Personalized Change Awareness: Reducing Information Overload in Loosely-Coupled Teamwork

Ofra Amir[a], Barbara J. Grosz[b], Krzysztof Z. Gajos[b], Limor Gultchin[c]

[a] *Technion - Israel Institute of Technology*
[b] *Harvard University*
[c] *University of Oxford*

Abstract

Complex tasks such as treating patients with chronic conditions and developing software products are typically accomplished by teams that collaborate over an extended time duration. To remain coordinated, team members need to be aware of others' activities if those activities are likely to affect their own actions. However, identifying such interactions and sharing information appropriately is challenging, especially when the activities of team members are loosely-coupled. In practice, team members often either lack important information about others' activities, or are overwhelmed by the need to review too much information. This paper presents *Personalized Change Awareness*, a new approach for supporting team coordination which aims to automatically identify and share the subset of information about others' activities that is most relevant to each of the team members. The paper formally defines the computational problem of information sharing in loosely-coupled teamwork, which underlies the personalized change awareness approach. It defines a new representation, Mutual Influence Potential Networks (MIP-Nets) and an algorithm, MIP-DOI, that uses this representation to determine the information that is most relevant to each team member. In contrast to existing information sharing algorithms in multi-agent teams, MIP-DOI does not assume the availability of a priori knowledge of a team's possible plans, because human teams rarely explicitly define detailed long-term plans in advance. We demonstrate the ability of MIP-DOI to identify relevant information using simulations of collaborative activities. We further evaluated the contribution of personalized change awareness to team performance in a controlled user study. To this end, we developed a personalized change awareness mechanism for collaborative writing, which used MIP-DOI to determine which changes to share with each author. This evaluation demonstrates that the Personalized Change Awareness approach resulted in higher productivity and lower perceived workload without any change in final quality compared to the currently prevalent approach of sharing all change information. Our results also demonstrate that merely reducing the amount of information shared with co-authors is not enough: sharing a random subset of changes resulted in significantly lower quality of work than sharing a personalized subset of changes.

Keywords: Information sharing, Teamwork Support

1. Introduction

Multi-author papers, software development, and clinical care in medicine are examples of the increasingly prevalent settings in which people collaborate with others to accomplish complex goals. While technologies such as Google Drive, Dropbox and Github provide an infrastructure for teams to share work artifacts and contribute to complex activities in a distributed, asynchronous manner, the coordination of team activities remains a challenge in some settings. In a study of complex health care teams, we identified five teamwork characteristics that raise significant challenges to the coordination of teams in highly distributed settings: (1) Flat team structure, (2) Loose-coupling of activities, (3) Extended duration of the teamwork, (4) Continued revision of plans, and (5) Syncopated time scales of team members [1]. Such teamwork differs fundamentally

[*] Corresponding author
Email address: oamir@technion.ac.il (Ofra Amir)

Accepted to Artificial Intelligence, doi: 10.1016/j.artint.2019.05.005

from the teamwork settings addressed by prior work in Artificial Intelligence, the social sciences and computer supported cooperative work, and is not appropriately supported by existing systems [1].

This paper focuses on the problem of coordination of teams that have two of the these characteristics: loose-coupling and extended-duration of teamwork. Loose-coupling of activities enables team members to act autonomously most of the time and reduces the need for negotiation [2]. However, it also makes identifying dependencies and conflicts harder, and it leads to a lack of structured channels for communication [3, 4]. These problems are usually exacerbated when the teamwork extends over a long period, as plans and dependencies between tasks may change. Coordination failures often arise when team members are unaware of activities of others that have an effect on their own work.

We illustrate these coordination challenges with a scenario of collaborative writing, in which activities are typically loosely-coupled and extend over time, and coordination of authors' edits has been shown to be challenging [2, 5]. Consider an interdisciplinary team of researchers writing a grant proposal, which includes AI, HCI and health care researchers. The writing of the proposal is loosely-coupled for the most part, as each of the sub-teams focuses on writing the sections related to their area of expertise. Even so, there are dependencies between different sections, and authors need to ensure that the proposal is coherent as it evolves and changes throughout the writing process. For instance, if a running example of a health care scenario is introduced in the proposal, all sections should refer to it and the text should be adjusted if a change is made to the example; if the algorithms described in the AI section assume certain inputs from the users, the HCI section should describe how these inputs might be elicited from users. Since the writing of the proposal spans several weeks, the structure of the proposal is likely to change, making it harder to track changes and consistency.

Current systems for managing shared artifacts (e.g., Google Drive, GitHub) often include *Change awareness* mechanisms [6, 7] such as diff and track changes tools. These mechanisms display change information such as modifications to text or code to assist team members in tracking each other's activities. By enhancing team members' knowledge of others' activities, they provide context for team members to evaluate their own actions and ensure they align with the team's activity as a whole [7]. These mechanisms, however, do not help team members identify the information that is relevant for their own activities. Consequently, team members often face one or both of the following coordination challenges: (1) high coordination overhead as a result of information overload when too much information is shared, or (2) coordination failure due to lack of important information when too little information is shared or when relevant information cannot be found [5, 8, 9, 1].

For example, in the collaborative writing scenario of writing a grant proposal, when an author returns to the document after a couple of days, she will likely be overwhelmed if all the changes made by co-authors are presented, and therefore might ignore them altogether. The authors can try to summarize important changes they have made and share these summaries with their co-authors, resulting in increased effort to maintain coordination. Furthermore, even if the authors proactively share the modifications they think are most important for their co-authors to know about, they might miss important information if they fail to recognize a dependency between different parts of the proposal. In some domains, increased coordination effort and coordination failures can have severe consequences. For instance, in the complex health care domain, in which we first identified Loosely-coupled, extended duration teamwork [1], the poor coordination of activities has been shown to result in lower quality of care, unmet health needs and potentially preventable health care crises [10].

In this paper, we propose *personalized change awareness*, a novel approach for supporting efficient information sharing in loosely-coupled teamwork. Personalized change awareness mechanisms aim to share with team members only a subset of the change information that is deemed most relevant for their own activities. They have the potential to improve coordination in distributed loosely-coupled teamwork through simultaneously (1) lowering coordination overhead by reducing the total amount of change information each team member needs to review, and (2) improving the chances of coordination success by ensuring that each team member receives the information most relevant to her task.

The development of personalized change awareness mechanisms requires solving the computational problem of predicting the relevance of change information to each of the team members. We formalize this problem, which we refer to as the Information Sharing in Loosely-Coupled Extended-duration Teamwork (ISLET) problem, and present new computational methods for solving it. Specifically, we develop Mutual Influence Potential Networks (MIP-Nets), a new representation that models the allocation of tasks to team members and the potential dependencies between tasks, and MIP-DOI, an algorithm that uses MIP-Nets to choose the subset of information to share with each collaborator. This approach utilizes the extended-duration of the teamwork to learn collaboration patterns by observing team members' interactions and representing them using MIP-Nets.

We conducted two types of evaluations of personalized change awareness. First, we conducted a simulation of collaborative activity in which agents jointly solved a graph coloring problem. The results of this study demonstrate the ability of MIP-DOI to identify relevant information and its robustness to different teamwork characteristics.

Second, we designed and implemented a personalized change awareness mechanism (based on the MIP-DOI algorithm) for supporting collaborative authoring of documents and conducted a human subjects experiment to evaluate the impact of personalized change awareness on the performance of teams collaborating on a complex writing task. In our implementation, when an author returns to a shared document, the personalized change awareness mechanism highlights a subset of edits made by collaborators since her last edit that are deemed most relevant to her, thus limiting the amount of information she needs to review prior to making her own writing contributions. We compared this personalized change awareness mechanism (denoted *Personalized*) to two baselines: a change awareness mechanism that showed *all* of the changes (denoted *All*), and a change awareness mechanism that showed the same number of changes as the personalized mechanism, but selected the changes to share at *random* (denoted *Random*).

Our results show that both mechanisms that reduced the number of changes shown, *Personalized* and *Random*, led to higher productivity (measured as the amount of new information added to the summary by each participant in each session) and reduced perceived workload compared to the condition in which the *All* mechanism was used. This result demonstrates the benefits of reducing coordination overhead. Further, the changes shared by the *Personalized* mechanism were rated as more helpful than changes shared by the *Random* and *All* mechanisms. Importantly, the quality of the team's documents was higher (i.e., those documents contained fewer instances of conflicting or redundant information) with the *Personalized* mechanism than it was with the *Random* mechanism, demonstrating the importance of sharing relevant changes. Moreover, although the *Personalized* mechanism shared fewer changes than the *All* mechanism, the quality of the documents produced in these two conditions did not differ.

The paper makes the following contributions:

- It introduces the notion of personalized change awareness mechanisms.
- It formalizes the computational problem of information sharing in loosely-coupled extended-duration teamwork, which is at the core of personalized change awareness mechanisms.
- It defines MIP-Nets, a novel representation for modeling teamwork activities, and MIP-DOI, an algorithm that uses MIP-Nets to determine what information to share with each team member.
- It evaluates MIP-DOI in a simulation of team activity, demonstrating its ability to identify relevant information and its robustness to different characteristics of the teamwork.
- It presents an implementation of a personalized change awareness mechanism which uses the MIP-DOI algorithm to reduce the coordination overhead in collaborative writing.
- It demonstrates the benefits of the personalized change awareness mechanism over indiscriminate information sharing through a user study, showing that it resulted in lower workload and higher productivity while maintaining the same quality of work.

Parts of this work (the formulation of the ISLET problem, MIP-Nets and MIP-DOI) have been previously published [19].

The remainder of the paper is organized as follows: Section 2 describes related work; Section 3 formalizes the problem of information sharing in loosely-coupled extended-duration teamwork; The MIP-Nets representation and MIP-DOI algorithm are presented in Section 4, and the evaluation of MIP-DOI in a collaborative activity simulation is presented in Section 5; Section 6 describes the implementation of a personalized change awareness mechanism for collaborative writing; The experiment evaluating the personalized change awareness mechanism is described in Section 7 and its results are presented in Section 8; We discuss the results and future work in Section 9.

2. Related Work

In this section we review prior work from the artificial intelligence literature on algorithms for reasoning about information sharing in teams, as well as prior work from the human-computer interaction literature on change awareness mechanisms. Last, we contrast the problem of personalized change awareness from typical recommender system problems (e.g., movie recommendations).

2.1. Information Sharing in Multi-Agent Systems

Prior work in the multi-agent systems literature has developed communication mechanisms to support agents' coordination. In particular, communication plays a key role in Belief-Desire-Intention (BDI) theories of teamwork [4, 20, 21], and BDI planning frameworks draw on these theories in designing communication rules and computing the value of information. Communication methods have also been developed in the context of decision-theoretic multi-agent planning models. We next review key approaches from the BDI and decision-theoretic planning, as well as approaches that combine BDI and decision-theory. Finally, we distinguish these planning- and value of information-based communication approaches from our approach which does not assume knowledge of teams' plan models.

Communication in BDI planning. Theories of teamwork and collaboration [4, 20, 21] emphasize the key role of communication in ensuring coordinated teamwork. BDI approaches to multi-agent planning typically base their communication mechanisms on these theories [4, 20, 21]. For example, the joint intentions model [20] defined conditions for communication, such as communicating to establish joint intentions, communicating the achievement of a goal, and communicating when learning that a goal cannot be achieved. These ideas were used in the STEAM multi-agent framework [22]. In this framework agents use a decision tree to determine whether to communicate information about operation termination by considering their belief about the joint intentions of the team and weighing (the domain specified) costs and risks of not communicating. Other works have developed teamwork programing languages that include explicit rule-based communication mechanisms [23, 24] as well as teamwork infrastructure that supports dynamic role allocation [25, 26, 27].

Communication in Decision-Theoretic Planning. Prior work on decision theoretic approaches to multi-agent communication can generally be classified to two types: approaches that reason about communication during planning time, and approaches that reason about communication during execution time. The DEC-POMDP-COM model [28] and the COM-MTDP model [29] provide a theoretical model for reasoning about communication during planning time and include communication actions in the agents' policies. Fowler et al. [18] develop the Intelligent Knowledge Distribution framework which is used to determine what information to share with whom using constrained action POMDPs. Spaan et al. [30] developed a model in which the communicated information is included in the actions vectors of agents, and is then incorporated into the observation vectors received by agents in the next time step. Offline approaches assume that all possible observations that agents may receive are known during planning time.

Other approaches reason about communication during execution time [31, 32, 33, 15, 16, 17]. Such approaches aim to identify situations in which communication would improve the group's performance, based on observations obtained by agents. This reduces computational complexity since agents do not need to consider in advance what to communicate at each possible scenario, but rather only reason about communication given their actual observations. For example, Roth et al. [34] reason about communication by growing a tree of the possible joint beliefs of the team. This work has been extended to consider not only when to communicate, but also what subset of observations to communicate to other team members [11]. The approaches proposed in the above two works assume a known joint policy, such that agents can deterministically determine what action each agent in the team would take following communication. Wu et al. [35, 14] propose an algorithm for online planning and communication. The algorithm merges observation histories based on their similarity in terms of future actions. They then use these history clusters to reason about histories incompatibility, which determines communication. This model does not include communication cost, but instead tries to maximize utility while minimizing the amount of communication.

Integrated BDI-Decision-Theoretic Communication Methods. There are few prior works that have combined BDI concepts with decision theoretic approaches to reason about communication. STEAM was the first system to integrate reasoning about the utility of sharing information with a BDI planning framework [22]. Inspired by BDI teamwork, Kwak et al. [36] define trigger points in which communication should be considered in the context of a DEC-POMDP model.

The Complete Plan Knowledge Assumption. All of the approaches described above, both BDI and decision theoretic, rely on a *complete plan knowledge assumption*; they assume availability of a complete domain model of the actions or plan library, state space, and utilities or goals. They use this model and knowledge of a team's plans or policies to compute the value of information. Although some approaches assume only incomplete knowledge of agents' plans and use reinforcement learning [37, 38] or plan recognition [39, 40] to infer other agents' plans or parts of the environment model (e.g., transition and reward functions), these approaches still assume a known planning domain (i.e., known state space and actions in MDP frameworks, or known plan library in plan recognition approaches). Recent work has also considered coordination in ad-hoc teams [41, 38] where no pre-coordination is done, but communication in these settings also assumes a known model of the world (e.g., an MDP).

In loosely-coupled human teamwork settings, such detailed plan models are rarely explicitly specified. For example, complex health care teams might agree on high-level treatment goals but never fully specify a long-term plan [1, 42]. The approach we present (first presented in Amir et al. [19]) does not rely on the complete plan knowledge assumption. Instead, it utilizes the extended-duration of the teamwork to learn about the underlying task structure without assuming any a priori knowledge.

2.2. Change Awareness Mechanisms

Change awareness (CA) mechanisms draw team members' attention to changes others have made to a shared artifact by marking those changes in some way. Many current commercial tools provide CA mechanisms, e.g., Word's track changes, Google Docs suggest mode and revision histories, GitHub commits, Wikipedia revision history. These tools, however, typically show users *all* of the changes that were made by their collaborators, which can lead to information overload.

Tam & Greenberg [6] provided a framework of the critical information people need if they are to maintain change awareness. They categorized the types of questions that can be answered by change awareness mechanisms, such as where changes have been made, by whom and when. The framework also describes information elements based on which a system could answer these questions, including edit history, location history and authorship history. Some prior CA mechanisms provide team members with change filtering options. For example, PastDraw provides filtering options such as choosing the types of changes to show (e.g., deletions, additions) and for which types of objects to display changes [6]. Prinz et al. proposed "anticipative" CA mechanisms [43], which allow team members to specify ahead of time the changes they would like to be notified about (for example, letting a user know when a document was opened by someone). In the context of collaborative writing, flexible Diff-ing [44] provides users with ability to filter changes based on features such as the granularity of the edits. In the context of multi-player games, awareness cues were suggested as a way to support communication between players [45].

While these approaches can help reduce information overload, they require manual input from the users in selecting the changes to show. In contrast, the proposed personalized change awareness mechanism automatically reasons about the relevance of changes to users. It does not require manual input from users. We use the same types of information elements described by Tam & Greenberg, in particular the *edit history*, but in a different way: rather than using the edit history only to extract change information, it is used by the system to learn about the interests of team members, the task allocation among collaborators and the dependencies between the tasks.

Most closely related to our approach are methods for filtering notifications in software development settings. The NeedFeed system [46], for example, models code relevance for developers by analyzing "touch histories" (which classes a developer has modified) and using more complex history-based classifiers. Similarly, Holmes & Walker [47] proposed a recommendation approach to filter notifications about change events based on deployment dependencies. Their approach uses various code features and code ownership analysis. Omoronyia et al. [48] developed a system that aims to enhance collaboration awareness by showing tasks, developers and artifacts that are considered relevant to a user given their current context.

These software development approaches are similar in spirit to the personalized change awareness methods in that they evaluate the relevance of information about others' actions to users. However, they rely heavily on the explicit structure that is available in software programs (e.g., class dependencies, code documentation, method call graphs, use cases). In many domains, there is much less explicit structural information. For instance, in writing there may be a hierarchical section structure, but the dependencies between sections are much less apparent than in code, where the dependencies between methods and classes are often explicitly defined. For example, there may be a dependency between the Results section of a paper and a paragraph describing the results in the Introduction section, but this dependency is never explicitly specified. This paper also differs from these prior works with respect to system evaluation: while prior work only assessed the relevance of the identified changes based on code revision histories, we directly evaluate the benefits that the developed personalized change awareness mechanism provides to the teamwork.

Other relevant prior work includes visualization tools that support distributed software development by creating visual representations of software artifacts and development activities [49, 50, 51, 52, 53]. In contrast to the personalized change awareness approach, these visualizations help users be aware of other artifacts and people who might be affected by their work rather than trying to draw their attention to relevant changes. Importantly, they do not create *personalized* views. Similarly to the systems described above, they also rely on the available structural information in software development.

2.3. Recommender Systems Approaches to Information Filtering

The problem of identifying relevant information to share with team members can be viewed as a type of recommendation problem, in that the goal is to choose a subset of items to draw a person's attention to. However, recommending relevant information in the context of a *collaborative activity* results in key differences compared to one-shot recommendations to individuals (e.g., movie recommendations).

First, the set of items is unique to a particular team activity and evolves quickly. For example, the paragraphs in a paper are specific to that paper as opposed to a set of movies which is consistent across people, and paragraphs are added and deleted at a fast pace. This means that there will not be sufficient data to apply collaborative filtering approaches [54], a prevalent approach in recommender systems. Content-based recommender systems [55] make use of metadata about items (e.g., movie genre, actors) to make recommendations. These approaches cannot be immediately applied because the assumption of known metadata about items does not hold, and even if some domain knowledge exists, there will not be substantial data available due to the uniqueness of different team tasks.

Second, the extended temporal duration of collaborative activities and their inherent structure is not addressed by existing recommender systems methods. Our approach utilizes the underlying structure of the task as it is revealed over time based on team members' interactions to determine the subset of information to be shared with each team member.

3. The Information Sharing in Loosely-Coupled Extended-Duration Teamwork Problem

Enabling personalized change awareness mechanisms requires methods that can quantify the relevance of changes to each of the collaborators. In this section, we formally define the Information Sharing in Loosely-coupled Extended-duration (ISLET) problem, that is, the computational problem of choosing the most relevant subset of change information to share with team members.

An *ISLET problem setting* comprises the following:

- P: a set of collaborating partners. The set can change over time with partners joining or leaving the team.
- O: a set of objects that partners interact with. The set can change over time as a result of partners' actions.
- A: the set of act-types $\{ADD, MOD, DEL\}$ for adding, modifying or deleting objects.[1] These general domain independent act-types are specialized to domain-specific act-types in each application domain.
- S: interaction sessions of partners. A session $s(p, t, (\langle a_1, o_1\rangle, ..., \langle a_{|s|}, o_{|s|}\rangle))$ is defined by a triple: the partner acting, the time of the session, and a set of pairs of act-types and the objects they operate on $(\langle a_i, o_i\rangle)$[2]. For brevity, we denote a session recorded at time t as s_t. Here, t denotes the starting time of the session.

[1] We note that other actions such as merging objects or splitting objects are not explicitly supported in this formulation. The result of such actions would in practice be the creation of a new object (in the case of splitting) or deletion of an object (in the case of merging).

[2] We use only the $\langle a_i, o_i\rangle$ pairs to emphasize that the partner and time are the same for all actions taken in a single session.

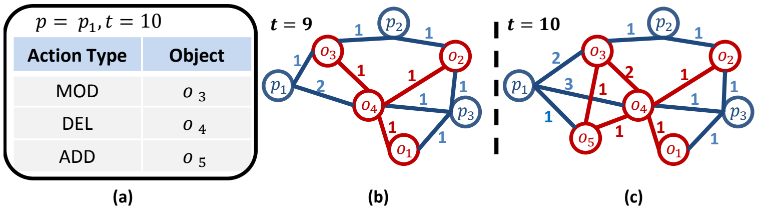


Figure 1: (a) An interaction session s_{10}; (b) The MIP-Net after sessions $s_1 - s_9$, numbers on edges correspond to the weights of the edge; (c) The updated MIP-Net after session s_{10} with updated edge weights (e.g., the weight on the edge connecting p_1 and o_3 increased from 1 to 2 because p_1 edited o_3 in the session).

The *ISLET problem* is to determine a set of objects $O_{share} \subset O$, where $|O_{share}| \leq l$, to inform $p \in P$ about, given sessions s_1 to s_{t-1} and the identity of the partner p who is starting s_t. The constraint on the cardinality of O_{share} (l) is a communication budget, which restricts the amount of information that can be shared. It reflects the need not to overwhelm partners with too much information. The objects in the set O_{share} should be *relevant* to the partner. The notion of relevance has been widely discussed in the literature on cognition and communication [56]. Intuitively, information is relevant if it will affect the partner's actions. The specific definition of relevance, however, is domain dependent.

To illustrate, we will describe a collaborative writing scenario which we will use as a running example throughout the paper. In this scenario, a group of researchers (the P), comprising Alice, Bob and Chris, writes a grant proposal together. The set of objects (O) includes the paragraphs of the proposal. Specializing to the domain and applying act-types (A) to objects yields such actions as writing new paragraphs, removing paragraphs or editing paragraphs.

Sessions (S) are added over time as Alice, Bob, and Chris edit the document. For example, assume Alice (p_1) edits the document on Monday morning (t_{10}), taking the following actions: modifying paragraph 3 ($\langle MOD, o_3 \rangle$), deleting paragraph 4 ($\langle DEL, o_4 \rangle$) and adding a new paragraph ($\langle Add, o_5 \rangle$). These actions together constitute the session shown in Figure 1(a). Note that the set O evolves as paragraphs are added or deleted and P can also evolve over time; for instance, Dan might join in writing the proposal.

In the writing context, the ISLET problem is to choose a subset of modified paragraphs to share with authors as they begin a new editing session. For example, assume that Chris begins editing the document on Tuesday after two days of not looking at the document, and $l = 2$. Then O_{share} should include the two paragraphs that have changed (edited, added or removed) since Chris completed his edits on Sunday and that are deemed most relevant to Chris' activities.

4. Reasoning about Information Sharing with Mutual Influence Potential Networks

To address the ISLET problem, we define a new representation, Mutual Influence Potential Networks (MIP-Nets). MIP-Nets represent interactions between partners and objects and between different objects. They are intended to capture useful information about the structure of the collaborative activities, that is, the potential relatedness of tasks and the allocation of tasks among team members. MIP-Nets are updated online based on partners' sessions, and are used to reason about what information to share with partners. We next describe in detail the MIP-Nets representation, the update procedure and the algorithm used to determine what information to share based on MIP-Nets.

4.1. The MIP-Nets Representation

MIP-Nets consists of nodes representing partners and objects. A particular partner $p \in P$ and an object $o \in O$ are represented by nodes n_p and n_o, respectively. Henceforth, we use $\mathbf{p}$ when referring to a particular partner and $\mathbf{o}$ for a particular object.

Formally, a MIP-Net consists of:

- N_P: a set of partner nodes.
- N_O: a set of object nodes.

- E, W: a set of edges E, each edge connecting a partner node with an object node or two object nodes, and weights W specifying the weight of each edge.

Particular nodes n_p and n_o are connected by an edge if **p** performed an action on **o**. The edge weight corresponds to the extent of the interaction: if **p** takes many actions that affect object **o**, this will be reflected by a high weight on the edge connecting n_p and n_o. Thus, the weights on such edges represent information about team members' responsibilities, which we refer to as "role allocation".

Similarly, n_o and $n_{o'}$ are connected by weighted edges based on the frequency at which the objects they represent are modified in the same sessions. Edges connecting object nodes thus represent object interactions, i.e., the extent to which team members tend to change one object when they change the other. We refer to these object interactions as the "task structure", because these groupings are likely to be a reflection of an underlying task. For instance, in the grant proposal example, paragraphs reporting results in a Results section and in the Introduction section might be frequently edited together as part of the same underlying task of adding new results to the paper.

Importantly, the sense of task structure modelled by MIP-Nets is much looser than that used in formal plan representations such as Hierarchical Task Networks. While plan representations explicitly specify dependencies between different tasks (e.g., using pre-conditions, effects), the MIP-Nets representation learns over time about potential relatedness of different tasks implicitly by observing partners' interactions. We also note that we chose not to directly model connections between different partners, as we do not assume access to direct communication among them. However, such interactions will be captured implicitly through connections to objects that both partners interact with. In domains where there is access to communication between partners (e.g., email messages), it might be beneficial to add edges that directly connect partner nodes.

Figure 1(b) shows a sample MIP-Net. Partner nodes and edges connecting partners and objects are shown in blue. Object nodes and edges connecting them are shown in red. Numbers on edges represent the edge weights.

4.2. Constructing and Updating MIP-Nets.

MIP-Nets are constructed and updated over time based on partners' sessions. At the end of each session s_t, the MIP-Net is updated. The MIP-Net update procedure, shown in Algorithm 1, first checks whether **p** is already represented by a node in the MIP-Net. If not, a new node is added to N_P (lines 1–2). Next, it iterates over all actions in the session; new object nodes are added as a result of ADD actions, and the weights of edges connecting n_p with object nodes representing objects on which that partner acted are incremented by d, where d corresponds to the importance of the action (e.g., considering the extent of change to a paragraph) (lines 4–8). Similarly, the weights of edges connecting object nodes representing objects that the partner interacted with in the same session are incremented by d (lines 9–15). Last, because collaboration patterns can change over time, weights on edges that have not been updated are multiplied by a decay factor of $0 < \lambda \leq 1$ to decrease the influence of past actions. This includes edges between n_p and nodes representing objects that **p** did not modify in this session (lines 16–20), as well as edges between objects that have not been co-edited in the session (lines 21–25). We note that nodes representing deleted objects persist in the MIP-Net as information about their connections can implicitly reveal interactions between other objects. However, their influence will decay over time depending on the λ parameter, such that eventually they will not have an influence over information sharing decisions.

To illustrate the MIP update procedure, consider the collaborative writing scenario described in Section 3: assume the MIP-Net at time $t = 9$ is the one shown in Figure 1(b). Following s_{10} (Figure 1(a)), the MIP-Net is updated, yielding the network shown in Figure 1(c). For simplicity, in this example we do not make use of the weight decay process (i.e., $\lambda = 1$), and assumed that edges are incremented by 1 (no effect to the extent of change). As shown in the figure, a node representing o_5 was added to the MIP-Net and the weight on edges connecting p_1 (the node representing Alice) with o_3, o_4 and o_5 were incremented. The weights on edges connecting all pairs of objects included in the session (e.g., o_3 and o_4) were also incremented. Although o_4 was deleted, the node representing it persists in the MIP-Net as its connections to other nodes may still be informative.

The computational complexity of this procedure is dominated by $|s|^2 + |E|$, where $|s|$ is the number of $\langle a, o \rangle$ pairs in the session and $|E|$ is the number of edges in the MIP-Net. The update procedure requires

ALGORITHM 1: The MIP-Net update procedure.

```
Input: $s(p, t, (\langle a_1, o_1\rangle, ..., \langle a_{|s|}, o_{|s|}\rangle))$
if $n_p \notin N_P$ then
    $N_P = N_P \cup n_p$
end
for $\langle a, o\rangle \in s$ do                                      // increment weights of p-o edges
    if $a = ADD$ then
        $N_O = N_O \cup n_o$
    $IncrementWeight(n_p, n_o, d)$
end
for $\langle a, o\rangle \in s$ do                                      // increment weights of o-o edges
    for $a', o' \in s$ do
        if $o \neq o'$ then
            $IncrementWeight(n_o, n_{o'}, d)$
        end
    end
end
for $(n_{p_i}, n_{o_j}) \in E_{p-o}$ do                                    // decay weights of p-o edges
    if $(p_i = p \quad \& \quad updated(n_{p_i}, n_{o_j}) = False)$ then
        $DecreaseWeight(n_{p_i}, n_{o_j}, \lambda)$
    end
end
for $(n_{o_i}, n_{o_j}) \in E_{p-o}$ do                                    // decay weights of o-o edges
    if $updated(n_{o_i}, n_{o_j}) = False$ then
        $DecreaseWeight(n_{o_i}, n_{o_j}, \lambda)$
    end
end
```

one iteration over the set of $\langle a, o\rangle$ pairs to update the weights connecting n_p with nodes representing the objects interacted with during the session, and a second iteration over all pairs of objects o, o' that were interacted with in the session to update weights on edges connecting object nodes. The decay procedure requires iterating over all the edges in the MIP-Net ($|E|$).

4.3. The MIP-DOI Algorithm

The MIP-DOI algorithm uses the MIP-Net to reason about information sharing in ISLET problem settings. To quantify the relevance of modifications to some object **o** to some partner **p**, we use the concept of *Degree-Of-Interest* (DOI). Furnas [57] defined $DOI(x \mid y)$ as the degree of interest a user has in an item x, given that the user is focused on some item y. It is computed it as follows:

$$DOI(x \mid y) = \alpha \cdot API(x) + \beta \cdot D(x, y) \tag{1}$$

$API(x)$ is the *a priori* importance of item x. It is independent of the user's identity and aims to reflect the *global* importance of an item. $D(x, y)$ is the distance between x and y. It aims to reflect the importance of x given the user's context.

The rationale behind this formulation is that, generally, a user will be interested in items that are close to her current focus of attention, as well as in items that are of general importance [57]. This notion of DOI fits our purposes, as collaborators will likely find value in information about objects that are closely related to objects they interacted with or currently focus on, as well as in information about objects that appear to be of significant importance to the team's activities as a whole.

As initially introduced, DOI was computed over items in a tree. Similar to Van Ham and Perer [58], we use a network-based DOI metric. In our formulation of DOI, we consider two different nodes as representing **p**'s focus of attention: (1) the node representing the partner in the MIP-Net (n_p), as the edges from n_p capture the extent of interaction between **p** and the different objects, and (2) the node representing the object that the partner acts on at the beginning of a session, denoted o_f for "focus object". In many settings, information about o_f is available to the system (e.g., observing the paragraph Alice starts editing) and can be integrated in the DOI computation. In sum, we measure DOI by computing:

$$DOI(o \mid p, o_f) = \alpha \cdot API(n_o) + \beta_1 \cdot D(n_o, n_p) + \beta_2 \cdot D(n_o, n_{o_f}) \tag{2}$$

The distance values $D(n_o, n_p)$ and $D(n_o, n_{o_f})$ can be computed using various distance measures for networks. We used the Adamic/Adar proximity metric [59], which measures the amount of shared links between two nodes while accounting for the likelihood of a node to be shared (i.e., shared neighbors that have a high degree get less weight as they are likely to be shared by many nodes). Adamic/Adar is computed as follows:

$$A(x, y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{log|N(u)|} \tag{3}$$

Where $N(u)$ is the set of nodes adjacent to u. We adapted to take into account edge weights by replacing $N(u)$ with the sum of all weights on edges connected to u. We chose to use this metric since it was shown to perform well on the problem of link prediction in networks [60] which also requires identifying connections between different nodes in a network. Since Adamic/Adar is a proximity (rather than distance) metrics, the β coefficients in our formulation are positive.

Network centrality metrics can be used to compute the *a priori* importance of an object node n_o. Our implementation uses $deg(n_o)$ (the sum of weights on edges connected to n_o). This measure will result in higher centrality to nodes representing objects that partners frequently and substantially interact with. Note that the importance of objects can change over time. For instance, if many partners interact with an object, its degree will increase and thus its centrality will increase.

To determine the set of objects $O_{share} \subset O$ to share with $\mathbf{p}$, the MIP-DOI algorithm computes $DOI(o \mid p, o_f)$ for each $o \in O$ and chooses the l objects with the highest DOI. (recall, l is the communication budget.) The computational complexity of MIP-DOI depends on the methods used to compute API and D. In our implementation it is dominated by $|O|^2$.

A key decision in the implementation of MIP-DOI is choosing the weights for the different components (i.e., α, β_1 and β_2). Assigning a high weight to α will result in preference for objects that are modified more frequently and substantially. Assigning high weights to the β coefficients will result in more personalization to the particular partner, with higher β_1 values resulting in personalization based on a partner's involvement in the teamwork over time, while higher β_2 values resulting in personalization that's focused on the partner's *current* activities.

We envision that ultimately, when integrating MIP-DOI into a personalized change awareness mechanisms, users will be able to control the weighting of the different components by interacting with the system (e.g., could ask to rank objects based on their importance to the team overall, their relevance to the current user's work, etc.). In addition, the system could learn to adapt the coefficients based on feedback from users regarding the relevance of shared information. We discuss this further in Section 9. In the simulations described in the next section, we include results for several coefficient configurations to explore the effect of coefficient values on the performance of MIP-DOI.

5. Evaluation of Relevance of Shared Information in a Collaborative Activity Simulation

The ultimate goal of personalized change awareness mechanisms is to support teamwork by reducing coordination overhead and improving coordination success. To achieve this, they need to be able to identify relevant information and share it with team members. Therefore, we first evaluated the ability of MIP-DOI to identify relevant information by designing a simulation of an abstract collaborative activity simulation. We conducted this simulation prior to integrating MIP-DOI into a real-world system and testing its impact on human teamwork for two reasons. First, in the simulation environment provided a ground truth for assessing the relevance of information. Second, it provided a more controlled environment in which to test and diagnose the performance of the algorithm. In particular, collaborative activities can vary in many aspects, including the size of the group, frequency of interactions and coupling of tasks. For example, Wikipedia articles are written by a large number of authors with a small percentage of the authors making the majority of contributions, while academic papers are typically written by a much smaller number of authors who act in a more coordinated way (e.g., they might divide responsibilities for different sections). Software projects hosted on GitHub also differ significantly in the nature of the collaboration on projects [61]. Some projects include a small group of collaborators that contribute fairly equally, while others have one or two main contributors and a large number of developers who make only a single contribution. In health care, the role allocation among care providers is much more strict due to their specialization. The simulation enables exploration of the effects of such aspects of teamwork in a controlled manner.

5.1. Design of the Simulation

In the collaborative activity simulation a group of partners (P) confronts a constraint satisfaction problem that abstracts the type of coordination problems that arise in collaborative activities. The partners collaboratively color a graph $G(V, E)$ using a set C of colors such that no two neighboring vertices are assigned the same color. Constraints on the colors of neighboring vertices correspond to a group's need to align their activities. For example, in the writing scenario, a paragraph summarizing the results in the introduction of the paper must align with the results described in the results section. In health care, a choice of a course of treatment for one condition can constrain treatment of other conditions if they have conflicting effects.

We formulate this collaborative activity as an instance of an ISLET problem as follows:

- P: collaborating partners.
- O: graph vertices.
- A: The act-types MOD, DEL and ADD are instantiated as follows: $mod(v, c, c')$ changes the color of v from c to c', where $c, c' \in C$. $add(v)$ adds a new vertex v' as a neighbor to an existing vertex v. $del(v)$ removes vertex v from the graph.
- S: Interaction sessions: the session $s(p, t, (\langle a_1, o_1\rangle, ..., \langle a_k, o_k\rangle))$ consists of the changes made to the graph by p at time t.

For simplicity, in this section we describe a simulation in which the set of objects is constant (i.e., only the MOD act-type is used).

Importantly, our goal is not to propose a new distributed algorithm for solving constraint satisfaction problems. Rather, our goal is to use this canonical abstract problem to enable testing of information sharing approaches. Specifically, we use it to measure the performance of MIP-DOI with respect to its ability to determine what information about vertices' colors to share with each partner before the partner decides which actions to take.

We designed the simulation in a way that preserves two key aspects of the ISLET settings: (1) the underlying task structure and role allocation are not explicitly specified and are therefore unknown to the information sharing algorithms, and (2) partners may know about the task structure, but cannot review all the actions of their collaborators. They therefore do not have complete knowledge of the state of objects that they do not directly interact with, and may not notice conflicts. For example, Alice might know that there is mutual dependency between different sections of a proposal, but not be aware of inconsistencies in current versions of the sections without reading them.

The first aspect is preserved by not providing the algorithms access to the graph structure (G). By observing partners' sessions they learn about the existence of objects (vertices) that partners interacted with and their colors, but they do not have information about edges. The second aspect is preserved by providing partners only with knowledge of the graph structure (i.e., the edges between vertices). Partners do not know the current color of a vertex unless it was shared with them, and they assume a vertex's color has not changed until they receive new information.

The graph structure and the way partners choose O_{modify} at each round affect the behavior of partners and the relevance of partners' actions to other team members. As described earlier, the proposed information sharing methods aim to support loosely-coupled teamwork. In such teamwork, we expect partners to have some (possibly non-strict) allocation of roles. This means that we expect each partner to focus on some subset of the objects. We further expect that there would be more constraints between objects that are modified as part of a specific role or task, than between objects that are modified in the context of different roles or tasks. For example, there are likely to be more interactions between paragraphs in the same section than between paragraphs in different sections. Similarly, in complex health care, there are likely to be more interactions between treatments related to one aspect (e.g., mobility) of the care, than between treatment related to different organ systems.

To model the loosely-coupled nature of the teamwork, we generate clustered graphs, where vertices within each cluster are more likely to have an edge connecting them than vertices that belong to different clusters. We specify the number of clusters, the probability of creating an edge between vertices in a cluster (p_{within}) and the probability of creating an edge between vertices in different clusters ($p_{between}$). We then randomly generate graph instances using these parameters.

To reflect role allocation among the partners, we assign each partner a probability distribution over vertices. Each partner is assigned a single primary cluster, and most of the probability mass ($p_{primary}$) is assigned to that cluster. The choice of the *focus object* (o_f) in each session is based on these distributions: with probability $p_{primary}$, **p** will choose an object o_f from its primary cluster, while it will choose o_f from another cluster with probability $1 - p_{primary}$. The remaining $k - 1$ vertices in O_{modify} are chosen in proportion to their distance from o_f in the graph. Specifically, the probability of choosing a vertex decreases exponentially with its distance from the o_f. The rationale for this choice of objects is that partners are more likely to choose objects related to their primary task, and further that they are more likely to modify a set of objects that interact with each other. For example, when editing a document, Alice will have a task in mind and will not simply choose a random set of 5 paragraphs to edit, but rather edit paragraphs that are related to some higher-level aspect of the paper).

In each round of the simulation procedure, shown in Algorithm 2, the partners take turns modifying vertex colors, as follows: (1) each partner **p** chooses a *focus object*, denoted o_f, and a set of k objects to modify denoted O_{modify} (line 3). The object o_f is chosen from the partner's primary cluster with probability $pr_{primary}$ (and from a different cluster with probability $1 - pr_{primary}$). The remaining $k - 1$ vertices in O_{modify} are chosen in proportion to their distance from o_f in the graph, to reflect higher likelihood of a partner carrying out activities that are closely related to each other in each session; (2) A set O_{share} of l objects to inform **p** about are chosen by the information sharing algorithm, given **p**, o_f and sessions s_1 to s_{t-1} (line 4); (3) The belief of **p** about vertices' colors is updated to reflect the shared information (line 5); (4) **p** chooses colors for objects in O_{modify}, such that the assignment minimizes the number of conflicts known to **p**, based on its updated belief (line 6); (5) The problem instance is updated to reflect the new coloring (line 7).

ALGORITHM 2: The graph coloring simulation procedure.

```
   Input: $P, problemInstance, k, l, maxRounds$
 1 while $t < maxRounds$ do
 2     for $p \in P$ do
 3         $O_{modify}, o_f = p.chooseObjects(k)$
 4         $O_{share} = getObjectsToShare(l, o_f)$
 5         $p.updateBelief(O_{share})$
 6         $s_t = p.chooseActions(O_{modify})$
 7         $problemInstance.update(s_t)$
 8     end
 9     $t = t + 1$
10 end
```

5.2. *Evaluation Metrics*

We consider an object $\mathbf{o} \in O_{share}$ relevant if there is an edge connecting **o** to at least one object in O_{modify}, as such information can directly affect **p**'s choice of action. We measure precision ($\frac{|O_{relevant} \cap O_{share}|}{|O_{share}|}$) and recall ($\frac{|O_{relevant} \cap O_{share}|}{|O_{relevant}|}$).

5.3. *Algorithm Comparisons*

We evaluate the performance of the following algorithms:

- **Omniscient**: has access to the graph structure and chooses objects in proportion to their distance from o_f.

Parameter	Description
$\|P\|$	Number of partners [5]
$\|Cl\|$	Mean cluster size [10]
$pr_{primary}$	Probability o_f is chosen from the primary cluster [0.8]
pr_{within}	Probability of creating an edge between vertices in the same cluster [0.3]
$pr_{between}$	Probability of creating an edge between vertices in different clusters [0.05]
$k = \|O_{modify}\|$	Number of actions p can take in a single session [3]
$l = \|O_{share}\|$	Number of objects that can be shared in a single session

Table 1: The parameters controlling simulation configurations. Values in brackets were used in the main experiments described in Section 5.4.

- **Most frequently changed**: chooses objects that were changed most *frequently* by partners.
- **Most recently changed**: chooses objects that were changed most *recently* by partners .
- **Random**: chooses objects randomly.
- **MIP-DOI**: varying the coefficients α, β_1 and β_2. We focus on the following configurations to test the effect of the different DOI components, and describe a few combinations of the components:
  - **MIP-DOI-centrality**: the DOI computation only considers objects' centrality ($\alpha = 1$). This configuration shares information only based on the extent to which an object is modified by the partners and does not provide any personalization.
  - **MIP-DOI-partner**: the DOI computation only considers objects' proximity to the partner node ($\beta_1 = 1$). This configuration personalizes information sharing according to overall involvement of a partner in the group activity.
  - **MIP-DOI-focus**: the DOI computation only considers objects' proximity to the focus object node ($\beta_2 = 1$). This configuration personalizes information sharing according to a partner's *current* activity.

With all MIP-DOI configurations, we used Algorithm 1 to update the MIP-Net at the end of each session. We did not use the weight decay process in the simulation (i.e., we set $\lambda = 1$). To ensure a fair comparison, all algorithms (MIP-DOI and baselines) choose the l vertices to share with $\mathbf{p}$ from the set of objects that were changed last by some $p' \neq \mathbf{p}$. We also used the same seed when generating random numbers for determining stochastic decisions (e.g., the choice of o_f) such that all algorithms are evaluated using the same conditions.

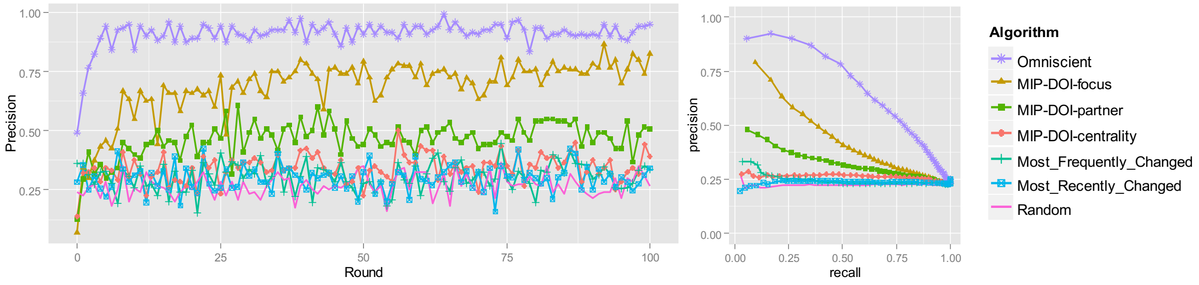
5.4. Simulation Results

This section reports in detail the results of a simulation that used the parameter values shown in brackets in Table A.8. The relative performance of the different algorithms was consistent across other parameter settings. That is, the relative performance of the algorithms compared to each other remained the same. Additional results using different parameter settings are included in the Appendix and summarized in this section.

Figure 2(a) shows the precision obtained by each of the algorithms with the communication budget $l = 3$. Overall, all MIP-DOI configurations significantly outperformed all baselines except, of course, for the omniscient baseline which has access to the graph structure. As can be seen in the figure, of the MIP-DOI configurations, MIP-DOI-focus achieved the best performance. Over time, its performance becomes close to that of the omniscient algorithm as more information about the task structure is accumulated in the MIP-Net.

If algorithms do not have access to o_f, MIP-DOI-partner (proximity of objects to partners) still outperforms all the uninformed baselines, demonstrating that MIP-Nets effectively recover information about partners' role allocation (i.e., their cluster assignment). MIP-DOI-centrality, despite not incorporating the proximity of objects to o_f or $\mathbf{p}$, still outperforms the other baselines, but achieves relatively low accuracy.

Figure 2(b) shows precision-recall curves for the algorithms. The curves were generated by varying the communication budget l between 1 (the leftmost points in Figure 2(b)) and the total number of changed objects considered for sharing. The results are aggregated starting from round 15, a point at which the MIP-Net



(a)

(b)

Figure 2: (a) Average precision by round (10 different graph instances with 5 runs each). (b) Precision-recall curve generated by varying l; each point shows the precision and recall for a given communication budget (l) with results aggregated from rounds $t_{15} - t_{99}$.

	$\beta_2 = \mathbf{1}$ (focus)	$\alpha = \mathbf{0.3}, \beta_2 = \mathbf{0.7}$	$\beta_1 = \mathbf{1}$ (partner)	$\alpha = \mathbf{0.3}, \beta_1 = \mathbf{0.7}$
$\mathbf{t_0 - t_{14}}$	0.40	0.45	0.33	0.36
$\mathbf{t_{15} - t_{99}}$	0.69	0.55	0.42	0.39

Table 2: Average precision obtained by MIP-DOI with different configurations in early and late rounds of the simulation. Incorporating the apriori importance (α) leads to better performance in early rounds, but to lower performance in later rounds.

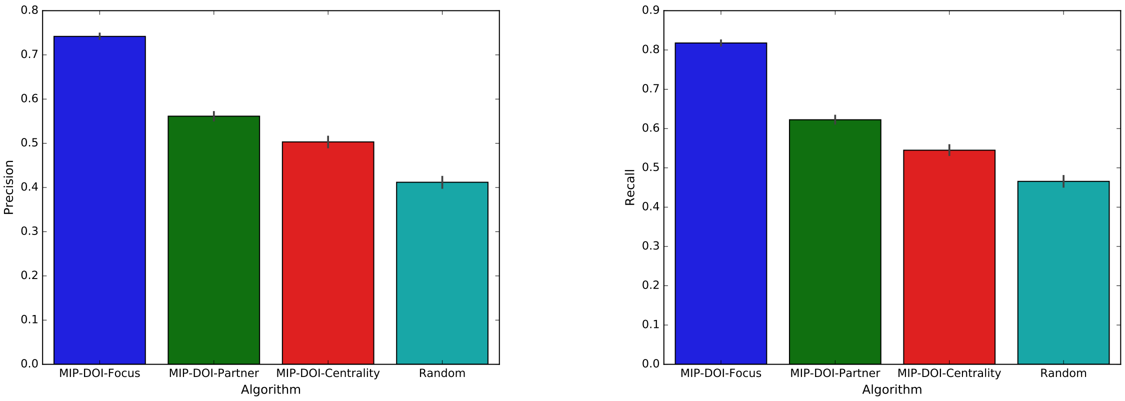
has accumulated some information about partners' activities. As can be seen in the figure, all configurations of MIP-DOI significantly outperform the uninformed baselines. The gap between the performance of MIP-DOI-focus and the omniscient algorithm is relatively small when using very limited communication budgets ($l \leq 3$), demonstrating that the MIP-Net representation can effectively distinguish between clearly relevant objects (high proximity to o_f) and clearly irrelevant objects (low proximity to o_f). For larger values of l, the MIP-Net representation is less capable of separating relevant and irrelevant objects and the difference between MIP-DOI and the omniscient algorithm is greater.

While MIP-DOI-centrality does not perform well, integrating α (object centrality) with either β_2 (proximity to o_f, when o_f is known) or with β_1 (proximity to the partner, when o_f is unknown) leads to improved performance in early rounds, as objects that are more central are likely to have more short paths connecting them with other objects, and thus higher probability of being chosen for O_{modify}. This can be seen in the first row of Table 2. However, once sufficient information about specific objects and partners' roles is accumulated, integrating α results in lower precision (second row of Table 2).

Team members are likely to have more difficulty identifying relevant information about objects they interact with infrequently. Therefore, we examined the extent to which MIP-DOI is able to retrieve relevant objects that do not belong to partners' primary clusters. When using MIP-DOI-focus with $l = 3$, 72% of the objects in O_{share} were from outside of the partners' primary clusters. Using MIP-DOI-partner leads to less sharing of information from outside the primary cluster (50%), as the DOI focuses on distance from the partner's node. MIP-DOI-centrality shares the most information from outside the primary cluster (87%), but at the cost of sharing many irrelevant objects.

These analyses were based on a specific configuration of the simulation, but the general trends in performance were robust across different parameter configurations of the simulation. Figure **??** shows the aggregate precision@5 and recall@5 values over a range of simulation configurations (the tested parameter settings are specified in the Appendix), showing that all MIP-DOI configurations outperform the random baseline significantly ($p < 10^{-5}$), and that of the MIP-DOI configurations, MIP-DOI-focus achieves the highest precision and recall values. We next describe the effects of varying the simulation parameters on the performance of MIP-DOI, and provide additional results in the Appendix.

Team size: varying the number of partners ($|P|$) does not substantially affect the performance of MIP-DOI-focus. More objects are modified in each session, resulting in higher precision when using MIP-DOI-focus (there is a higher likelihood that a shared object will be relevant). Recall, however, does not increase because there are overall more relevant objects that change between two subsequent sessions of the same partner. The performance of MIP-DOI-partner degrades with increased team size, as it takes longer to learn the role allocation.



(a)

(b)

Figure 3: precision@5 (a) and recall@5 (b) aggregated over all parameter values examined. All MIP-DOI configurations significantly outperform the random baseline. MIP-DOI-focus performs best of the MIP-DOI configurations, followed by MIP-DOI-partner. Error bars show 95% confidence intervals.

Cluster size: increasing the number of objects in each cluster ($|Cl|$) leads to lower precision of all MIP-DOI configurations, as it takes the MIP-Net longer to capture the dependencies (constraints) between objects and the roles of partners.

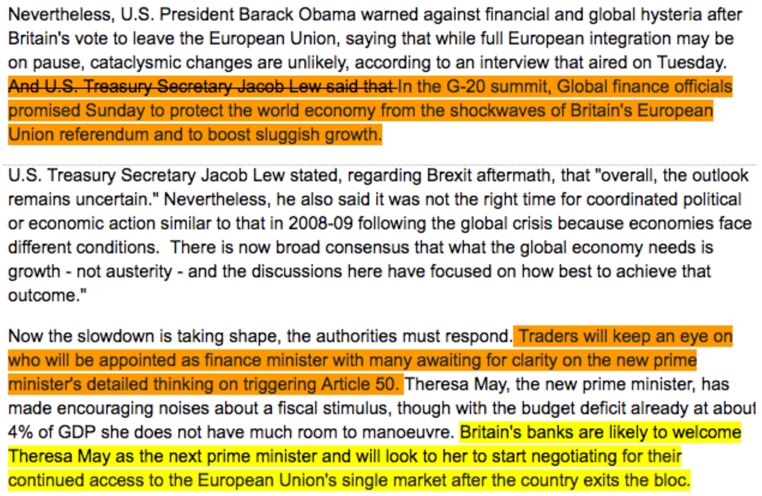
Number of modified objects: when increasing the number of objects a partner can change in a session (k), there are two effects: on the one hand, more information is incorporated in the MIP Update procedure (as more actions are taken). On the other hand, the relationship between pairs of objects is less indicative of constraints between them (e.g., there is a higher likelihood of choosing more distant objects to change together with o_f). Overall, the performance of MIP-DOI is similar across different values of k. Precision increases with k as there are simply more relevant objects, but recall does not.

Role allocation strictness: the strictness of role allocation is determined by $pr_{primary}$, that is, the probability that a partner chooses o_f from its primary cluster. The performance of MIP-DOI-partner is affected most by the changes to role allocation: with more strict role allocation (higher $pr_{primary}$), it is easier to capture the roles of different partners, and thus the proximity between object nodes and the partner node is more indicative of relevance. The other algorithms are not affected much by these changes. Their precision slightly decreases when increasing $pr_{primary}$ as fewer relevant objects change between each partner's consecutive sessions, but recall remains similar.

Graph structure: the parameters pr_{within} and $pr_{between}$ determine the likelihood of edges (constraints) connecting vertices in the same and in different clusters respectively. Generally, increasing both probabilities means that there are more edges in the graph, and thus more potentially relevant objects to share. Therefore, precision generally goes up with higher values of pr_{within} and $pr_{between}$, while recall does not. The exact effect depends on the specific values of these probabilities. $pr_{between}$ in essence controls the level of coupling between partners' activities. With smaller values of $pr_{between}$, it becomes harder for MIP-DOI to learn about the interactions of different partners' activities, and thus it becomes harder to share with a partner relevant information from *outside* that partner's primary cluster.

6. A Personalized Change Awareness Mechanism for Collaborative Writing

The results of the evaluation of MIP-DOI in a simulation demonstrated that it can efficiently and correctly model the relevance of actions of team members to each other. Next, we investigated the use of the algorithm in a realistic teamwork setting. In particular, we tested whether a personalized change awareness mechanism that uses MIP-DOI can improve collaboration outcomes while reducing coordination overhead in an actual collaborative scenario. To this end, we implemented a personalized change awareness mechanism in the context of collaborative writing. We used Google Docs as the framework for collaborative editing.



Nevertheless, U.S. President Barack Obama warned against financial and global hysteria after Britain's vote to leave the European Union, saying that while full European integration may be on pause, cataclysmic changes are unlikely, according to an interview that aired on Tuesday. ~~And U.S. Treasury Secretary Jacob Lew said that~~ In the G-20 summit, Global finance officials promised Sunday to protect the world economy from the shockwaves of Britain's European Union referendum and to boost sluggish growth.

U.S. Treasury Secretary Jacob Lew stated, regarding Brexit aftermath, that "overall, the outlook remains uncertain." Nevertheless, he also said it was not the right time for coordinated political or economic action similar to that in 2008-09 following the global crisis because economies face different conditions. There is now broad consensus that what the global economy needs is growth - not austerity - and the discussions here have focused on how best to achieve that outcome."

Now the slowdown is taking shape, the authorities must respond. Traders will keep an eye on who will be appointed as finance minister with many awaiting for clarity on the new prime minister's detailed thinking on triggering Article 50. Theresa May, the new prime minister, has made encouraging noises about a fiscal stimulus, though with the budget deficit already at about 4% of GDP she does not have much room to manoeuvre. Britain's banks are likely to welcome Theresa May as the next prime minister and will look to her to start negotiating for their continued access to the European Union's single market after the country exits the bloc.

Figure 4: Highlighted changes as shown in the experiment. Added text is highlighted using different colors for different authors. Deletions are marked with a strikethrough.

The personalized change awareness mechanism for collaborative writing shares with authors information about edits made by their co-authors. Similar to diff and track changes tools, the mechanism highlights the edits on the shared document as shown in Figure 4. In contrast to diff and track changes tools, the personalized change awareness mechanism highlights only a *subset* of the edits made to the document that are predicted to be most relevant to the particular author. By highlighting fewer edits, the mechanism aims to reduce information overload and focus the authors' attention on edits that are most likely to affect their own writing. The number of highlighted edits can be determined either by pre-specifying a "budget" l as in the ISLET problem definition (which can be an absolute number, or a certain percentage of the edits), or by specifying a threshold of degree-of-interest score.

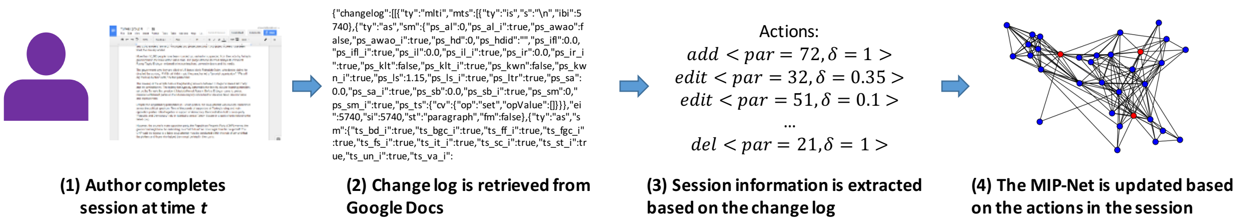
In the subsequent sections, we describe the mapping between the problem of personalizing the sharing of change information in collaborative writing to the ISLET problem, and the use of MIP-Nets and MIP-DOI in this domain.

6.1. Applying MIP-Nets and MIP-DOI in Collaborative Writing

In the context of collaborative writing, the ISLET setting comprises the following components:

- P: the authors of the document.
- O: the set of paragraphs in the document. Note that this set changes over time, as paragraphs are added and deleted.
- A: The act-types MOD, DEL and ADD correspond to modifying, deleting or adding paragraphs. To utilize additional domain-specific information, actions also describe the extent of change made to the paragraph, denoted δ, where $0 < \delta < 1$. That is, $mod(par, \delta)$ represents the action of editing an existing paragraph par with change extent δ; $add(par, 1)$ denotes the addition of a new paragraph par, and the extent of change is considered 1 as the entire paragraph was edited; $del(par, 1)$ denotes the deletion of an existing paragraph par from the document with the extent of change considered to be 1 as for added paragraphs.
- S: Interaction sessions: the session $s(p, t, (a_1, ..., a_k))$ consists of the edits made to the document by partner p when completing an editing session at time t.

We note that this formulation does not handle splitting and merging of paragraphs. In practice, if a paragraph is split into two, this will result in an added new node representing one of the two paragraphs resulting from the split (which will not incorporate the old weights), while a MOD action would be applied to the other paragraph (maintaining its node in the graph). Similarly, paragraph merging will result in a DEL action for one of the paragraphs and a MOD action for the other.



{"changelog":[[{"ty":"mlti","mts":[{"ty":"is","s":"\n","ibi":5740},{"ty":"as","sm":{"ps_al":0,"ps_al_i":true,"ps_awao":false,"ps_awao_i":true,"ps_hd":0,"ps_hdid":"","ps_ifl":0.0,"ps_ifl_i":true,"ps_il":0.0,"ps_il_i":true,"ps_ir":0.0,"ps_ir_i":true,"ps_klt":false,"ps_klt_i":true,"ps_kwn":false,"ps_kwn_i":true,"ps_ls":1.15,"ps_ls_i":true,"ps_ltr":true,"ps_sa":0.0,"ps_sa_i":true,"ps_sb":0.0,"ps_sb_i":true,"ps_sm":0,"ps_sm_i":true,"ps_ts":{"cv":{"op":"set","opValue":[]}}},"ei":5740,"si":5740,"st":"paragraph","fm":false},{"ty":"as","sm":{"ts_bd_i":true,"ts_bgc_i":true,"ts_ff_i":true,"ts_fgc_i":true,"ts_fs_i":true,"ts_it_i":true,"ts_sc_i":true,"ts_st_i":true,"ts_un_i":true,"ts_va_i":

Actions:

$add < par = 72, \delta = 1 >$

$edit < par = 32, \delta = 0.35 >$

$edit < par = 51, \delta = 0.1 >$

...

$del < par = 21, \delta = 1 >$

(1) Author completes session at time *t*

(2) Change log is retrieved from Google Docs

(3) Session information is extracted based on the change log

(4) The MIP-Net is updated based on the actions in the session

Figure 5: The process of updating a MIP-Net in the personalized change awareness mechanism for collaborative writing.

With the formal representation of the ISLET problem settings in the context of collaborative writing, MIP-DOI can be used to choose the subset of changes to share with each author. For simplicity, in the following we assume asynchronous editing, such that each revision is written by a single author, during a single editing session. The proposed approach can also handle synchronous editing. This can be done by defining sessions based on editing time (i.e., identify end of session of a particular user based on inactivity), and updating the MIP-NET based on each user's sessions.

6.1.1. Updating the MIP-Net Based on Editing Sessions

Figure 5 describes the process of updating the MIP-Net at the end of an author's editing session. Once an author completes her editing session, we retrieve a change log of the author's edits from Google Docs[3]. This change log describes all edits made to the document at the character level (e.g., insert e' in position 532). In our experiments, authors had limited time for each editing session and therefore the end of an interaction session was clearly determined when an author left the document. More generally, editing sessions can be segmented based on the authors' idle times. For example, if an author hasn't edited the document for over 15 minutes (based on the timestamps in the change log), we might consider the set of previous actions as a single session, and consider future actions as part of a new session.

Based on the change log, the system determines the updated text of each of the paragraphs. This is done by executing the actions from the change log (insertions and deletions of characters) on the latest version of the document that was saved. We identify the paragraphs that changed, and the extent to which those paragraphs were modified. This task is non-trivial because paragraphs can be moved, deleted and added over time. We used the method introduced by Gehrmann et al. [63], which computes pairwise similarities between paragraphs in the previous and current revisions to identify paragraphs that were moved.

Once paragraphs are matched, actions are identified by comparing the old and new versions of the same paragraph. The system extracts *add*, *del* and *mod* actions based on the pairwise mapping. For *add* and *del* actions we set $\delta = 1$. For *mod* actions, the extent of change is computed as follows:

$\delta = 1 - cosim(par^{t-1}, par^{t})$,

Where par^{t-1} is the text of the paragraph before the author's edits and par^{t} is the text of the paragraph after the edits and $cosim(par^{t-1}, par^{t})$ is the cosine similarity between the word vectors corresponding to the paragraph's content before and after the edit.

Once the session information (i.e., the sequence of editing actions) is extracted, the system updates the MIP-Net representing the collaborative activity as described in Section 4.2. When incrementing weights on edges, we consider the extent of change (δ) made to the paragraph, resulting in stronger connections between authors and paragraphs they substantially edited, and weaker connections to paragraphs for which they made minor changes such as fixing a typo. For example, if an author changes a paragraph substantially (e.g., $\delta = 0.4$), the weight between the node representing the session's author in the MIP-Net and the node representing the modified paragraph would be incremented by 0.4, while if only a typo was fixed (e.g., $\delta = 0.07$), the weight on that edge will not be affected substantially.

6.1.2. Sharing Change Information with Authors Using MIP-DOI

At the beginning of a new editing session, the current MIP-Net is used to determine the changes to share with the author who began this new session, who we refer to as the *current author*. The process is illustrated

[3] We used the process described in [62]

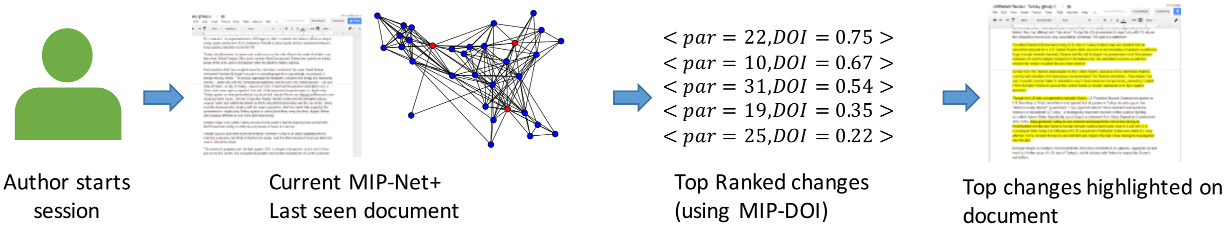


Figure 6: The process of determining which changes to share with a team member and presenting them to the current author.

in Figure 6. MIP-DOI is applied to the complete set of changes that were made since the last editing session of the current author. Therefore, the first step is to compare the text in the version of the document that was last seen by the author, with the text of the current revision. Based on this comparison the set of changed paragraphs is extracted. Next, the modified paragraphs are ranked using the MIP-DOI algorithm described in Section 4.3. Note that we do not know the focus object (o_f) because the system cannot tell which paragraph an author begins to edit.

We adapted the general MIP-DOI algorithm to make use of two types of additional information that is available in the writing domain. First, we consider the extent of the changes made to the paragraph. This information is incorporated to distinguish between minor edits (e.g., typo fixes) and more substantial edits. We filter the set of changes considered for sharing to include only those changes for which the change exceeds a threshold of 0.05.[4]

Second, we consider the proximity of the author not only to the changed paragraphs, but also to other paragraphs that appear in the same section of each of the changed paragraphs. By doing this, we utilize knowledge of the document structure to include more information about the authors' areas of responsibility and interests. That is, if an author changed some paragraphs in a certain section, we consider that as a signal suggesting possible interest in other paragraphs belonging to the same section.

In sum, we consider all paragraphs that have been changed since the current author's last editing session and which exceed the change extent threshold. We rank them using the following DOI computation:

$$\mathrm{DOI}(o \mid p) = \alpha \cdot API(n_o) + \beta \cdot D(n_o, n_p) + \gamma \cdot D(S_o, n_p) \tag{4}$$

where $D(S_o, n_p)$ is the proximity of the team member node to the nodes in the section that contains the modified paragraph, computed by averaging the proximity to each of the paragraphs in the section. We used $\alpha = 0.1, \beta = 0.8, \gamma = 0.1$ and a decay factor $\lambda = 0.9$. The values were determined empirically by testing MIP-DOI on a change prediction task using data from Wikipedia revision histories [19]. These values give a relatively low weight to the apriori importance of paragraphs, and assign most of the weight based on proximity between paragraphs in the MIP-Net.

Finally, the top ranked l changes are highlighted on the document, such that the author can review the changes before making her own contributions.

7. Empirical Evaluation of Personalized Change Awareness

To evaluate the personalized change awareness approach, we needed a teamwork task in which the activities of team members would be loosely coupled and in which coordination is challenging. We chose a collaborative writing task, which has been shown to have these characteristics [2, 5]. The particular writing task we chose was editing and updating summaries of prominent news stories. This editing task has a dynamic nature due to developments in the news story, and therefore requires iterated editing of the document as new information becomes available. We make this writing task loosely-coupled by assigning authors with different editing roles. While the main goal of each editor is to update the summary with new information, doing this in a way that maintains a coherent document requires awareness of others' edits.

[4] determined empirically using Wikipedia data

News story	Example news items
Brexit	UKIP leader Nigel Farage hailed it as the UK's "independence day", while Boris Johnson said the result would not mean "pulling up the drawbridge". (political editor)
	Britain's vote to leave the European Union has thrown financial markets into turmoil and means the U.S. Federal Reserve's ambitions for two rate rises this year have been placed on hold. (economics editor)
Turkey coup	Turkey's main opposition Republican People's Party (CHP) said the repose to a failed coup attempt must be conducted within the rule of law and that the plotters and those who helped them must be tried in the courts. (political editor)
	French President Francois Hollande said he expected there would be a period of repression in Turkey in the aftermath of a failed coup by some elements of the military. (foreign editor)
US Elections	More support for Clinton, as the race looks like it's over: Bernie Sanders endorses Democratic rival Hillary Clinton. She must become our next president', Sanders said in his statement. (democratic-party editor)
	U.S. Secret Service officials say security planning for the Republican national conventions took into account large-scale terrorism threats like the vehicle attack that occurred in France. (republican party editor)

Table 3: Examples of news items given to participants for each of the news stories.

We compared personalized change awareness to two baselines: one in which all changes were shown (which is the currently prevalent mechanism), and one in which only a subset of changes were shown (thus limiting the coordination overhead), but the changes were selected randomly. This design allowed us to separately investigate the impacts of the relevance and the quantity of change information shared with each team member. Since personalized change awareness mechanisms aim to reduce coordination overhead and improve the teamwork, we assessed the effect of change awareness mechanisms on participants' cognitive load and productivity (i.e., the amount of work completed), as well as the quality of the team's work, focusing on coordination-related aspects (e.g., conflicts in the documents). We further include measures to elicit participants' subjective perceptions of the helpfulness of the change awareness mechanisms.

7.1. Participants

We recruited 18 undergraduate students to participate in the study (ages 18–22, 11 females). Participants received $40 for completing the study. Participants were divided into six teams of three.

7.2. Task

News summaries. Participants in the study collaboratively edited summaries of three news stories (contemporary at the time of the experiment): *Brexit*, *the coup attempt in Turkey*, and *the 2016 US presidential elections primaries*. For each news story, participants were given an initial summary of the story (3000 words, 11 sections, 60 paragraphs), and were asked to expand the summary based on additional news items given to them in each session. We chose to start with an existing summary to make it non-trivial to identify all the edits made by co-authors from the start. The initial news summaries were adapted from summary articles taken from major news sources.

In each of their editing sessions, participants were given a list of news items to add to the summary. A few samples of these news items are given in Table 3. We gave participants the news items so that they would focus their efforts on incorporating the information into an appropriate place and ensuring it aligned with other parts of the summary, rather than on composing new prose or synthesizing information. This choice led to a controlled setting in which the team performance did not depend on the writing abilities of team members but rather on their ability to maintain an up-to-date and coherent document. For the most part, participants could simply paste each of the news item snippets into the summary, and update other places to avoid duplication of information or conflicts. In some cases, participants changed some of the wording to make the text fit better within an existing paragraph.

The news items were taken from short news pieces published in Associated Press or Reuters, in chronological order, such that updates which appeared in the general media at the same time tended to appear together in our lists as well. Participants were instructed to ignore their knowledge of the news story and update the summary based only on the information provided to them.

This task setup mimicked the kind of work that is done in a real news desk, where one of the authors had worked. In the news desk, journalists update a shared document and need to avoid redundancies and conflicts under strict time constraints.

Editing teams. Participants worked in 3-person editing teams. A team size of 3 had the essential coordination complexity in terms of participants not being able to easily keep track of everybody's edits, but was small enough that the entire task could be completed in a reasonable time frame for an experiment (over 7–10 days).

Each team member was assigned an editing role. As in real news desks, there were two types of editing roles, *specific* and *general*. Two members of each team were assigned *specific* editing roles. They were responsible for updating the summary with respect to a particular aspect of the news story. The *specific* roles were chosen based on the news story: political and economics editors for the Brexit story, political and foreign editors for the Turkey coup attempt story, and Republican-party and Democratic-party editors in the US elections story.

The *specific* editing roles meant that each of these editors' changes were, for the most part, focused on sections related to their primary area of responsibility. For example, the economics editor for Brexit made many of her edits in sections about implications of Brexit and membership in the European single market, while the political editor covered the process of electing a new prime minister. However, some news items were related to the areas covered by both editors (e.g., the election of a new prime minister affecting market performance), and therefore when one *specific* editor introduced such news items her edits often affected sections that were primarily edited by the other editor.

The third team member was assigned the *general* editor role, and was responsible for adding news items that were not yet added by the *specific* editors, enhancing coherence and enforcing a 3000 word limit. The *general* editor thus had to keep track of everyone's edits and edited sections related to all aspects of the story.

7.3. Conditions

Our experiment included three conditions, differing in the change awareness mechanism used to share change information with team members. The three conditions were:

1. *All*: all of the changes made to the document since the participant's last edit were highlighted.
2. *Random*: a random subset of up to 5 changes was highlighted.
3. *Personalized*: a subset of up to 5 changes, considered most relevant to the participant using MIP-DOI, was highlighted.

We included the *All* baseline to reflect current systems which show all the changes made to a document. This condition gives participants the possibility of reviewing all of the changes, but imposes more coordination overhead than mechanisms that limit the amount of shared information. Participants in the *Random* and *Personalized* conditions were presented with a similar (small) number of changes to review. We included the *Random* condition as a second baseline to evaluate the impact of the personalized change selection independent of the number of changes that had to be reviewed.

With both the *Random* and *Personalized* conditions, up to 5 changed paragraphs were highlighted, with the limit that at most two-thirds of the changed paragraphs should be highlighted. We did not make the number of highlighted changes entirely conditioned on the total number of changes to avoid creating substantial differences in the number of highlighted changes between these two conditions. We chose to restrict the number of highlighted changes to 5 as pilot studies determined the typical number of changes was 10–15, and we wanted to enforce substantial filtering.

In all conditions, the changes chosen to share were highlighted using the visualization style of Google Docs revision history view, which highlights modified text, marking deleted text with strikethrough. We used different highlighting colors for each of the editors, as shown in Figure 4.

7.4. Procedure

We used a within-subject design: each team edited one summary in each of the conditions: *Personalized*, *All* and *Random*. To this end, each team edited each of the three stories throughout the experiment, with each story being assigned with a different change awareness mechanism. Within each group, the assignment of a condition to a story persisted throughout (e.g., in group 1 *Personalized* was used for Brexit, *Random* for Turkey and *All* for elections; in group 2 *All* was used for Brexit, *Personalized* for Turkey and *Random* for elections). This enabled us to evaluate the final summary that was produced when a particular change awareness mechanism was used. The ordering of the conditions and the assignment of conditions to stories across groups were counterbalanced (we had 6 groups, covering the 6 possible combinations of condition-story

Activity	**Time**
Review news story 1	Round 1: 10 minutes; rounds 2-4: 30 seconds
Edit news story 1	12 minutes
Questionnaire news story 1	2 minutes
Review news story 2	Round 1: 10 minutes; rounds 2-4: 30 seconds
Edit news story 2	12 minutes
Questionnaire news story 2	2 minutes
Review news story 3	Round 1: 10 minutes; rounds 2-4: 30 seconds
Edit news story 3	12 minutes
Questionnaire news story 3	2 minutes
Final questionnaire	Only at last round, 10 minutes

Table 4: The activities in each of the editing sessions.

assignments). Each participant held a similar role (*specific* or *general* editor) in each of the three summary tasks.

We designed the procedure to reflect extended-duration collaborative writing and to control the amount of information and number of changes that were introduced in each revision. To this end, each participant participated in four rounds of editing. In each editing round, participants completed three editing sessions, one with each of the experimental conditions. To maintain a similar number of changes between each of the editors' sessions, the sessions were done in turns, with each editor returning to the document after the two other editors completed their sessions. The *general* editor was the last to edit in each round. Editing sessions were done remotely and were typically conducted one day apart from each other for each of the editors. The overall duration of the study for each team was 7–10 days during which all editors completed the four rounds of editing sessions, where in each of the rounds they edited all three stories. Due to scheduling constraints, one of the groups completed only three editing rounds.

The duration of each editing session was limited to 12 minutes. Participants who were assigned a *specific* editor role were given 10 items to add to the story. Participants who were assigned to the *general* editor role were given the two lists (of 10 items each) which were given to the two *specific* editors on their team, and were asked to add any missing items from the list and ensure they appear in appropriate places and do not conflict with other information. They were also asked to maintain a 3000 word limit, which meant they had to edit even if the *specific* editors managed to add all the news items during their sessions. The length of the editing sessions and the number of news items were determined based on pilot studies, with the goal of creating a task that would be hard to complete in the given time frame, while maintaining a reasonable session length (up to an hour for completing the task for all three summaries).

We recognize that co-authors in real-world settings have many ways to communicate, and a deployment of a personalized change awareness mechanism in the real-world would allow this. We did not allow participants in this study to communicate with each other or to use the commenting feature on the document, however, because we needed to ensure a controlled environment in which change awareness of team members did not depend on the particular communication strategies of team members and was manipulated only by our experimental conditions. Similarly, we did not allow for synchronous editing to ensure that the number of changes that occur between each author's editing sessions will be fairly similar.

Each editing session consisted of a cycle of three stages for each of the news stories, summarized in Table 4: (1) reviewing the current summary, (2) editing the summary, and (3) answering a questionnaire about their subjective experience.

In the **reviewing stage**, participants were first given time to read the current summary without ability to edit. In the first session, participants were given 10 minutes to read the current summary. In rounds 2–4, they were given only 30 seconds to review changes made to the document by their co-authors. The purpose of the short reviewing period was to draw participants' attention to the fact that their co-authors' had made changes. The time provided was intentionally short so that participants would not be able to review all changes before beginning their own editing session. Participants were still able to review changes later while editing the document.

In the **editing stage**, *specific* editors were given 12 minutes to incorporate 10 news items assigned to them based on their roles. They were instructed to add as many of the items as they could, while maintaining

Study Design Choices	Rationale
Within-subject design	Reducing the noise resulting from individual differences in writing skills, enabling participants to make comparisons between conditions.
Role allocation (editing roles)	Focusing on a loosely-coupled teamwork setting where team members have individual responsibilities but need to ensure their activities align.
Multiple sessions	Focusing on extended-duration teamwork where collaboration spans beyond a one-time interaction.
Limited editing time	Being able to compare outcomes across groups. ensuring reasonable experiment duration.
Asynchronous editing and turn-taking	Ensuring a similar number of edits to the article for authors returning to the document.
Adding news items to the summaries	Focusing on the coordination aspects of collaborative writing (e.g., avoiding redundancies and conflicts) as opposed to participants' writing skills.
No communication among team members	Ensuring that coordination outcomes are only affected by the experimental intervention (change awareness mechanism), focusing on testing the ability of the change awareness mechanisms to draw authors' attention to important edits.
Groups of size 3	Ensuring that there are substantial edits between each authors' sessions and non-trivial dependencies while maintaining a reasonable experiment duration (both for each session and for the time span of all sessions).

Table 5: Design choices made in the experimental setup and the rationale for each of the choices.

coherence (e.g., adding the information in an appropriate place) and avoiding conflicts and redundancies. For instance, when adding information about Boris Johnson deciding not to run for prime minister to the Brexit summary, the editor would also need to remove information suggesting that Johnson is the leading candidate in the race. During the editing stage, participants could still review the changes made by other team members. For the first three groups who participated in the study, the highlighted changes were shown on a separate, non-editable document, and participants could switch between editing and reviewing by clicking a button. Because many participants in these first groups commented that they would have preferred to have the changes highlighted on the document they were editing, for the remaining 3 groups the highlighted changes appeared on the document they were editing; they were instructed that they could remove the highlighting if they wished. We did not find differences between the earlier and later groups and therefore analyzed them together despite the difference in the way changes were presented.

The editing stage was followed by a **questionnaire** about their experience which is described below. The process of reviewing, editing and answering the questionnaire was repeated three times in each session, once for each news story.

After completing their last round of editing, participants were asked to answer a final questionnaire comparing the different change awareness mechanisms (referred to as "highlighting methods" when communicating with participants). Participants were provided access to the summaries with the highlighted changes and were asked (for each summary separately) to identify up to two highlighted changes that were relevant and helpful for them, and up to two highlighted changes that were irrelevant and unhelpful. They were asked to comment on the ways in which the relevant changes helped them. Participants were also asked to rank the highlighting methods based on their preferences. Because participants did not know which method was used for each news story, they were provided the associated story for each method (e.g. "method 1 (Brexit)").

The key design choices made in our experimental setup as well as the rationale for these choices are summarized in Table 5.

7.5. Design & Analysis

The dependent measures consisted of both objective and subjective measures related to participants' performance and experience. Participants provided subjective responses about the following aspects of the task at the end of each editing session for each summary in each round. All questions are shown in Table 6:

- Teamwork (e.g., "I was able to build on my co-authors' work").
- Helpfulness of the shared changes (e.g., "seeing the highlighted changes helped me ensure my edits align with others' edits").
- Workload assessment using the NASA TLX scale [64]. (e.g., "How mentally demanding was the task"; we omitted the physical demand question as it was irrelevant for the experimental task we used).

For teamwork and helpfulness of changes statements, participants rated their level of agreement on a 7-point Likert scale (1 = "strongly disagree", 7 = "strongly agree"). Workload questions (NASA TLX) were rated on a 7-point Likert scale (1 = "very low", 7 = "very high"). In the analysis of perceived workload, we considered the sum of all 5 items in the NASA TLX scale, which is common in the analysis of TLX measures [64].

Our dependent measures also included the following performance measures:

- Coverage: the number of news items that were added to the summary. We measured coverage at the session level (i.e., the specific contributions made by a single participant in a single session). Assessment of coverage was objective (was a particular news item incorporated into the summary or not) and was done by the authors. We measured coverage only for *specific* editors because the number of news items incorporated by *general* editors largely depended on the number of changes introduced by their team members.
- Quality: the quality of the final document in terms of consistency and coherency. We assessed the quality metrics for the final summary, which reflected the final product of the teamwork. Assessing the quality was done by comparing the final document to the initial document, and rating each paragraph that was added along the following dimensions (each rated on a scale from 1=poor to 3=good): (1) whether the text is redundant with other parts of the summary or conflicts with other information in the summary; (2) whether the paragraph is coherent with the remainder of the section; (3) whether it appeared in an appropriate section based on the section titles (e.g., information about May being elected should appear in the section about the race for the new prime minister), and (4) whether it appeared close to other related information. The reason for considering (3) and (4) separately was that in some cases an editor introduced text under an inappropriate section title, and later another editor placed related news updates nearby. If so, we wanted to penalize the new edit in terms of section location, yet acknowledge that it was rightfully located next to relevant information.

The quality assessment was done by two judges, with each summary being evaluated by a single judge, blind to condition. Both judges used the same assessment rubric, and they calibrated their ratings by evaluating one of the summaries independently and then comparing their evaluations. Their initial agreement was good (Krippendorff's alpha of 0.7) and they resolved any disagreements to be more aligned moving forward. One of the judges was an author who was not involved in running the study and who had worked in the past as a news editor. The other judge was an undergraduate student in the humanities. Because we were interested in comparing the performance within each of the groups, we were more concerned with having consistent rating for the three summaries that the same group edited. Therefore, each of the judges evaluated all 3 summaries (Brexit, Turkey coup, US elections) of 3 of the groups.

Analyses. Our analysis of subjective measures and of the objective coverage measure was done on the data from the last round of the study. We focused on the last round as we expected the personalized mechanism to take time to learn about participants' areas of responsibility. The quality assessment was done on the final summary as a whole, and therefore reflected the teamwork throughout all rounds.

We analyzed the Likert-scale items and the number of items added (between 1–10) using ordinal logistic regression [65]. The main effects we considered were the condition and the editor type, and we controlled for the topic of the news summary. We also included the participant id (unique identifier of each participant) as a covariate as the study was a within-subject design. Each participant id appeared three times in each analysis, once for each of the conditions. Thus, the model included the condition {All, Personalized, Random}, editor type {Specific, General}, topic {Brexit, Turkey coup, US elections}, participant id {1,...,18}.

Ordinal logistic regression estimates the likelihood of obtaining a *higher* value of a dependent measure given a change in the values of the independent variables. It is more appropriate for the analysis of Likert scale items than tests for comparing means since the ratings are given on an ordinal scale and are not normally distributed. The fitted model can be meaningfully interpreted by considering the *odds ratio* (OR) values of the regression coefficients of the independent variables[5], which we report in the Results section. For example, when estimating the effect of the study condition on participants' workload, the odds ratio value of *All* vs. *Personalized* was 8.8. This odds ratio value is interpreted as follows: all else being equal (i.e., same topic, participant and editor type), the odds of reporting a workload value that is *higher* than k (for any possible value of k) in the *All* condition are 8.8 times as large as the odds of reporting a value higher than k in the *Personalized* condition. That is, participants are much more likely to experience higher levels of workload when seeing all changes than with personalized change awareness.

To make interpretation easier, we always report odds ratio values greater than one. To do this, we take the multiplicative inverse value of odds ratios lower than 1, and interpret them as the odds ratio for obtaining a *lower or equal* value of the dependent measure (rather than a higher value). For example, if we get an odds ratio of 0.5 for the likelihood of obtaining a higher value, we convert it to an odds ratio of 2 for obtaining a lower or equal value. Odds ratios can be interpreted as effect size, similar to Cohen's d [66]. Values between 1.5 and 3 are interpreted as a small effect, between 3 and 5 as medium, and above 5 as large [67, 68].

To account for multiple hypotheses testing, we adjusted p-values with the Holm's sequentially rejective Bonferroni procedure, which introduces fewer Type II errors than the simple Bonferroni correction [69, 70]. We did this separately for subjective and objective measures. In post-hoc analyses we adjusted the p-values of pairwise comparisons with the simple Bonferroni correction. All of the p-values described in the Results section are the adjusted values.

Some Likert-scale items were intended to measure the same construct. We combined participants' responses to those items by summing their values if they had an acceptable Cronbach's alpha ($\alpha > 0.7$). This resulted in combining the responses to the items "seeing the highlighted changes helped me decide where to make my edits" and "seeing the highlighted changes helped me ensure my edits align with others' edits", and combining the responses to the items "I was overwhelmed by the changes to the document's content" and "Keeping track of my co-authors' edits was difficult."

The quality measures were also analyzed using ordinal logistic regression (scales of 1–3), and included as effects the condition, topic of the summary and the group that produced the summary[6]. We combined the four quality ratings (Cronbach's $\alpha = 0.83$) into a single quality measure by averaging the values.

Participants' final ranking of the change awareness mechanisms was analyzed using the nonparametric Friedman test with change awareness mechanism as the independent variable.

8. Results: Personalized Change Awareness in Collaborative Writing

Table 6 summarizes the results of the study. While ordinal logistic regression does not directly compare means, we show the mean values to illustrate the results. When a statistically significant difference was found between two conditions, we report the *odds ratio* values that quantify the difference between conditions.

Workload. We observed a significant main effect of condition on subjective workload (${\chi^2}_{(2,N=54)} = 14.8, p = 0.001$). Participants experienced significantly higher workload when seeing changes in the *All* condition compared to either *Personalized* ($OR = 8.88, \chi^2_{(1,N=54)} = 10.3, p = 0.006$) or *Random* ($OR = 9.04, \chi^2_{(1,N=54)} = 10.6, p = 0.003$) conditions. The effects sizes for both pairwise comparisons were large ($OR > 5$). We did not find a significant effect for editor type.

Teamwork-Related Items. We observed a significant main effect of condition on participants' subjective difficulty of keeping track with others' work and their feeling of being overwhelmed by changes (${\chi^2}_{(2,N=54)} = 8.3, p < 0.05$). Participants were more likely to report greater difficulty of keeping track of others' changes

[5] Odds ratio values are computed by exponentiating the regression coefficients, which estimate log odds ratios.

[6] We did not find an effect of the judge on the ratings, so did not include it in our model.

	Measure	**Personalized (Mean)**	**Random (Mean)**	**All (Mean)**	**Adjusted p-values**	**Significant pairwise differences**
Workload	TLX (5–35, 5 is best)	18.2 (4.6)	**17.8** (4.5)	21.6 (4.6)	**0.001**	P-A (OR: 8.9) R-A (OR: 9.0)
Teamwork measures	I was able to build on my co-authors' work with my edits. (1 – 7, 7 is best)	3.76 (1.3)	3.65 (1.4)	**4.12** (1.4)	0.46	-
	I was overwhelmed by the changes to the document's content; Keeping track of my co-authors' edits was difficult. (2–14, 2 is best)	7.29 (2.5)	**6.82** (2.8)	7.94 (3.0)	**0.05**	R-A (OR: 7.4)
Helpfulness of changes	Seeing the changes made by others helped me decide where to make my edits ; Seeing the changes made by others helped me ensure my edits align with others' edits. (2–14, 14 is best)	**7.59** (2.8)	7 (2.8)	7.47 (3.0)	0.49	-
	Seeing the changes made by others did not help me make my own edits . (1–7, 1 is best)	**3.12** (1.53)	4.29 (1.8)	4.06 (1.7)	**0.02**	P-A (OR: 6.5) P-R (OR: 6.6)
Preferences	Final preference ranking (1–3, 1 is best)	**1.83** (0.8)	2.06 (0.6)	2.06 (1.0)	0.77	-
Performance measures	Coverage (1–10, 10 is best)	**6.92** (2.5)	**6.92** (2.5)	5.67 (2.3)	**0.04**	P-A (OR: 4.8) R-A (OR: 6.5)
	Quality (1–3, 3 is best)	**2.8** (0.3)	2.75 (0.36)	2.78 (0.34)	**0.04**	P-R (OR: 1.7)

Table 6: The means and standard deviations (in parentheses) for each of the measures, and the Holm-Bonferroni adjusted p-value of the test for the significance of the condition effect. In cases where the differences were significant, the last column shows which pairwise comparisons of conditions were statistically significant (P-A indicates a significant differences between *Personalized* and *All*, R-A indicates a significant differences between *Random* and *All* and P-R indicates a significant differences between *Personalized* and *Random*), and the odds ratio (OR) of the difference. OR values between 1.5 and 3 are interpreted as a small effect, between 3 and 5 as medium, and above 5 as large

in the *All* condition compared to *Random* and *Personalized*. The only statistically significant pairwise comparison was between *All* and *Random* ($OR = 7.4, \chi^2_{(1,N=54)} = 7.5, p = 0.02$). We did not find significant differences in participants' reported ability to build on their co-authors work. We did not find a significant effect for editor type.

Helpfulness of the Shared Changes. With respect to the helpfulness of changes, we found a significant main effect of condition (${\chi^2}_{(2,N=54)} = 10.2, p = 0.024$). The shared changes were found more helpful in the *Personalized* condition compared to both the *All* ($OR = 6.5, \chi^2_{(1,N=54)} = 7.2, p = 0.02$) and *Random* ($OR = 6.6, \chi^2_{(1,N=54)} = 7.3, p = 0.02$) conditions. There was no effect for editor type on the helpfulness of changes.

To better understand the ways in which seeing changes helped participants accomplish their editing tasks, we examined participants' open-ended responses. At the end of the final editing session, participants were asked to copy up to 2 highlighted changes that were relevant and to explain how each of the relevant highlighted changes helped them make their edits. A common response among *specific* editors was that seeing others' edits helped them navigate the document, and that sometimes the highlighted edits addressed topics similar to some of their news items, and thus seeing where they were located helped them decide where to add new information. For example, one participants wrote "It helped me know where to place a couple of my items and allowed me to check that certain info was updated", while another participant commented "I saw them [the changes] as a context for the news items I received later."

General editors commented both about the helpfulness in terms of deciding where to place items and knowing which items were already added by their co-authors, e.g. "The first edit was helpful because it informed me where in the document a lot of the political edits were made (especially concerning Erdogan's 'crackdown') and seeing it also made me quickly aware that I did not have to add that piece of information, which was included in one of the lists."

Preference Rankings. The *Personalized* mechanism was ranked best ($M = 1.83$), followed by *All* and *Random* (both with $M = 2.06$), but this result was not statistically significant ($\chi^2_{(2,N=18)} = 0.5352, p = 0.765$).

Interestingly, we did not find different patterns in the ranking by *general* and *specific* editors. We expected that *general* editors may prefer to see more changes because their role required more awareness of the other editors' work, but observed that half of them ranked *All* as most preferred while the other half ranked it as least preferred.

Coverage and Quality. We observed a statistically significant effect of condition on the number of news items added by participants (${\chi^2}_{(2,N=36)} = 6.3, p = 0.04$). Participants added fewer items in the *All* condition compared to both *Personalized* ($OR = 4.8, \chi^2_{(1,N=36)} = 3.96, p = 0.04$) or *Random* ($OR = 6.5, \chi^2_{(1,N=36)} = 4.86, p = 0.03$). We did not observe an effect for editor type.

We also found a statistically significant effect of condition on the quality of the final summaries (${\chi^2}_{(2,N=719)} = 7.72, p = 0.04$). Summaries produced in the *Personalized* condition were significantly more likely to have higher quality rating than summaries produced in the *Random* condition ($OR = 1.69, {\chi^2}_{(1,N=719)} = 7.56, p = 0.006$). Other pairwise differences between conditions were not statistically significant.

We further examined what dimensions related to quality led to the differences in the aggregated quality ratings. In this analysis, we looked at the proportion of news items that were penalized by the judges, i.e., received a rating lower than 3, for each of the quality dimensions. This analysis was done using a logistics regression model which predicts the likelihood of a news item to be penalized. That is, the dependent variable was whether a paragraph was penalized, and the independent variables were the change awareness mechanism, the group that generated the summary and the topic of the document. Table 7 summarizes the proportion of news items that were penalized for each of the quality categories for documents produced using the different change awareness mechanisms.

Overall, 45% of the news items were penalized in summaries produced in the Random and All conditions, compared to 40% of the items penalized in summaries produced in the Personalized condition. The main differences in quality ratings resulted from the placement of news items in the summary. That is, whether the text appeared in an appropriate section based on the section titles and whether it appeared close to other related information (the average of the two "flow" measures). 29% of the items in the Random condition were penalized for their placement, compared to 22% for All and 19% for Personalized. We found a statistically

	Redundancy	**Conflict**	**Flow (section)**	**Flow (close)**
Personalized	0.07	0.33	0.11	0.15
All	0.07	0.39	0.12	0.19
Random	0.07	0.30	0.19	0.25

Table 7: The proportion of news items that were penalized for each of the quality measures when using the different change awareness mechanisms.

significant difference between Personalized and Random ($OR = 2.06, {\chi^2}_{(1,N=719)} = 8.78, p = 0.003$) and a marginally significant difference between All and Random ($OR = 1.54, {\chi^2}_{(1,N=719)} = 8.78, p = 0.06$).

8.1. Personalized Sharing of Change Information

To better understand the way in which personalization was reflected by the changes shared in the *Personalized* condition, we examined the MIP-Nets modeling the teamwork in the different groups. We illustrate the personalization of change sharing with examples of changes that were deemed relevant and irrelevant to different editors as this summary evolved.

Figure 7 shows the MIP-Net at the end of all rounds of editing of the Brexit summary produced by group 6. Author nodes are shown in red, with g', p' and e' representing the general, political and economics editors correspondingly. To make the network more compact, we collapse all nodes in a section to a single "section node" (blue nodes). The weights on edges connecting author and section nodes were computed by averaging the proximity of the author to each of the paragraphs in the section. The weights on edges connecting two sections were computed by averaging the proximity of all pairs of paragraphs in both sections. The intensity of edges connecting pairs of nodes reflects the weight of the edge (thicker edges reflect higher weights).

The loose-coupling of the task can be seen in the MIP-Net, in that each editor node is most strongly connected to nodes representing different sections. The political editor is most strongly connected with nodes representing sections 1 ("Supporters and opposers of leaving the EU"), 4 ("Is the kingdom still united?") and 5 ("A new conservative Prime Minister to be announced"), which focus on political aspects of Brexit. In contrast, the economics editor is most strongly connected with the nodes representing sections 6 ("Britain's economy awaits Brexit's aftermath") and 8 ("British taxpayer's money to be directed to NHS?") which focus on economical aspects of Brexit, and section 10 ("Other implications of Brexit"), which evolved to include economics related information during this group's edits (e.g., it included a news item about Obama warning against financial hysteria following the vote). While the focus of the *specific* editors was on different sections, note that the two *specific* editors also overlapped in editing other sections, hence their activities were not entirely decoupled. The proximity of the *general* editor to the various sections was more evenly distributed, which is expected as that editor was responsible for the overall summary.

By capturing each editor's unique focus, the MIP-Net enabled personalized assessment of relevance of edits to them. To illustrate, we consider the relevance assessments for two of the news items which were added to the summary by the *general* editor during her second round of editing:

1. *"Morgan Stanley sources said that it had started the process of moving about 2,000 staff based in London to either Dublin or Frankfurt. Ahead of the vote, the president of the investment bank, Colm Kelleher, told Bloomberg that Brexit would be 'the most consequential thing that we've ever seen since the war' "*
2. *"Germany's foreign minister says he hopes new British Foreign Secretary will want to maintain a 'sensible relationship' with the European Union."*

The first item was considered relevant for the *economics* editor. It had a high degree of interest because of its proximity to edits made by the economics editor; it appeared in the section "Britain's economy awaits Brexit's aftermath" (blue node #6 in Figure 7), which was substantially edited by the economics editor in previous editing sessions. This paragraph was rated much lower in terms of relevance to the political editor, as its proximity to that editor's edits was much lower. In contrast, the second item, which appeared in a section that discussed more of the political aspects of Brexit was considered relevant to the *political* editor, but not to the economics editor.

These two news items exemplify cases in which edits were fairly clearly related to either the economic or political aspects of the Brexit summary. However, some aspects of the story required more tight coupling of edits. For example, the section "Britain to stay in the single market?" (blue node #3) touched both on

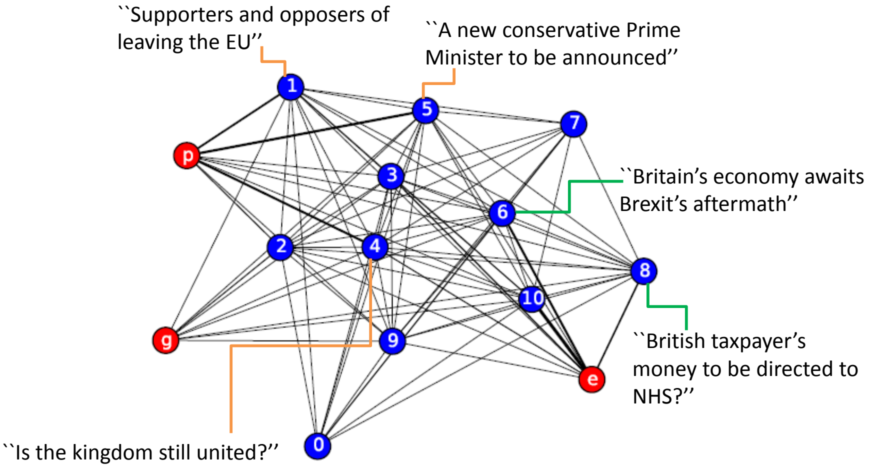

``Supporters and opposers of leaving the EU''
``A new conservative Prime Minister to be announced''
``Britain's economy awaits Brexit's aftermath''
``British taxpayer's money to be directed to NHS?''
``Is the kingdom still united?''
1
5
7
p
3
6
2
4
8
10
g
9
e
0

Figure 7: The final MIP-Net for the Brexit summary written by group 6, shown at the section granularity level. Author nodes are shown in red (e': economics editor, p': political editor, g': general editor), section nodes are shown in blue. The intensity of edges corresponds to their weights (darker edges have higher weight).

economy and politics related implications of Brexit, and therefore both editors had relatively high proximity to this section. Similarly, although the section "Britain's economy awaits Brexit's aftermath" (blue node #6) was primarily edited by the economics editor, the political editor occasionally edited it too, for instance when adding information about the G-20 summit which was already discussed in that section. In such cases, a news item added by one of the specific editors would have a relatively high degree of interest to the other specific editor, despite the differences in their areas of responsibility. We note that such dependencies take longer for the algorithm to learn about.

9. Discussion and Future Work

In this paper, we introduced the concept of *personalized* change awareness mechanisms, with the goal of reducing coordination overhead while still allowing team members to maintain awareness of activities of others that are relevant to their own activities. Such mechanisms can be particularly helpful in loosely-coupled teamwork, as team members' activities are relatively independent of each other, but where recognizing their occasional interactions can be difficult.

Providing personalized change awareness requires methods capable of reasoning about the relevance of information to different team members. To this end, we introduced the MIP-Nets representation which models collaborative activities, and the MIP-DOI algorithm which uses MIP-Nets to determine the relevance of information to different team members. We first evaluated MIP-DOI in a collaborative activity simulation, demonstrating its ability to identify information that is relevant to specific team members. We then designed and implemented a personalized change awareness mechanism for reducing the amount of shared change information in the context of collaborative writing.

The results of a controlled experiment, in which teams collaboratively edited news story summaries, demonstrated that the proposed Personalized Change Awareness approach can effectively support teamwork. While reducing the number of changes alone led to lower perceived workload, the helpfulness of changes was rated higher when the selection of changes was personalized rather than random. Furthermore, the final quality of the summaries produced with the *Personalized* change awareness mechanism was higher than with the *Random* change awareness mechanism. This result suggests that merely reducing the amount of change information shared with co-authors is not enough, and that identifying the relevant changes is crucial. When relevant changes were shown, participants were better able to align their edits with those made by their collaborators, leading to improved coherence of their documents. While more news items were added to summaries composed with the *Personalized* change awareness than to those produced with the *All* mechanism, the quality of the documents was similar, and thus the increased productivity did not hurt performance.

In summary, compared to the currently most prevalent approach of presenting users with all changes made by their collaborators, the developed Personalized Change Awareness mechanism resulted in significantly reduced subjective workload and significantly increased productivity without any detrimental effect on the quality of the work.

The personalized change awareness mechanism has several limitations, rooted in the limitations of the MIP-DOI algorithm. First, learning about the task structure requires observations of several interactions. Our study showed that the algorithm can learn meaningful information even with only a small number of interactions, but the learning rate will largely depend on the extent to which the task is structured and well-defined. Although the algorithm suffers from the "cold start" problem, we expect this problem to be alleviated by the fact that at the beginning of a teamwork activity the shared artifacts will have less content and therefore tracking changes will be easier. Tracking changes is likely to become much more difficult as the artifact evolves, and by then the algorithm will have more information to base its information sharing decisions on. Second, as the teamwork evolves the focus of different team members may change. To address this problem, we included decaying of weights in the procedure for updating the MIP-Net. However, determining the appropriate rate at which to decay weights may require using more sophisticated methods that handle "concept drift" [71], as well as designing interactions that would elicit more inputs from users and would help the system adapt.

Future work. In the study reported in this paper, personalized change awareness was done entirely automatically. We envision designing mixed-initiative interactions for personalized change awareness which will enable users to reveal more information about their goals (e.g., which section they plan to work on), and to provide the system with feedback about shared information. With these additional inputs, the system could adjust the algorithm to better match the users' interests and context.

Another direction we intend to explore is using information sharing algorithms to enhance team members' awareness of possible effects that their actions may have on other aspects of the team's work. For example, when the political editor of the Brexit story adds information about the opinions of the new prime minister regarding membership in the European single market, the system would alert her that this information might also need to be addressed in the section describing the economic implications of Brexit. This approach suggests a shift from providing only *retrospective* change awareness, to providing *prospective* awareness of the implications of changes.

Last, in our study design, we did not allow for communication between participants. In real collaborative writing settings (and in other collaborative activities), different types of information (explanations, task assignments, etc.) are shared between participants in various channels. We see interesting opportunities for incorporating personalized change awareness methods to these other communication channels as well.

Acknowledgements

This work was supported in part by NIH grant 1R01CA204585-01 as part of the NSF/NIH Smart and Connected Health program, and by a Siebel Scholarship. The authors thank Ezra Zigmond, Sebastian Gehrmann and Lauren Urke for contributing to the implementation of the collaborative writing personalized change awareness system, and Kobi Gal, Stuart Shieber, Stan Rosenschein and Bernd Huber for helpful discussions.

Appendix A. Additional Results: Varying Simulation Parameters

We report additional results from simulations varying the different parameter values. The parameters and their default values (which were used in the simulations reported in the main paper) are given in Table A.8. In the following, each analysis focuses on a single parameter, keeping constant the remaining parameter values (using their default values). Each analysis reports precision and recall values for three different communication budgets ($l = 1, l = 3, l = 5$). We denote precision at l by $p@l$ (e.g., $p@1$ means precision with communication budget of 1). Similarly, we use $r@l$ to denote recall at l. We report p-values based on t-tests, and controlling for multiple comparisons.

For each analysis, we also report the performance of the different algorithms ($p@5$ and $r@5$) relative to that of the Omniscient baseline, which provides an upper bound and thus allows comparison of the algorithms that is not affected by the absolute difficulty of identifying relevant information in a given parameter setting.

Parameter	Description
$\|P\|$	Number of partners [5]
$\|Cl\|$	Mean cluster size [10]
$pr_{primary}$	Probability o_f is chosen from the primary cluster [0.8]
pr_{within}	Probability of creating an edge between vertices in the same cluster [0.3]
$pr_{between}$	Probability of creating an edge between vertices in different clusters [0.05]
$k = \|O_{modify}\|$	Number of actions p can take in a single session [3]
$l = \|O_{share}\|$	Number of objects that can be shared in a single session [1,3,5]

Table A.8: The parameters controlling simulation configurations. The values in brackets were used in the experiments reported in the Results section. The communication budget l was varied.

Appendix A.1. Team Size

Figure A.8 shows precision@5 and recall@5 values obtained with the different MIP-DOI configurations and the Random baseline, relative to the performance of the Omniscient baseline (which captures an upper bound on performance), when varying the number of partners in the team (3, 5 or 7). Table A.9 provides more detailed results of the (absolute) performance of the algorithms.

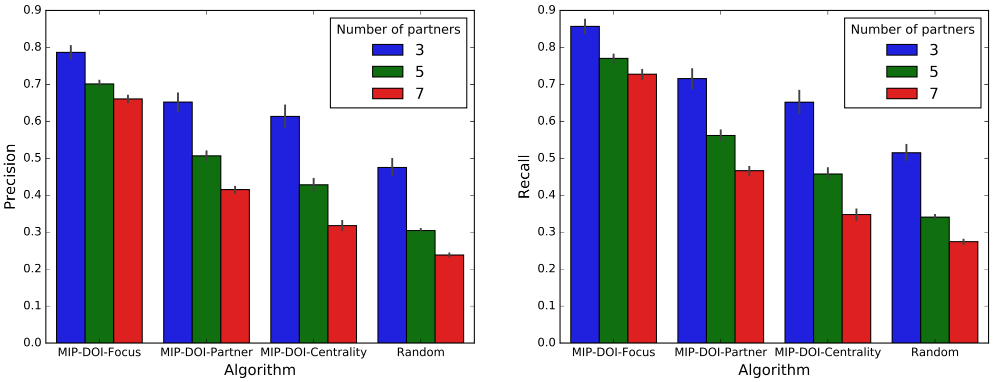


Figure A.8: Precision@5 (left) and recall@5 (right) relative to that obtained by the Omniscient baseline when varying the team size. Error bars show 95% confidence intervals.

The performance of the Omniscient and MIP-DOI-focus algorithms is stable across team sizes ($p > 0.3$). The reason for the higher precision values is that more objects change in the rounds between each partner's turns, such that the set of relevant objects is generally higher. However, recall is significantly lower (again, because the set of relevant objects is bigger). For MIP-DOI-partner, performance degrades as the team size grows ($p < 10^{-5}$ for both precision and recall) as it takes longer for the MIP-Net to learn the role allocation. The performance of MIP-DOI-centrality also degrades with increase in team size ($p < 10^{-5}$ for both precision and recall), since there are more objects.

	|P| = 3						**|P| = 5**						**|P| = 7**					
	p@1	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**	**p@1**	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**	**p@1**	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**
Omniscient	0.89	0.16	0.80	0.52	0.72	0.62	0.92	0.05	0.91	0.29	0.85	0.43	0.92	0.04	0.93	0.22	0.89	0.32
MIP-DOI-focus	0.81	0.23	0.61	0.46	0.57	0.53	0.83	0.09	0.72	0.27	0.59	0.33	0.85	0.07	0.73	0.21	0.62	0.25
MIP-DOI-partner	0.52	0.16	0.40	0.33	0.47	0.46	0.45	0.05	0.46	0.19	0.44	0.26	0.36	0.03	0.41	0.13	0.39	0.16
MIP-DOI-centrality	0.48	0.13	0.32	0.23	0.44	0.43	0.38	0.04	0.35	0.14	0.35	0.20	0.29	0.03	0.33	0.09	0.30	0.12
Random	0.28	0.10	0.23	0.17	0.31	0.31	0.28	0.03	0.27	0.12	0.27	0.15	0.21	0.02	0.23	0.07	0.22	0.09

Table A.9: Algorithm performance when varying the number of partners in the team.

Appendix A.2. Cluster Size

The results from simulations varying the cluster sizes are shown in Table A.10 and Figure A.9. As expected, the performance of MIP-DOI decreases with increasing cluster size, as it takes the MIP-Net longer to learn about each of the objects ($p < 10^{-5}$ for all precision and recall comparisons). The effect is particularly large for MIP-DOI-partner, as learning partners' role-allocation requires many more rounds.

	|Cl| = 5						**|Cl| = 10**						**|Cl| = 15**					
	p@1	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**	**p@1**	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**	**p@1**	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**
Omniscient	0.90	0.49	0.88	0.42	0.80	0.64	0.92	0.05	0.91	0.29	0.85	0.43	0.92	0.04	0.96	0.16	0.89	0.31
MIP-DOI-focus	0.90	0.57	0.75	0.42	0.67	0.60	0.83	0.09	0.72	0.27	0.59	0.33	0.80	0.06	0.64	0.12	0.59	0.23
MIP-DOI-partner	0.80	0.54	0.51	0.29	0.53	0.47	0.45	0.05	0.46	0.19	0.44	0.26	0.40	0.03	0.43	0.09	0.41	0.16
MIP-DOI-centrality	0.67	0.50	0.45	0.23	0.45	0.39	0.38	0.04	0.35	0.14	0.35	0.20	0.35	0.03	0.34	0.07	0.32	0.12
Random	0.53	0.46	0.28	0.15	0.32	0.28	0.28	0.03	0.27	0.12	0.27	0.15	0.25	0.02	0.26	0.06	0.25	0.10

Table A.10: Algorithm performance when varying the size of clusters assigned to partners.

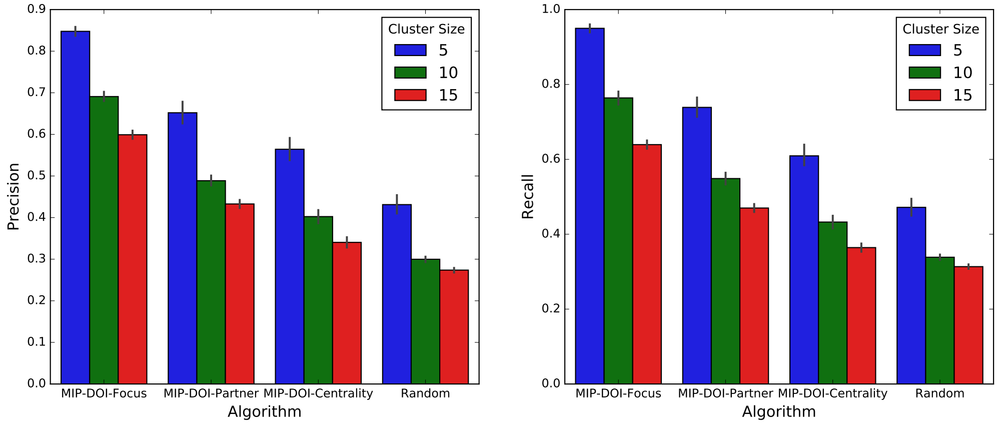


Figure A.9: Precision@5 (left) and recall@5 (right) relative to that obtained by the Omniscient baseline when varying the number of nodes in each cluster. Error bars show 95% confidence intervals.

Appendix A.3. Number of Modified Objects

When increasing k (the number of objects a partner can change in a session), there are two effects: on the one hand, more information is incorporated in the MIP Update procedure (as more actions are taken). On the other hand, the relationship between pairs of objects is less indicative of constraints between them (e.g., there is a higher likelihood of choosing more distant objects to change together with o_f). As can be seen in Table A.11, the overall performance of the algorithms is similar when using different values of k. Precision increases ($p < 10^{-5}$) as there are simply more relevant objects, but recall decreases for the same reason ($p < 10^{-5}$ for MIP-DOI-focus and MIP-DOI-partner, $p = 0.09$ for MIP-DOI-centrality) . Figure A.10 shows that when comparing the performance of the algorithms relative to the Omniscient baseline, performance is better when the number of modified objects grows. Again, this is because overall, more objects are relevant.

	k = 3						**k = 5**					
	p@1	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**	**p@1**	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**
Omniscient	0.91	0.19	0.92	0.29	0.85	0.46	0.99	0.07	0.98	0.24	0.94	0.40
MIP-DOI-focus	0.84	0.24	0.70	0.27	0.62	0.39	0.89	0.10	0.85	0.27	0.79	0.38
MIP-DOI-partner	0.55	0.21	0.47	0.19	0.46	0.30	0.63	0.09	0.65	0.21	0.63	0.32
MIP-DOI-centrality	0.47	0.19	0.38	0.15	0.37	0.24	0.49	0.07	0.46	0.14	0.56	0.28
Random	0.35	0.17	0.27	0.11	0.28	0.18	0.40	0.06	0.40	0.13	0.43	0.23

Table A.11: Algorithm performance when varying the size of O_{modify}.

Appendix A.4. Role Allocation

The strictness of role allocation is determined by $pr_{primary}$, that is, the probability that a partner chooses o_f from its primary cluster. Table A.12 shows the performance of the algorithms in simulations using the values 0.4, 0.6 or 0.8 for $pr_{primary}$. The performance of MIP-DOI-partner is affected most by the changes to role allocation (also shown in Figure A.11): with more strict role allocation (higher $pr_{primary}$), it is easier to capture the roles of different partners, and thus the proximity between object nodes and the partner node is more indicative of relevance ($p < 10^{-5}$ for both precision and recall). The other algorithms are not affected

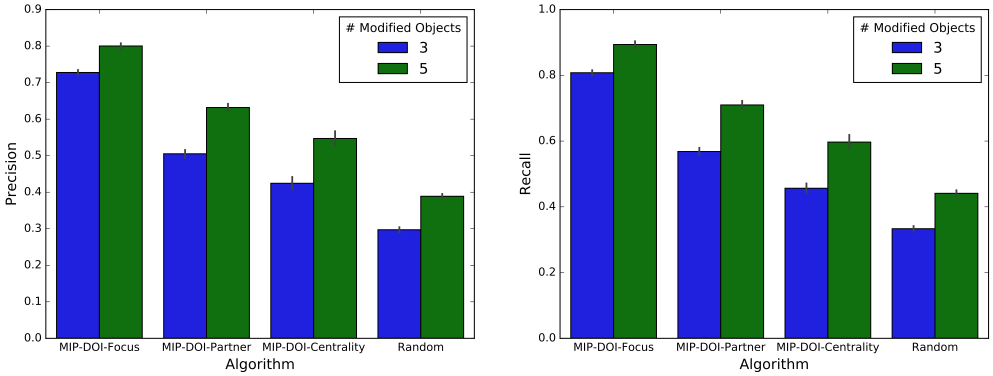


Figure A.10: Precision@5 (left) and recall@5 (right) relative to that obtained by the Omniscient baseline when varying the number of modified objects. Error bars show 95% confidence intervals.

much by these changes (MIP-DOI-centrality: $p > 0.1$ for both precision and recall for all comparisons; MIP-DOI-partner: comparison to $p_{primary} = 0.8$ results in significantly higher precision for lower $p_{primary}$ values, $p < 10^{-5}$). Their precision slightly decreases with $pr_{primary} = 0.8$ as less relevant objects change between each partner's consecutive sessions, but recall remains similar.

	$\mathbf{pr_{primary}} = \mathbf{0.4}$						$\mathbf{pr_{primary}} = \mathbf{0.6}$						$\mathbf{pr_{primary}} = \mathbf{0.8}$					
	p@1	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**	**p@1**	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**	**p@1**	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**
Omniscient	0.89	0.02	0.94	0.23	0.90	0.39	0.91	0.07	0.94	0.28	0.86	0.42	0.92	0.05	0.91	0.29	0.85	0.43
MIP-DOI-focus	0.87	0.08	0.78	0.25	0.64	0.32	0.87	0.12	0.76	0.28	0.66	0.36	0.83	0.09	0.72	0.27	0.59	0.33
MIP-DOI-partner	0.37	0.04	0.40	0.12	0.34	0.17	0.42	0.07	0.40	0.16	0.37	0.21	0.45	0.05	0.46	0.19	0.44	0.26
MIP-DOI-centrality	0.39	0.04	0.41	0.12	0.32	0.15	0.44	0.07	0.37	0.14	0.32	0.18	0.38	0.04	0.35	0.14	0.35	0.20
Random	0.28	0.03	0.26	0.08	0.26	0.13	0.30	0.06	0.26	0.10	0.26	0.15	0.28	0.03	0.27	0.12	0.27	0.15

Table A.12: Algorithm performance when varying the strictness of role allocation.

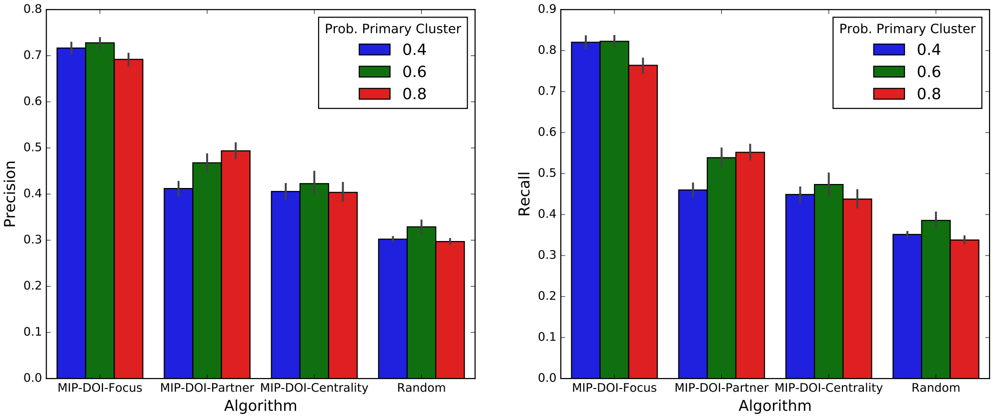


Figure A.11: Precision@5 (left) and recall@5 (right) relative to that obtained by the Omniscient baseline when varying the probability of selecting an object from the primary cluster. Error bars show 95% confidence intervals.

Appendix A.5. Graph Structure

The parameters pr_{within} and $pr_{between}$ determine the likelihood of edges (constraints) connecting vertices in the same and in different clusters respectively. Table A.13 shows the performance of the algorithms in simulations with three different combinations of values for these parameters. Generally, increasing both probabilities means that there are more edges in the graph, and thus more potentially relevant objects to share. Therefore, as expected, precision generally goes up with higher values of pr_{within} and $pr_{between}$ while recall does not. The exact effect depends on the specific values of these probabilities.

$pr_{between}$ in essence controls the level of coupling between partners' activities. With smaller values of $pr_{between}$, it becomes harder for MIP-DOI to learn about the interactions of different partners' activities, and thus it becomes harder to share with a partner relevant information from outside that partner's primary cluster. For all MIP-DOI configurations, precision increased while recall decreased when increasing $p_{between}$ from 0.05 to 0.01 ($p < 0.001$). When increasing p_{within} from 0.3 to 0.4 (while keeping $p_{between}$ the same), there was a significantly higher precision for MIP-DOI-centrality and MIP-DOI-partner ($p < 10^{-5}$), without a significant change in precision ($p > 0.5$). For MIP-DOI-focus there weren't statistically significant differences in either precision and recall.

	$\mathbf{pr_{within} = 0.3, pr_{between} = 0.05}$						$\mathbf{pr_{within} = 0.3, pr_{between} = 0.1}$						$\mathbf{pr_{within} = 0.4, pr_{between} = 0.1}$					
	p@1	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**	**p@1**	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**	**p@1**	**r@1**	**p@3**	**r@3**	**p@5**	**r@5**
Omniscient	0.92	0.05	0.91	0.29	0.85	0.43	0.91	0.08	0.93	0.22	0.83	0.45	0.93	0.04	0.97	0.16	0.93	0.30
MIP-DOI-focus	0.83	0.09	0.72	0.27	0.59	0.33	0.82	0.12	0.68	0.19	0.62	0.37	0.83	0.07	0.69	0.13	0.64	0.22
MIP-DOI-partner	0.45	0.05	0.46	0.19	0.44	0.26	0.53	0.08	0.52	0.15	0.43	0.26	0.50	0.04	0.51	0.10	0.50	0.18
MIP-DOI-centrality	0.38	0.04	0.35	0.14	0.35	0.20	0.50	0.08	0.41	0.11	0.32	0.19	0.35	0.03	0.43	0.08	0.43	0.15
Random	0.28	0.03	0.27	0.12	0.27	0.15	0.27	0.05	0.27	0.08	0.26	0.16	0.36	0.03	0.39	0.08	0.39	0.14

Table A.13: Algorithm performance when varying the graph structure (edge probabilities).

When comparing the performance of the algorithms to the Omniscient baseline (Figure A.12), we observe the MIP-DOI-focus actually maintains performance closer to the upper bound when p_{within} and $p_{between}$ are lower, while MIP-DOI-centrality and MIP-DOI-partner are closer to the upper bound for higher values of these variables.

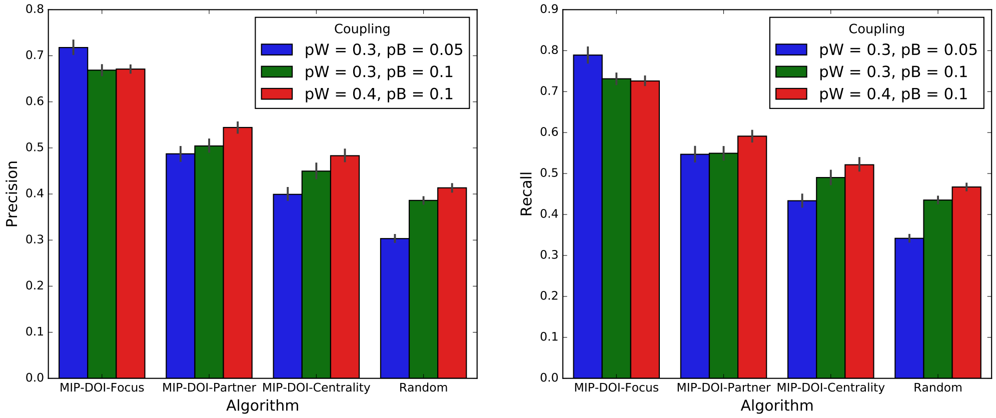


Figure A.12: Precision@5 (left) and recall@5 (right) relative to that obtained by the Omniscient baseline when varying the graph structure ($p_{between}$ and p_{within}). Error bars show 95% confidence intervals.

References

[1] O. Amir, B. J. Grosz, K. Z. Gajos, S. M. Swenson, L. M. Sanders, From Care Plans to Care Coordination: Opportunities for Computer Support of Teamwork in Complex Healthcare, in: CHI'15, 2015.

[2] J. S. Olson, S. Teasley, Groupware in the wild: Lessons learned from a year of virtual collocation, in: Proceedings of the 1996 ACM conference on Computer supported cooperative work, ACM, 419–427, 1996.

[3] E. Hutchins, Cognition in the Wild, MIT press, 1995.

[4] B. Grosz, S. Kraus, Collaborative plans for complex group action, Artificial Intelligence 86 (2) (1996) 269–357.

[5] P. B. Lowry, A. Curtis, M. R. Lowry, Building a taxonomy and nomenclature of collaborative writing to improve interdisciplinary research and practice, Journal of Business Communication 41 (1) (2004) 66–99.

[6] J. Tam, S. Greenberg, A framework for asynchronous change awareness in collaborative documents and workspaces, International Journal of Human-Computer Studies 64 (7) (2006) 583–598.

[7] P. Dourish, V. Bellotti, Awareness and coordination in shared workspaces, in: Proceedings of the 1992 ACM conference on Computer-supported cooperative work, ACM, 107–114, 1992.

[8] M. P. Steves, E. Morse, C. Gutwin, S. Greenberg, A comparison of usage evaluation and inspection methods for assessing groupware usability, in: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work, ACM, 125–134, 2001.

[9] D. Pinelle, C. Gutwin, Loose coupling and healthcare organizations: deployment strategies for groupware, Computer Supported Cooperative Work (CSCW) 15 (5-6) (2006) 537–572.

[10] L. Leape, Order From Chaos: Accelerating Care Integration, National Patient Safety Foundation, 2012.

[11] M. Roth, R. Simmons, M. Veloso, What to communicate? Execution-time decision in multi-agent POMDPs, Distributed Autonomous Robotic Systems 7 (2006) 177–186.

[12] O. Amir, B. J. Grosz, R. Stern, To Share or Not to Share? The Single Agent in a Team Decision Problem, in: Models and Paradigms for Planning under Uncertainty: a Broad Perspective, 2014.

[13] F. S. Melo, M. T. Spaan, S. J. Witwicki, QueryPOMDP: POMDP-based communication in multiagent systems, in: Multi-Agent Systems, Springer, 189–204, 2012.

[14] F. Wu, S. Zilberstein, X. Chen, Online planning for multi-agent systems with bounded communication, Artificial Intelligence 175 (2) (2011) 487–511.

[15] V. V. Unhelkar, J. A. Shah, ConTaCT : Deciding to Communicate during Time-Critical Collaborative Tasks in Unknown, Deterministic Domains, in: AAAI, 2016.

[16] G. Best, M. Forrai, R. R. Mettu, R. Fitch, Planning-aware communication for decentralised multi-robot coordination, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 1050–1057, 2018.

[17] R. J. Marcotte, X. Wang, D. Mehta, E. Olson, Optimizing multi-robot communication under bandwidth constraints, Autonomous Robots (2019) 1–13.

[18] M. Fowler, P. Tokekar, T. C. Clancy, R. K. Williams, Constrained-Action POMDPs for Multi-Agent Intelligent Knowledge Distribution, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 1–8, 2018.

[19] O. Amir, B. Grosz, K. Z. Gajos, Mutual Influence Potential Networks: Enabling Information Sharing in Loosely-Coupled Extended-Duration Teamwork, in: Proceedings of IJCAI'16, 2016.

[20] P. Cohen, H. Levesque, Intention is choice with commitment, Artificial intelligence 42 (2) (1990) 213–261.

[21] D. Kinny, M. Ljungberg, A. Rao, E. Sonenberg, G. Tidhard, Planned team activity. Artificial Social Systems, Lecture notes in AI 830, C. Castelfranchi and E. Werner, editors, 1992.

[22] M. Tambe, Agent Architectures for Flexible Practical Teamwork, AAAI 97 (1) (1997) 997.

[23] D. Weerasooriya, A. Rao, K. Ramamohanarao, Design of a concurrent agent-oriented language, in: Intelligent Agents, Springer, 386–401, 1995.

[24] A. Pokahr, L. Braubach, W. Lamersdorf, Jadex: A BDI reasoning engine, in: Multi-agent programming, Springer, 149–174, 2005.

[25] P. Scerri, E. Liao, J. Lai, K. Sycara, Y. Xu, M. Lewis, Coordinating very large groups of wide area search munitions, in: Theory and Algorithms for Cooperative Systems, World Scientific, 451–480, 2004.

[26] G. A. Kaminka, I. Frenkel, Integration of coordination mechanisms in the BITE multi-robot architecture, in: Robotics and Automation, 2007 IEEE International Conference on, IEEE, 2859–2866, 2007.

[27] G. A. Kaminka, I. Frenkel, Flexible teamwork in behavior-based robots, in: Proceedings Of The National Conference On Artificial Intelligence, vol. 20, 108, 2005.

[28] C. V. Goldman, S. Zilberstein, Optimizing information exchange in cooperative multi-agent systems, in: Proceedings of the second international joint conference on Autonomous agents and multiagent systems, ACM, 137–144, 2003.

[29] D. V. Pynadath, M. Tambe, The communicative multiagent team decision problem: analyzing teamwork theories and models, Journal of Artificial Intelligence Research 16 (1) (2002) 389–423.

[30] M. T. Spaan, G. J. Gordon, N. Vlassis, Decentralized planning under uncertainty for teams of communicating agents, in: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, ACM, 249–256, 2006.

[31] P. Xuan, V. Lesser, S. Zilberstein, Communication decisions in multi-agent cooperation: Model and experiments, in: Proceedings of the fifth international conference on Autonomous agents, ACM, 616–623, 2001.

[32] F. A. Oliehoek, M. T. Spaan, N. Vlassis, Dec-POMDPs with delayed communication, in: The 2nd Workshop on Multi-agent Sequential Decision-Making in Uncertain Domains, 2007.

[33] R. Emery-Montemerlo, G. Gordon, J. Schneider, S. Thrun, Game theoretic control for robot teams, in: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, IEEE, 1163–1169, 2005.

[34] M. Roth, R. Simmons, M. Veloso, Reasoning about joint beliefs for execution-time communication decisions, in: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, ACM, 786–793, 2005.

[35] F. Wu, S. Zilberstein, X. Chen, Multi-Agent Online Planning with Communication., in: ICAPS, 2009.

[36] J.-y. Kwak, R. Yang, Z. Yin, M. E. Taylor, M. Tambe, Robust execution-time coordination in DEC-POMDPs under model uncertainty, in: Sixth Annual Workshop on Multiagent Sequential Decision Making in Uncertain Domains (MSDM-2011), 39, 2011.

[37] C. Zhang, V. Lesser, Coordinating multi-agent reinforcement learning with limited communication, in: Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, International Foundation for Autonomous Agents and Multiagent Systems, 1101–1108, 2013.

[38] S. Barrett, P. Stone, Cooperating with unknown teammates in complex domains: A robot soccer case study of ad hoc teamwork, in: AAAI, 2015.

[39] G. A. Kaminka, D. V. Pynadath, M. Tambe, Monitoring Teams by Overhearing: A Multi-Agent Plan-Recognition Approach, Journal of Artificial Intelligence Research 17 (2002) 83–135.

[40] O. Amir, K. Gal, Plan Recognition and Visualization in Exploratory Learning Environments, ACM Transactions on Interactive Intelligent Systems (TiiS) 3 (3) (2013) 16–1.

[41] P. Stone, G. A. Kaminka, S. Kraus, J. S. Rosenschein, et al., Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination., in: AAAI, 2010.

[42] O. Amir, B. J. Grosz, E. Law, R. Stern, Collaborative health care plan support, in: Proceedings of the 12th international conference on Autonomous agents and multi-agent systems, 793–796, 2013.

[43] W. Prinz, E. Hinrichs, I. Kireyev, Anticipative awareness in a groupware system, in: From CSCW to Web 2.0: European Developments in Collaborative Design, Springer, 3–20, 2010.

[44] C. M. Neuwirth, R. Chandhok, D. S. Kaufer, P. Erion, J. Morris, D. Miller, Flexible diff-ing in a collaborative writing system, in: Proceedings of the 1992 ACM conference on Computer-supported cooperative work, ACM, 147–154, 1992.

[45] J. Wuertz, S. A. Alharthi, W. A. Hamilton, S. Bateman, C. Gutwin, A. Tang, Z. Toups, J. Hammer, A design framework for awareness cues in distributed multiplayer games, in: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, ACM, 243, 2018.

[46] R. Padhye, S. Mani, V. S. Sinha, NeedFeed: taming change notifications by modeling code relevance, in: Proceedings of the 29th ACM/IEEE international conference on Automated software engineering, ACM, 665–676, 2014.

[47] R. Holmes, R. J. Walker, Customized awareness: recommending relevant external change events, in: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1, ACM, 465–474, 2010.

[48] I. Omoronyia, J. Ferguson, M. Roper, M. Wood, Using developer activity data to enhance awareness during collaborative software development, Computer Supported Cooperative Work (CSCW) 18 (5-6) (2009) 509–558.

[49] J. Froehlich, P. Dourish, Unifying artifacts and activities in a visual tool for distributed software development teams, in: Proceedings of the 26th International Conference on Software Engineering, IEEE Computer Society, 387–396, 2004.

[50] C. R. de Souza, S. Quirk, E. Trainer, D. F. Redmiles, Supporting collaborative software development through the visualization of socio-technical dependencies, in: Proceedings of the 2007 international ACM conference on Supporting group work, ACM, 147–156, 2007.

[51] M. R. Jakobsen, R. Fernandez, M. Czerwinski, K. Inkpen, O. Kulyk, G. G. Robertson, WIPDash: Work item and people dashboard for software development teams, in: IFIP Conference on Human-Computer Interaction, Springer, 791–804, 2009.

[52] C. De Souza, J. Froehlich, P. Dourish, Seeking the source: software source code as a social and technical artifact, in: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work, ACM, 197–206, 2005.

[53] C. Gutwin, K. Schneider, D. Paquette, R. Penner, Supporting group awareness in distributed software development, in: International Workshop on Design, Specification, and Verification of Interactive Systems, Springer, 383–397, 2004.

[54] Y. Koren, R. Bell, Advances in collaborative filtering, in: Recommender systems handbook, Springer, 145–186, 2011.

[55] P. Lops, M. De Gemmis, G. Semeraro, Content-based recommender systems: State of the art and trends, in: Recommender systems handbook, Springer, 73–105, 2011.

[56] D. Sperber, D. Wilson, Precis of relevance: Communication and cognition, Behavioral and brain sciences 10 (04) (1987) 697–710.

[57] G. W. Furnas, Generalized fisheye views, vol. 17, ACM, 1986.

[58] F. Van Ham, A. Perer, Search, Show Context, Expand on Demand: Supporting Large Graph Exploration with Degree-of-Interest, Visualization and Computer Graphics, IEEE Transactions on 15 (6) (2009) 953–960.

[59] L. A. Adamic, E. Adar, Friends and neighbors on the web, Social networks 25 (3) (2003) 211–230.

[60] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, Journal of the American society for information science and technology 58 (7) (2007) 1019–1031.

[61] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, D. Damian, The promises and perils of mining GitHub, in: Proceedings of the 11th Working Conference on Mining Software Repositories, ACM, 92–101, 2014.

[62] How I reverse engineerd Google Docs, `http://features.jsomers.net/how-i-reverse-engineered-google-docs/`, accessed: 2018-02-35, 2018.

[63] S. Gehrmann, L. Urke, O. Amir, B. J. Grosz, Deploying AI Methods to Support Collaborative Writing: a Preliminary Investigation, in: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems, ACM, 917–922, 2015.

[64] S. G. Hart, NASA-task load index (NASA-TLX); 20 years later, in: Proceedings of the human factors and ergonomics society annual meeting, vol. 50, Sage Publications, 904–908, 2006.

[65] F. E. Harrell, Ordinal logistic regression, in: Regression modeling strategies, Springer, 311–325, 2015.

[66] M. E. Rice, G. T. Harris, Comparing effect sizes in follow-up studies: ROC Area, Cohen's d, and r, Law and human behavior 29 (5) (2005) 615–620.

[67] M. Borenstein, L. V. Hedges, J. Higgins, H. R. Rothstein, Converting among effect sizes, Introduction to meta-analysis (2009) 45–49.

[68] H. Chen, P. Cohen, S. Chen, How big is a big odds ratio? Interpreting the magnitudes of odds ratios in epidemiological studies, Communications in StatisticsSimulation and Computation® 39 (4) (2010) 860–864.

[69] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics 6 (65-70) (1979) 1979.

[70] J. P. Shaffer, Multiple Hypothesis-Testing, Annual Review of Psychology 46 (1995) 561–584.

[71] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, Machine learning 23 (1) (1996) 69–101.