALFA -05 dataset

so that commonly used, domain-independent words (stop words) would not taint our results. In addition to the RI-based text classification method, a String Subsequence Kernel was also used for analyzing the grammatical structure of messages (see Section 5.4) and for comparison of the RI text classification results. For non-text classification, we used four different classifiers, representing four classes of inference mechanisms:

1. J48, a classifier based on decision-trees (Quinlan 1993)
2. a Decision Table classifier (Kohavi 1995)
3. PART, a rule-based classifier (Frank and Witten 1998)
4. a Nave Bayes classifier (John and Langley 1995)

The first three classifiers were selected based on the accessibility of the models they construct, and the fourth, the Bayesian classifier, was selected due to previously reported results on classifying messages with respect to workflow-related activity types (Geng et al. 2009) with a Bayesian classifier. We conducted the evaluation within the WEKA knowledge analysis framework (Hall et al. 2009), within which we also implemented an RI-based text classifier.

In deserves to be noted that, due to the relatively small datasets available for training the classifiers, we assumed that there would be differences in precision which could be attributed to classifier selection, apart from the differences in what models they build. For larger datasets, it has been argued that classifier selection may become less important (Banko and Brill 2001), which is why the issue of evaluating classifiers may make more sense with smaller datasets.

When studying how accessible the models produced by the classifiers were, we tried to elicit the heuristics that the classifier models expressed in order to establish whether they were sound compared to how human experts would reason. The decision tree in Figure 3 shows, in ASCII format, a number of decision branches where the shortest branch indicates that the presence of a question mark should classify the message as being a question (Message type 1 in the ALFA -05 dataset). Then, there are a number of conditions in the tree which correspond to combinations of sender, the organizational level of the recipient and time, which can be explained by the change in interactions between the command center and the field units during the scenario. Early in the scenario, most exchanges concerned information exchanges (Message type 2), whereas later exchanges, initiated by higher command, concerned orders (Message type 3).

5.3 Dataset features

The ALFA -05 and C3Fire -05 datasets differed in several aspects from the LKS dataset:

- All messages had been categorized by hand according to a scheme with 19 speech-act-related categories,
- there were many more participants,
- there were many more texts in both ALFA -05 and C3Fire -05 compared to the LKS scenario, and
- each participant in the ALFA -05 scenario belonged to a certain position in an organization.

We knew that, when analyzing the ALFA -05 scenario by hand to find critical transitions in the workflow, researchers had made use of meta-information that was not encoded directly in the messages. Therefore, we decided to add 4 such additional features for better non-text classifier performance:

1. The *direction in the chain of command* was the first feature we added, based on the conjecture that the correct classification of a message (such as it being a command or information) would be related to the roles of those involved in the communication. Commands usually travel downwards in the chain of command whereas information usually flows up, from ground units to their superior officers. We therefore encoded the direction in the chain of command as a specific message attribute.
2. The *rank of the sender and recipient* of messages was added as an absolute value in contrast to the relative value of direction in the chain of command (direction being equal to the difference in rank between sender and recipient). The ranks were encoded as nominal values.
3. The *occurrence of question marks* was added to the attributes of messages, with the conjecture that messages with question marks would be labelled *questions* more often than messages of other classes. This was a nominal, binary feature, indicating whether 1 or more question marks occurred in each message.
4. We reasoned that *the phrase structure* of the messages might be related to the manual classification, so that a message classified as a question would display a different sequence of phrase structure tokens than a message classified as an order. The phrase structure of a message was encoded as a substitute for the message text and evaluated using a string-based classifier.

5.4 Phrase structure classification

Typically, questions have one particular grammatical structure whereas orders have another. Therefore, as an alternative to using the message text in itself, we extracted the phrase structure of each message in the ALFA -05 dataset and replaced the message text with its (shallow) grammatical structure, so that the message would consist not of a sequence of words but rather of a sequence of phrase grammar indices. The sequence of indices was treated as text that was classified with a kernel-based classifier using a String Subsequence Kernel (SSK) as its kernel function (Lodhi et al. 2002).

Although SSKs are computationally expensive, the texts that we subjected it to were relatively small. We therefore decided to investigate whether an SSK-based approach would attain reasonable results with respect to execution time.

```
tack ↦ nn.neu.sin.ind.nom
5914 ↦ rg.utr/neu.plu.ind/def.nom
nn.neu.sin.ind.nom ↦ 1
rg.utr/neu.plu.ind/def.nom ↦ 0
```

Fig. 4 An example of substitution of phrase structure for message text.

Table 4 Mean classifier precision results from a 3-by-3 stratified cross-validation on the ALFA -05 datasets

	J48	**Decision Tables**	**PART**	**Nave Bayes**
Precision (%)	**49,15**	49,00	47,11	48,37

A kernel-based classifier maps a document to a higher-order vector space just as RI. The difference compared to the RI method of reducing the vector space to a more computationally manageable size is that an SSK uses a set of sub-sequences of the text as the feature space that it maps documents into and uses for comparison. A defining feature of an SSK is that it treats strings (documents) as more similar if substrings occur in the same order in both strings. This feature makes it more suitable than RI for comparing whether the grammatical structure of two documents is similar, or phrased differently, if sequences of grammar tokens come in the same order in two texts. The formal alphabet used for an SSK should represent the words available in the *dictionary*, where the dictionary is a set of all distinguishable tokens in the input. When comparing the grammatical structure of texts, we therefore used a dictionary of phrase structure parts. To extract them, we mapped words to grammar parts and then to simple indices that simplifies the work of the SSK. Figure 4 shows how we mapped a simple acknowledgement message "tack 5914" ("thank you 5914") to a sequence of phrase grammar tokens and subsequently to indices. Ideally, each phrase grammar part should represent a single, unique letter in an alphabet to maximize kernel performance. When used with an SSK-based kernel classifier, we considered the use of indices in a phrase grammar vector as an appropriate approximation that would preserve the discriminating features of a phrase grammar structure.

6 Results

We began our classification evaluation by investigating the relative importance of text-based to non-text-based classification when combined in a meta-classifier. The meta-classifier used assigned a weight to each as an indication of the precision of each classifier during training.

6.1 Non-text classifier comparison

The four non-text classifiers presented in Section 5.2 were evaluated with respect to classification precision when tested against the man-made categorizations in the ALFA -05 scenario using stratified cross-validation. Table 4 presents the results from applying the classifiers on ALFA -05 dataset. All classifiers performed similarly on the data set and had access to all attributes listed in Table 1.

6.2 Text-based classifier comparison

We compared the Random Indexing-based classifier to the SSK classifier on both the ALFA -05 and C3Fire -05 datasets, as shown in Table 5. The results were inconclusive, as the

Table 5 Mean classifier precision results in percent from a 3-by-3 stratified cross-validation on the ALFA -05 and C3Fire -05 datasets

	RI	**SSK**
ALFA -05	48,96	**58,85**
C3Fire -05	**45,67**	40,25

Table 6 Relative importance of text-based and non-text-based classifiers when determining the class of messages in the ALFA -05 workflow as factors used in the linear regression model of the Stacking meta-classifier.

Message category	**Text**	**Non-text**
Questions	0.9204	-0.0239
Information	0.9224	0.0037
Orders	0.9304	-0.0119
Other messages	0.9299	0.0937

RI classifier outperformed the SSK on the C3Fire -05 dataset, whereas the SSK classifier performed better on the ALFA -05 dataset. However, the execution time of the SSK classifier was prohibitively high with the ALFA -05 dataset, requiring several hours to build a classification model.

6.3 Comparison of text-based and non-text-based classification

As text-based classification cannot readily be combined with non-text classification, we wished to establish the relative precision of text-based compared to non-text-based classification. To this end, we studied the linear regression model built by a Stacking meta-classifier (Seewald 2003) combining both types of classifiers, which gave us indications that text was the most important feature for classification. In Table 6, we see the relative weights attached to each classifier by the Stacking algorithm when classifying messages of the ALFA -05 dataset as belonging to one of the four categories. The weights signify the relative performance of the algorithms during the training phase of the evaluation. The text-based classifier had much higher precision than the non-text-based one and therefore contributed to a much larger degree to the overall predictions made by the meta-classifier.

Furthermore, we conducted a 3x3 stratified cross-validation of four combined Stacking classifiers which differed with respect to the non-text classifier used. The non-text classifiers in Section 6.1 were combined with an RI-based text classifier and evaluated with respect to classification precision on the ALFA -05 dataset. All four approaches showed similar results (49.20% precision, $\sigma = 3.17$), which indicates that the text-based classifier, common to all four approaches, determined outcome of the combined classifier, as suggested by the regression model in Table 6. Having established that the precision results of a combined classifier on the ALFA -05 dataset did not depend on the non-text classifier, we decided not to explore more option for non-text-based classification. Instead, we evaluated two more options for text-based classification. The results in Section 6.2 were inconclusive, but indicated that the SSK approach was able to provide precision results above the RI-based classifier on some data sets. However, the computational requirements of SSK were prohibitive. As described in Section 5.4, mapping message words to their phrase grammar tokens would make message texts smaller, and therefore possibly computationally more tractable for the SSK approach, while preserving the grammatical structure of the texts.

Table 7 Prediction precision results from a set of classifier evaluations under different conditions. In each condition, a combined classifier was evaluated on the ALFA -05 dataset with messages tagged as belonging to one of the four different categories.

Condition	Precision
combined classifier	51%
phrase structure classification	54%

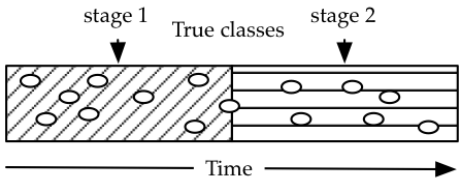

Fig. 5 The simplest workflow consisting of two stages separated in time. Ovals represents messages along a timeline.

Table 7 presents a set of results from a train-and-test evaluation of classifier performance. 90% of all messages, randomly selected with representations of all message categories, were used as training messages, and the remaining 10% as tests of classifier accuracy. The SSK classifier used in phrase structure classification continued to have a prohibitively high computation time[2] which would effectively prevent it from being a viable option for classifying messages in an online workflow support system irrespective of precision. It was also the computation time of the SSK classifier that restricted the comparison method to train-and-test as compared to a stratified cross-validation. The combined classifier showed precision results that were within one standard deviation from those obtained in the comparison of the two text-based classifiers.

6.4 LKS classification

The LKS dataset consisted of two days' worth of e-mail messages divided in two workflow phases: one for day one and one for day two. 61 messages were sent the first day of the exercise and 52 the second, yielding a total of 113 messages. Our evaluation of classification on this dataset was performed to establish that classifier performance was dependent on the domain-significant "accuracy" of our own division of messages between the two phases, so that a cut between the decision classes *stage 1* and *stage 2* that was not consistent with the real division of messages would yield a comparatively worse predictive performance compared to a more accurate division. If the classification results were noticeably better when dividing the messages at the point in time when the LKS participants got a new task, the classifier would probably pick up on domain-significant features in the message set.

Figure 5 describes a transition in a workflow, where one stage (task) leads to another. Messages in each stage come from one or several actors and are supposed to be associated to only one stage per message. In this trivial workflow, a transition involves the activity of

[2] Approximately 24 hours for finishing a single train-and-test evaluation procedure on the 849 messages in the ALFA -05 dataset.

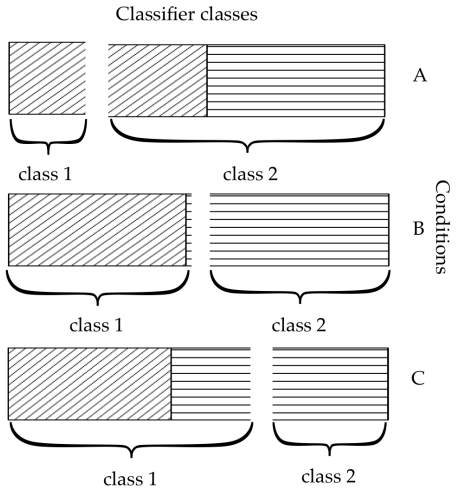

Fig. 6 Examples of conditions for evaluating the performance of a classifier model with respect to the precision of classifying messages. The "true" workflow stage division is in the middle.

all those involved in the first stage. This representation makes it possible for us to investigate whether a workflow transition would be possible to identify in the simplest possible workflow with a transition: two sequential stages.

To evaluate classification for this purpose, we cut the set of messages in two parts to investigate whether there would be any single point in the message flow where the classifier could perform comparatively better. Specifically, the message flow was divided between *stage 1* and *stage 2* after 20, 30, ...90 messages which generated a set of conditions under which a classifier was evaluated (Figure 6) and for each of these conditions, standard cross-validation procedures were applied for evaluating classifier prediction accuracy in comparison to random classification. Since the performance of a classifier may depend on the number of messages in each decision class as noted above, we compared classification results in each condition to those expected from random classification. Our conjecture was that conditions similar to condition B in Figure 6 would yield the highest relative precision. Condition B describes the case in which classes provided to the classifier for training are most similar to the true workflow stages (Figure 5). If a learning algorithm were only to achieve precision on par with random classification, then the classification would not be a function of the contents of the messages but merely a reflection of the proportions of messages in each decision class.

Figure 7 shows the results of using a combined classifier for determining which of the two stages in the workflow a message belongs to compared to random classification. The diagram shows the precision of classifying messages in the LKS dataset in two categories as a function of the number of messages in the first category (*stage 1*, intelligence). The results in Figure 7 indicate the strongest relative classifier performance compared to random prediction at about 60 messages. 61 messages were sent during the first stage (the first day) and 52 the second, which gave us the indication that the text-based classifier did achieve the best relative performance at the expected point in the workflow.

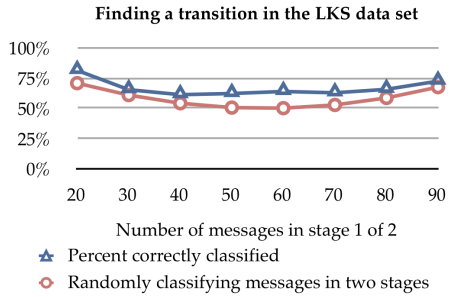

Fig. 7 The difference in precision of a combined classifier when predicting the workflow stage of messages in the LKS dataset compared to a random classifier.

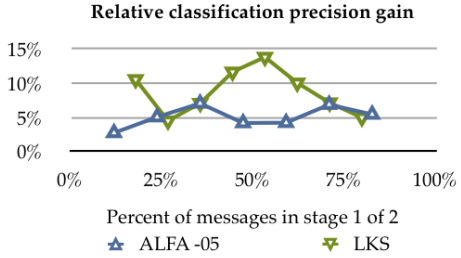

Fig. 8 The precision of a combined classifier when predicting the workflow stages of messages in the ALFA -05 and LKS datasets minus the results of random classification.

6.5 Summary of results

Our classification of command and control text messages was conducted to establish whether man-made classification used attributes that would be significant for automated classification. Our approaches to classify messages yielded these results:

- When comparing non-text classifiers against one another on the dataset that was most rich in non-textual metadata, there was little difference between a rule-based classifier, a decision tree classifier, a Bayesian classifier and a decision-table classifier.
- When comparing two text-based classifiers on the two similar data sets ALFA -05 and C3Fire -05, the results were inconclusive regarding precision. However, a noticeable difference was that one of the approaches (the SSK-based classifier) displayed a prohibitively high computation time.
- A combined classifier which used both the message text as well as other attributes of messages for predicting a workflow transition would almost exclusively use the classifications predicted from the message text, not from other attributes.
- All classification approaches tested on the ALFA -05 dataset yielded similar classification precision results of approximately 50%.
- The LKS classification achieved the highest relative gain compared to random classification at the point in the LKS message flow when participants were expected to move from the first stage of the operation to the second (see Figure 8). This seemed to indicate that the RI-based text classifier would find relevant transitions.

In Figure 8 we summarize the results of predicting which class a message belongs to in a workflow stage. The graph shows precision gains over random classification as a function of the number of messages allocated to the first stage out of two in the respective datasets. For example, at 50 percent the graph shows the relative improvement over random classification when detecting a workflow transition if dividing the datasets evenly in two stages. In the case of the LKS dataset, the highest relative gain over a random classifier was attained at the most even split in two stages which also coincided with the manual, "true" classification. The difference in precision was approximately 15 percent. For comparison, we include the results of classifying the ALFA -05 dataset as if it contained two linearly separable workflow stages, which we had no reason to believe it did.

7 Discussion of results

The LKS dataset consisted of two workflow stages that could successfully be identified using text-based classification.

In a support tool for analyzing command and control communications data using machine learning techniques, a text-based classifier would be able to tell two classes of messages apart in the same manner as a human observer (in the LKS case), but only be able to classify approximately 45-50% of messages similarly to human observers in the ALFA -05 and C3Fire -05 datasets, even though the C3Fire -05 dataset had been sanitized to only contain messages two researchers had categorized similarly.

Classification of the ALFA -05 dataset with respect to message categories yielded precision results of approximately 50 % compared to the expected random precision of 29 %, which indicate that the classification did extract useful information from the message flow with respect to the four categories. The String Kernel-based classification of phrase structure sequences yielded the best precision of all four approaches, but demonstrated a prohibitively high execution time for text classification of even moderately sized corpora, which had previously also been noted by Lodhi et al. (2002). The text classification results were better overall than the non-text results, and in particular, when combined using a meta-classifier, the text-based classification showed higher precision results than the non-text-based classifiers. Taken together, text-based classification would therefore seem to be an appropriate candidate for applying

7.1 Implications for support systems

The results from searching for a transition in the LKS dataset and classifying message categories in the ALFA -05 and C3Fire -05 datasets gave indications of the classifier precision for predicting which message classes. The precision was not sufficient for supporting command and control researchers with automation of message classification, but could yield insights on how a set of messages can be divided in clusters that are of importance in the domain. Having established that the precision possible with machine learning approaches is approximately 50% with these datasets, we have some limitations on the possibilities for automatically using classification in the ESDA workflow of command and control research.

However, precision results alone are insufficient for determining the utility of automatic classification as a basis for supporting researchers in the analysis of command and control communications. Although the precision results were low, they were compared against human evaluations, which may have been made according to domain knowledge not encoded

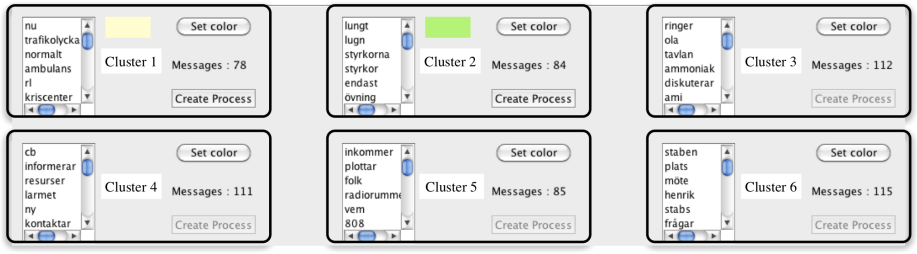

Fig. 9 The Workflow Visualizer tool uses a text-based classifier to support the analysis of messages in a workflow. Here, a number of possible clusters of messages are identified by the key terms occurring in them. Users can assign colors to messages in a cluster and plot them along a timeline for closer inspection.

properly for machine classification. In exploratory sequential data analysis, there are no predefined quality measures of classification to apply. In fact, categories for labeling messages may be developed as part of the analysis itself, and two independent analysts may categorize the same set of messages differently with the same classification scheme as demonstrated in the C3Fire -05 dataset (see Section 4.2). However, exposing the classifier model of a classifier which has been trained on one dataset to categorize messages from another could possibly provide a valuable support system for exploring possible patterns in command and control communications. When devising such a support system, the issue of how to make the classifier model usable is likely to be the most challenging.

7.2 Support system requirements

In the analysis process of ESDA, the requirements for transparency and traceability (Albinsson et al. 2004; Thorstensson et al. 2001) present a challenge for automatic message classification since the classifier must maintain a clear trace between individual messages and the model created of how they relate to one another. To support workflow analysis through classification therefore require us to use the classifier model to primarily *highlight possible relations* among messages during exploration, and require that there are *several options* for how to classify (with respect to message categories, transitions or other workflow-related features). We name these two requirements *transparency* and *graceful regulation* (Leifler 2008).

Transparency represents the degree to which a computer-generated model of a dataset, such as a vector-based model of the texts in a communication flow, can be related directly to the underlying data sources the model is based on. Transparency can be achieved by exposing the defining features of the model through a graphical interface where the connection between raw data and computer model is as simple as possible to understand. In the case of a vector-space model of words and texts, this translates into making the terms extracted by text classification part of the interface for selecting and inspecting messages, and part of the description of message clusters. Rules or decision trees extracted by a non-text classifier could be made part of a selection interface in a similar manner.

Graceful regulation can be understood the ability to choose different uses of a computer-based model depending on how the user trusts the model and what the user needs. In the case of communication analysis, this translates into using message classification for two distinct purposes: selecting and inspecting parts of the communication flow based on specific key terms or selecting and inspecting clusters of messages according to the model. The

former method only requires the user to rely on the computer model to present the most frequent terms, whereas the latter requires the user to trust the vector-based model to produce contextually significant clusters of messages. We have implemented these two requirements in a prototype support tool for exploring relations in communications.

7.3 Workflow Visualizer

Our implementation of a support tool for exploring communication patterns is called Workflow Visualizer and consists of components for selecting and visualizing messages that are part of a structured workflow in command and control. Figure 9 shows a view of Workflow Visualizer, in which an RI-based model has been constructed from a series of crisis management scenarios. The evaluations reported in this paper indicate the contextual validity of the RI approach. Based on these evaluations, we used the vector-based RI representation of message texts for creating message clusters to present *possible* patterns in data. Such patterns can be used by command and control researchers who study C^2 teams and trace communication trails for information on how certain concepts have been communicated and understood and how the sensemaking process of the team has worked.

The message clusters were extracted from written communications and text logs in a command and control scenario and are represented in the graphical interface by the most significant terms in each cluster. The clustering shown in Figure 9 comes from a set of 10 similar command and control scenarios studied previously by researchers[3]. During analysis, the researchers were interested in finding if there had been deviations from standard communication patterns in any of the scenarios which could indicate stress or fatigue. One way of studying this is to explore the automatic clusters generated by the Workflow Visualizer. For example, cluster 2 identifies messages that relate to low workload (key terms being “lugn” (calm) and “lugnt” (calmly)). Those messages are not evenly distributed across all 10 scenarios and could give insights into whether the staff had had different experiences during the exercises. In future work, we will evaluate the how command and control researchers will make use of the Workflow Visualizer when used with authentic scenarios to help answer realistic research questions.

8 Conclusions

We have established that, given a known transition in a multi-actor workflow manifested in written communication, Random Indexing-based text classification is able to successfully detect the transition through a series of classification trials. We have also established a baseline for the precision attainable when using both text-based and non-text-based classification for identifying classes of messages that are relevant for helping researchers identify transitions in a command and control team workflow. Based on the precision results and a discussion of how to support command and control researchers, we have described two general requirements, transparency and graceful regulation, for tool support in command and control research and presented a prototype tool for supporting C^2 researchers find workflow-related patterns in communications.

The most time-consuming work in the analysis of command team behavior is the selection and filtering of data from scenarios and in particular communication data. In the

[3] These scenarios contained uncategorized data and were therefore not part of the evaluations reported in this paper

study of structured team work environments such as command and control, we argue that automatic text clustering offers a viable technological basis for interactive exploration and analysis that offers concrete advantages for understanding of how groups of people work.

9 Acknowledgments

This work was supported by the Swedish National Defense College. The LKS dataset was contributed by the Swedish Defense Research Agency (FOI) and the ALFA -05 dataset was generously shared by Jiri Trnka, now at FOI.

References

Albinsson, P.-A., Morin, M., and Thorstensson, M. (2004). Managing metadata in collaborative command and control analysis. In *Proceedings of the 48th Annual Meeting of the Human Factors and Ergonomics Society*.

Argyle, M. (1972). *The Social Psychology of Work*. The Penguin Press, London, UK.

Banko, M. and Brill, E. (2001). Mitigating the paucity-of-data problem: Exploring the effect of training corpus size on classifier performance for natural language processing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Brehmer, B. (2005). The Dynamic OODA Loop: Amalgamating Boyd's OODA Loop and the Cybernetic Approach to Command and Control. In *Proceedings of the 2005 Command and Control Research and Technology Symposium*.

Brehmer, B. (2007). Understanding the functions of C^2 is key to progress. *The International C^2 Journal*, 1(1):211–232.

Chalamalla, A., Negi, S., Subramaniam, L. V., and Ramakrishnan, G. (2008). Identification of class specific discourse patterns. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1193–1202. ACM.

Eriksson, H. (2007). The semantic document approach to combining documents and ontologies. *International Journal of Human–Computer Studies*, 65(7):624–639.

Fayyad, U. M. and Irani, K. B. (1992). On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8:87–102.

Frank, E. and Witten, I. H. (1998). Generating accurate rule sets without global optimization. In Shavlik, J., editor, *Fifteenth International Conference on Machine Learning*, pages 144–151. Morgan Kaufmann.

Franz, T., Staab, S., and Arndt, R. (2007). The X-COSIM integration framework for a seamless semantic desktop. In *K-CAP '07: Proceedings of the 4th International Conference on Knowledge Capture*, pages 143–150. ACM.

Geng, L., Buffett, S., Hamilton, B., Wang, X., Korba, L., Liu, H., and Wang, Y. (2009). Discovering structured event logs from unstructured audit trails for workflow mining. In Rauch, J., Ras, Z., Berka, P., and Elomaa, T., editors, *Foundations of Intelligent Systems*, volume 5722 of *Lecture Notes in Computer Science*, pages 442–452. Springer Berlin / Heidelberg.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explorations*, 11(1).

Jensen, E. (2009). Sensemaking in military planning: a methodological study of command teams. *Cognition, Technology & Work*, 11:103–118.

Johansson, B., Persson, M., Granlund, R., and Mattsson, P. (2003). C3fire in command and control research. *Cognition, Technology & Work*, 5(3):191–196.

John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufmann.

Kanerva, P., Kristoferson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*.

Klein, G. A., Orasanu, J., Calderwood, R., and Zsambok, C. E., editors (1993). *Decision Making in Action: Models and Methods*. Ablex Publishing corporation.

Kohavi, R. (1995). The power of decision tables. In *Proceedings of the 8th European Conference on Machine Learning*, pages 174–189. Springer.

Kushmerick, N. and Lau, T. (2005). Automated email activity management: An unsupervised learning approach. In *Proceedings of the Conference on Intelligent User Interfaces*.

Larsson, P. and Jönsson, A. (2009). Automatic handling of frequently asked questions using latent semantic analysis. In *Proceedings of the IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

Leifler, O. (2008). Combining Technical and Human-Centered Strategies for Decision Support in Command and Control — The ComPlan Approach. In *Proceedings of the 5th International Conference on Information Systems for Crisis Response and Management*.

Leifler, O. and Eriksson, H. (2009). Domain-specific knowledge management in a semantic desktop. In *Proceedings of I-KNOW '09, The International Conference on Knowledge Management*.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. J. C. H. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444.

McDowell, L. K. and Cafarella, M. (2006). Ontology-driven information extraction with ontosyphon. In *Proceedings of the 5th Internation Semantic Web Conference*.

Medina-Mora, R., Winograd, T., Flores, R., and Flores, F. (1992). The action workflow approach to workflow management technology. In *CSCW '92: Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 281–288. ACM.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.

Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E. (1998). A bayesian approach to filtering junk e-mail. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*.

Salas, E., Cooke, N. J., and Rosen, M. A. (2008). On teams, teamwork, and team performance: Discoveries and developments. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(3):540–547.

Sanderson, P. and Fisher, C. (1994). Exploratory sequential data analysis: Foundations. *Human-Computer Interaction*, 9:251–317.

Sanderson, P., Scott, J., Johnston, T., Mainzer, J., Watanabe, L., and James, J. (1994). MacSHAPA and the enterprise of exploratory sequential data analysis (ESDA). *International Journal of Human-Computer Studies*, 41(5):633–681.

Scerri, S., Handschuh, S., and Decker, S. (2008). Semantic email as a communication medium for the social semantic desktop. In *The Semantic Web: Research and Applications*. Springer Berlin/Heidelberg.

Seewald, A. K. (2003). Towards a theoretical framework for ensemble classification. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*.

Silverman, D. (2006). *Interpreting qualitative data: Methods for analyzing talk, text, and interaction*. SAGE Publications Ltd.

Thorstensson, M., Axelsson, M., Morin, M., and Jenvald, J. (2001). Monitoring and analysis of command post communication in rescue operations. *Safety Science*, 39:51–60.

Trnka, J. and Jenvald, J. (2006). Role-playing exercise – a real-time approach to study collaborative command and control. *The International Journal of Intelligent Control and Systems*, 11(4):218–228.

Trnka, J., Johansson, B., and Granlund, R. (2006). Information support in collaborative command and control work – empirical research using a role-playing exercise approach. In *Proceedings of the 11th International Command and Control Research and Technology Symposium (ICCRTS)*.

van der Aalst, W., van Dongen, B., Herbst, J., Maruster, L., Schimm, G., and Weijters, A. (2003). Workflow mining: A survey of issues and approaches. *Data and Knowledge Engineering*, 42(2):237–267.

van der Aalst, W. and van Hee, K. M. (2002). *Workflow management: models, methods, and systems*. MIT Press, Cambridge, MA, USA.

Weick, K. E. (1995). *Sensemaking in organizations*. SAGE Publications Ltd.

Wen, L., Wang, J., and van der Aalst, W. M. P. (2009). A novel approach for process mining based on event types. *Journal of Intelligent Information Systems*, 32:163–190.

Witten, I. H. and Frank, E. (2005). *Data mining : Practical Machine Learning Tools & Techniques, Second Edition*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers Inc.