Activity-Aware Recommendation For Collaborative Work in Operating Rooms

ABSTRACT

This paper presents a recommender system for teams of medical professionals working collaboratively in hospital operating rooms. The system recommends relevant virtual actions, such as retrieval of information resources and initiation of communication with professionals outside the operating rooms. Recommendations are based on the current state of the ongoing operation as recognised from sensor data using machine learning techniques. The selection and non-selection of virtual actions during operations are interpreted as implicit feedback and used to update the weight matrices that guide recommendations. A pilot user study involving medical professionals indicates that the adaptation mechanism works effectively and that the system provides adequate recommendations.

Author Keywords

Information Retrieval, Collaborative Work, Hospitals, Context-Aware, Filtering, Adaptation, Recommendation, Groups, Workplaces

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces—*Evaluation/methodology*; J.3 Computer Applications: Life and Medical Science—*Medical information systems*

General Terms

Algorithms, Design, Experimentation

INTRODUCTION

Context-aware information retrieval refers to systems that help providing resources that are relevant to the current situation of users. In many work environments, carrying out a task demands massive information access and management, and at the same time sharing and coordinating this information with co-workers. Context-aware computing in such settings can reduce the need for manual information retrieval and assist users in bringing forward relevant resources and hence decreasing the time and effort needed to manage and update the information.

So far, research in context-aware computing has mainly focused on adapting the device and systems to individual users' needs or behaviour [**?**, **?**, **?**, **?**, **?**]. The system detects different patterns of a user's work and adapts the user interface or the information resources accordingly. However, in a collaborative work environment such as a hospital, adaptive information retrieval goes beyond covering the needs of individuals, as medical cases often involve several users working concurrently sharing systems, devices, and documents. This type of work involves standard procedures but at the same time varies significantly from case to case depending on the patients' health conditions. One way of handling this issue can be building a decision-making or a rule-based system which learns different procedures and common practices and acts according to the situation. The problem with this approach is that the medical work includes many variations that are sometimes unpredictable and hence difficult to be addressed with specific rules.

In this study, we propose an innovative application of recommendation for information adaptation in shared settings. By incorporating collaborative filtering and content-based techniques, we dynamically provide most relevant types of information that has been used in similar situations. The main contributions of this approach is an adaptation mechanism that:

1. incrementally builds and updates its model of relevant information for every situation based on similar past situations, and
2. incorporates users' information choices as implicit feedback and retrains the model to provide more satisfactory information assistance.

More specifically for the medical domain, this approach contributes to:

1. Improving creative and assistive technology for the extensive and time consuming process of accessing information from different medical systems.
2. Covering the demand of consultation among clinicians by giving access to similar cases as well as help updating medical guidelines and procedures with actual best practices used in different cases.

Our approach emphasises activities being performed in the physical environment as essential contextual information, and uses such information for more precise information adaptation.

We assume that the hospital in which the system is installed is a smart environment with numerous embedded public displays around the building. We specifically focus on work in the operating rooms (ORs), as surgical procedures are performed by a team of clinicians who are co-located in one room but have different information need. This means that the information resources presented on an e.g., wall display

Submitted for review to IUI 2012.

in the OR should adapt according to several people's requests at the same time.

The paper starts with related research in supporting context-aware adaptation followed by a study of surgical procedures in the hospitals. We then describe the details of the process and the meta algorithm in recommending information resources, and finally, we present the experimental evaluation of the approach.

RELATED WORK

Activity-centric systems for personal computers (used in the office or at home) monitor and maintain knowledge about the users' desktop activities, and provide assistance for their ongoing tasks [**?**, **?**, **?**]. By monitoring the stream of context events, such as opening documents and keyboard use, these systems help find relevant resources such as other documents or related folders.

Moving beyond traditional personal computers, ubiquitous context-aware systems use different types of contextual information, such as location, to show relevant information on mobile or embedded devices [**?**, **?**, **?**]. For example, in a location-aware guide system, information about the places the user is visiting is automatically shown on his mobile device [**?**]. Examples of such systems are GUIDE [**?**] and COMPASS [**?**]. Similarly, context-aware systems in so-called Smart Spaces adapt to changes in the environment by e.g., turning on the light when a person enters the room [**?**].

Recommender systems are developed to recommend the most relevant items to individual users based on the items they have shown interest in and their similarity with other users. Traditional recommender systems do not take into consideration any contextual information, such as time, place and the company of other people (e.g., for watching movies or dining out). In other words, they deal with applications having only two types of entities, users and items, and do not put them into a context when providing recommendations. This approach is widely used by Amazon, Yahoo, and Netflix. The most applied recommendation methods in such systems are item-based recommendation and collaborative filtering [**?**].

In contrast to these systems which are centred around individual user preferences, group recommendation approaches consider the interest of a group of people. Examples include MusicFX [**?**] which chooses a radio station for playing music in a fitness centre that suits a group of people working out at a given time; or INTRIGUE [**?**] that recommends places to visit for groups of tourists based on the characteristics of the group members such as children or elderlies.

The mentioned studies define and use context based on the application of the system. In desktop-based applications, e.g., CAAD [**?**], or in web-based recommender systems, e.g., News@hand [**?**], the context is digitally obtained from the user's data or interaction history. In contrast, mobile systems, e.g., Cyberguide [**?**] obtain the context mostly from the physical world, e.g., the user's location. None of these approaches infers user's activities in the physical world to be used as context in the digital world, but this is what we find essential for more precise information adaptation.

Moreover, while all these studies focus on users and provide items or information based on their profiles, interests, and interactions, our research is centred around tasks in which users participate and their activities are considered as context.

In this paper, we outline how we address these requirements with our proposed activity-aware recommendation approach.

STUDIES OF SURGICAL PROCEDURES

Our research is rooted in long-term studies of hospital work. Specifically, a series of detailed, video-recorded observations of more than 25 surgical operations have been conducted in two hospitals, combined with numerous semi-structured interviews. Our general observations of surgical operations reveal that in a typical surgery at least 6 clinicians with different specializations participate. The team includes at least one anesthesia nurse mainly responsible for patient monitoring during the operation, an anesthesiologist, a surgeon, a surgical assistant, a surgical nurse assisting with instruments, and a circulating nurse for general help and communication between inside and outside of the room. Based on what distinguishes surgical actions from each other and allows them to be detected individually, we have identified a set of 17 main tasks which we refer to as physical actions (pa). These actions are listed in Table 1. A surgical activity follows a temporal and sequential pattern:

The procedure usually starts with the anesthesia team preparing devices (pa_1), drugs and instruments (pa_2, pa_3) followed by preparation of the patient for anesthetization (pa_4). While the patient is being anesthetized (pa_5), the surgical instruments and devices are prepared by surgical- and circulating nurses (pa_6). After the patient is anesthetized and intubated (pa_7), his body is prepared (pa_8) for the incision process (pa_9). During the surgical execution (pa_{10}), the patient's condition is monitored (pa_{11}) by the anesthesia nurse. Before the procedure is finished, the surgical instruments are gathered (pa_{12}) and the patient is prepared for waking up (pa_{13}). Closing the cut (pa_{14}) means that the instruments and devices can be removed from the patient's body (pa_{15}). The operation is considered as ended when the patient is extubated (pa_{16}) and is transferred to the recovery room (pa_{17}).

Surgical Actions

We have identified three types of actions in a surgical activity:

- Actions that are common in all types of surgeries and are carried out in certain phases of the operation. These actions are prerequisites for initiating other actions. For instance, anesthetization always takes place during the preparation phase and before the surgery starts. If the patient or the specific area of his/her body is not anesthetized, the surgery cannot be started.
- Actions that happen to be performed in some surgeries, and their occurrence is in particular phases of the surgery. For example, the intubation is not necessary for all types of surgeries, but if the patient should be intubated, it will be conducted before the incision process.

Action	*Label*
pa_1	Checking the anesthesia machine
pa_2	Anesthetic preparation
pa_3	Anesthesia-instrument preparation
pa_4	Patient preparation for anesthesia
pa_5	Surgical-instrument preparation
pa_6	Anesthetization
pa_7	Intubation
pa_8	Preparing the patient for operation
pa_9	Incision
pa_{10}	Main procedure
pa_{11}	Patient monitoring
pa_{12}	Collecting surgical instruments
pa_{13}	Waking the patient
pa_{14}	Closing
pa_{15}	Cleaning up
pa_{16}	Extubation
pa_{17}	Recovery ready

Table 1. The list of physical actions in an operating room

- Actions that are carried out in some surgeries without being bound to a certain phase. For example, the nurse anesthetist can order the blood either before, during, or after finishing the surgery.

Some actions do not follow a specific order. For instance, it makes no difference if the anesthetist checks the devices before or after preparing the medicine. The only thing that matters is that both medicine and devices are ready before the patient enters the OR. *These variations in the order and number of surgical actions indicate that providing relevant information to medical staff requires the system being able to recognize physical actions in the OR.*

The surgical team members collaborate to carry out an overall task which entails performing a series of actions in parallel. Some of these actions can be done by only one clinician, such as monitoring the vital signs of the patient during surgery, while others involve several people, like the surgical procedure which at minimum involves the surgeon and the assisting surgical nurse.

We also investigated the type of information that was needed, retrieved, or discussed by clinicians. Our three main findings are: (i) most information and documents are related to the patient's medical record; (ii) work procedures are standardized and exists in the form of guidelines and checklists; and (iii) there is a huge demand for consultation, discussion and research in case of complications and emergency in patient cases.

THE OVERALL APPROACH

Activity-aware recommendation in multi-user shared setting of an OR is supported in an overall architecture containing a Context Acquisition Component, a Physical-Action Recognition Component, a Virtual-Action Discovery Component, and a Recommender Surface as presented in figure 2. The contextual information, e.g., location of a person acquired from sensors or other sources is used for recognition of physical actions (pas) performed by clinicians (ps). The collection of recognized pas is used by Virtual-Action Discovery component which finds and adjusts a set of relevant virtual actions (vas) to be presented on the Recommender Surface.

Figure 1. An example of showing relevant surgical information on large wall displays inside an operating room

Before entering the technical details of the approach, we describe a scenario that presents a situation during a surgical procedure to which a wall display in the operating room adapts (see e.g., figure 1).

The surgical nurse (p_1) is preparing the instruments (pa_5) while the anesthesia nurse (p_2) and the anesthetist (p_3) are anesthetizing (pa_6) the patient. The device detects the situation and the display adapts the anesthesia guideline to show the anesthetization step (va_1), updates the list of instruments detected on the operating trolley (va_2), and keeps showing x-ray images (va_3) related to that patient on the screen. 10 minute later, the patient is anesthetized; the surgical instruments are ready for use; and the patient's body is prepared for the operation (pa_8). The surgeon (p_4) and his assistant (p_5) enter the room and the incision (pa_9) starts. The system recognizes that anesthesia guideline (va_1) is no longer relevant to this situation, checking surgical instrument (va_2) is less relevant, x-ray images (va_3) are still relevant and patient monitoring (va_4) is highly relevant from now on. The display changes so that va_1 disappears from the screen, va_2 becomes passive, va_3 remains activated and va_4 appears on the screen.

Physical Action Recognition

In many situations, simple contextual elements such as location and people's identity can be used to filter unnecessary information. For example, if the location is an operating room, then only the information related to surgical operations can be relevant there. However, in order to provide relevant information inside the OR (where the location remains unchanged), the system needs to recognize participants' physical actions.

The general process in activity recognition is to model the activities, learn the model from training data, and then use

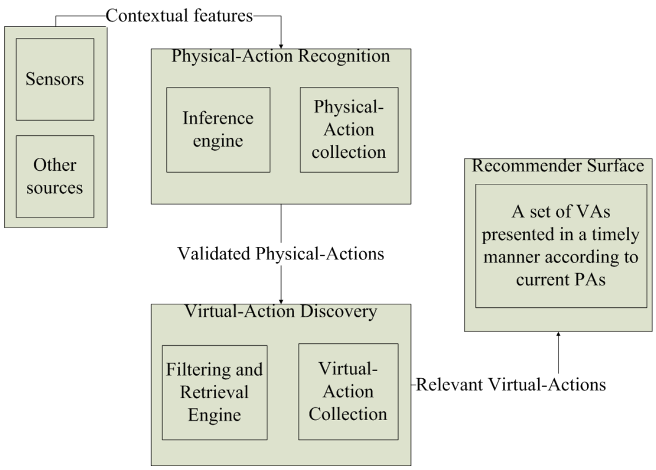

Figure 2. Overall architecture for supporting context- and activity-awareness

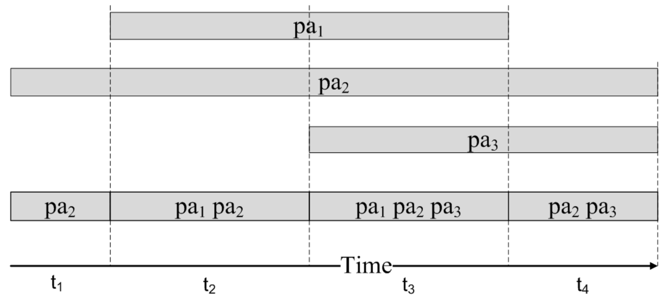

Figure 3. Distinct combinations of multiple actions observed at each fragment of time are called joint-actions.

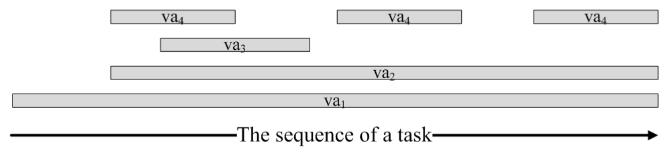

Figure 4. Different cases of information relevance in the sequence of a task

the learned model to infer activities from new data instances. We apply machine learning techniques on the data obtained from sensors to recognize concurrent actions spread over time and space. Since collaborative work involves several activities either done by one person (multi-tasking) or by multiple collaborating users in a concurrent manner, the recognition mechanism should handle multiple actions occurring in the environment. According to Allen [**?**], the interval relations including *before*, *meets*, *overlaps*, *starts*, *during*, *finishes*, and *equals* describe any relative positioning of two intervals. In case of multiple pas, we face the same set of co-occurrence situations. As shown in figure 3, at time t_1, only pa_2 is observed, where in t_2 both pa_1 and pa_2 occur. We define the notion of *joint-actions* (ja) as a block of multiple pa_ks observed in a fragment of time t. In figure 3, there are four distinct combinations of pas and hence 4 jas, namely pa_2 (ja_1), pa_1pa_2 (ja_2), $pa_1pa_2pa_3$ (ja_3), and pa_2pa_3 (ja_4).

The process of learning joint-actions in details is out of the scope of this paper and is presented in [**?**]. This paper assumes that at each step of the surgery, a set of physical actions are recognized by the activity recognition component. The rest of the paper is focused on the techniques used to find and recommend relevant virtual actions to a set of physical actions during a collaborative work situation such as a surgical procedure.

Virtual Action Adaptation

Adaptation mechanism responds to each detected pa with a set of *virtual actions* (va) which are associated with a collection of information resources and services (r). In other words, let $R = \{r_j, 0 \leq j \leq m\}$ be the set of information resources. Then $VA = \{va_i, 0 \leq i \leq I\}$ is the set of virtual actions where $\forall pa, \exists va$, where $va \subseteq VA$, and $va_i = \{r, r \subset R\}$. An example of a virtual action is showing and updating the patient's medical record (r_1) with information obtained from monitoring devices (r_2) during the operation.

Each task can be composed of several steps spread in different spaces. This means that even when changing the situation, a virtual action (va) which has been relevant in the last situation can still be 1) highly relevant, 2) less relevant, or 3) irrelevant. Hence, during the sequence of a task, a va can (figure 4):

- be relevant from the beginning to the end (va_1), e.g., access to a patient medical record remains relevant during the surgical procedure.
- become relevant in one step (based on a situation) and remain relevant to the end (va_2), e.g., monitoring a patient becomes relevant after anesthetization of the patient for surgery and remains relevant until the operation ends.
- be relevant only in a step of the task and become irrelevant after the situation changes (va_3), e.g., ordering blood in case of emergency.
- become relevant in some situations (on and off) (va_4), e.g., an instrument checklist becomes relevant during the surgical preparation and again when cleaning up after the operation is finished.

Having several users engaged in multiple concurrent physical actions (pa)s indicates that in each situation, a collection of vas will be relevant. A list of different cases for content interest is shown in figure 5. For a pa, only one va (1) or several different vas (2) can be relevant. Distinct vas can be relevant for distinct pas (5), or some vas can be of common interest for distinct pas (3, 4).

Similar situations in the physical setting entails similarity in the information need and thus similarity in adaptation mechanism. For example, the treatment process for patients with similar diseases is more or less the same involving resembled information requests. Although there are differences in new

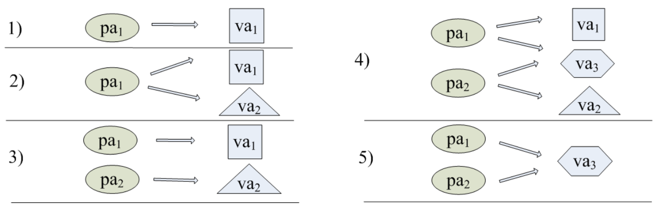

Figure 5. Different cases of content interest. For a pa, only one va (1) or several different vas (2) can be relevant. Several pas can have separate interests in vas (5), or they can share some interests (3, 4).

situations, past patterns can still suggest an initial collection of vas that might be of interest in the new situation.

This implication suggests considering similar past situations as sources that can be used for initialization of the va collection while a change in the situation occurs. However, the set of suggestions should be filtered so that the vas are customized to current situation and task. Each situation can either have been experienced or have existed in form of known procedures containing rules and practices. We, therefore, use three sources in the adaptation algorithm with regard to an entity, such as a task:

1. *Profile* (Pr) – all information that relates to a particular entity, e.g., information about a task, its location, duration, participants, etc. In other words, $Pr = (at_1...at_n)$ is the set of attributes of an overall task. For example, the profile of a surgical operation includes information about the disease, operation type, duration, patient's age and gender.

2. *Community* (C) – The C source is the set of all other entities of the same genre that share the same profile structure, e.g., other surgical operations.

3. *Domain Knowledge* (D)– The D source is all knowledge about an entity that can be trusted, e.g., the set of rules and procedures to perform different surgeries.

The primary set of vas is created from a co-occurrence matrix which keeps track of relevance weights of the vas in accordance to pas. The relevance weight is the probability of action va_i while occurring pa_j and is adjusted according to physical actions and users' interactions with vas.

Hence, during each task, there are two adaptation steps as presented in figure 6. At step A, the initial set of potentially relevant vas is collected by applying similarity measures using three introduced sources (Pr, C, and D). The profile is used as a pre-filtering source to further identify the community and domain knowledge source. During the process, those vas whose resources relate to procedures and domain knowledge (e.g., guidelines) are customized by looking at the D source, while other vas with content related to similar cases are customized in C.Step B is dynamic where the presented collection of the vas from step A is updated based on changes in e.g., current pas. Change in the set of pas, triggers the adaptation algorithm to update the collection of presented vas. Each va can have a different relevance weight to a different pa.

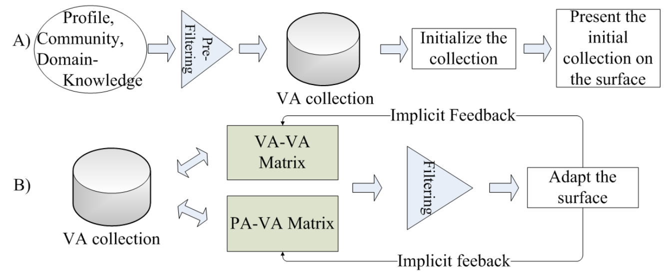

Figure 6. Adaptation steps for virtual actions related to each situation

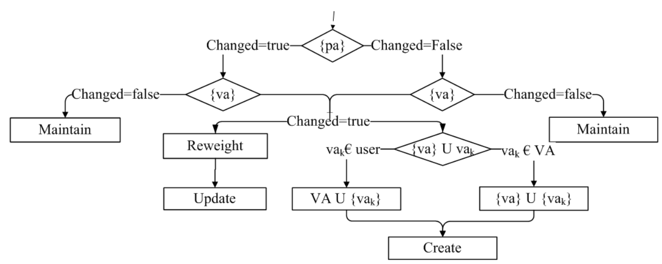

Figure 7. The flow of system actions based on different combinations of physical and virtual actions at each time step.

The weight of vas is updated by incorporation of *implicit feedback* that is given to the presented collection on the surface by e.g., the users. In this case, the implicit feedback mainly includes visible interactions with the vas, e.g., in form of a click. The feedback updates two co-occurrence matrices $pa - va$ and $va - va$. The former keeps relevance weights between pa and vas and the latter updates co-occurrence weights for pairs of vas.

Our design integrates the collaborative filtering concept [**?**] which suggests that the interest of a user can be predicted by collecting information from many other users. We generalize this definition by saying that the behaviour of an entity (human or object) can be predicted by collecting behaviour patterns of similar entities. In this case, what 'interests' a situation in terms of relevant vas might also have been of interest in other past situations. Hence, the algorithm finds relevant vas by looking at past situations using entity's profile as the filtering vector, and then, it presents them on the surface as a set of recommendations for current situation. The users' interactions with the suggested vas indicate whether the presented set has been of any interest or use in the situation. The generated set is then updated based on the feedback.

RECOMMENDATION PROCESS

The state of vas can be '*created*', '*updated*', or '*maintained*' according to changes in the situation (figure 7).

Creation of a va occurs in both steps when it is attached to the current task for the first time. After pre-filtering, the primary set of vas is created from matrices. A va is also created during the second step either by the users in form of a complete new collection of resources or from an existing set of

vas which is not currently presented by the recommender.

The *update* of the suggested collection occurs during the online process in two cases:

1. When the set of pas has changed, it might cause suggestion of a new collection of vas. This new collection can be completely different from the previous one, have some common vas with different or same weights, or have identical set of elements with different weights.
2. After suggesting the collection, the weights of presented vas is changed via implicit feedback from the user.

A va remains in the *maintenance* state when its relevance is unchanged despite changes in the set of pas.

Finding and suggesting relevant virtual actions ($\{va\}$) in a time series of physical actions ($\{pa\}_t$) is done as described in algorithms 1 to 5. As mentioned, we use two matrices to keep weights. one is

$$M_{pa-va} = [m_{ij}], i = 0...|PA|, j = 0...|VA| \ , \quad m_{ij} = w_{va_j|pa_i}$$

where w is the current relevance weight between pa_i and va_j. The other one is the co-occurrence matrix for virtual actions defined as

$$M_{va-va} = [n_{ij}], i = 0...|VA|, j = 0...|VA| \ , \quad n_{ij} = w_{va_i,va_j}$$

where w is the current co-occurrence weight between va_i and va_j.

Initialization

The pre-filtering process is conducted in the *initialization* step (algorithm 1). The set of top-n similar candidates in C and top-m relevant candidates in D with least distance to current situation and task are chosen using a weighted similarity metric applied on profile vectors. The weight indicates the importance of an attribute and empowers the filtering process. For instance, the value of the disease attribute is a stronger filter for irrelevant operations than the gender of the patient. As each attribute might be of a different type, the distance values are normalized for further computation.

Filtering and Adaptation

In the beginning, i.e., at time $t = 0$, the set of vas generated by the initialization algorithm is presented to the users on the surface. This collection is then adapted based on changes in the set of pas observed at each fragment of time. In the filtering step (algorithm 2), the recommended set of items is updated by finding the top-k weighted va candidates in M_{pa-va} and then choosing different top-k vas in M_{va-va} matrix which are different from the set that was already chosen. The value of k is either an arbitrary constant or is chosen using a function.

Implicit Feedback Incorporation

The weight of recommended virtual actions is updated based on the users' interactions with the suggested vas (e.g., in form of clicks), and each use is considered a positive feedback which increases the relevance weight of the corresponding va (algorithm 3). The weights are then updated in both matrices and the filtering process is repeated.

Algorithm 1 Initialization

set $pr_{sP} = (at_1, ..., at_n)$ /* profile vector
set $pw_{sP} = (pw_1, ..., pw_n)$ /* priority vector
for all $c_i \in C$ **do**
  $dist(pr_{sp}, pr_{c_i}) = (d_{i1}, ..., d_{in})$ /* distance vector
  normalize $d_{i1}, ..., d_{in}$
  add priority weights (pw_{sp}) to $dist(pr_{sp}, pr_{c_i})$
  calculate $dist_{sp,c_i} = \sqrt{\sum_{j=1}^{n} d_{ij}}$
  set $sim_{sp,c_i} = \frac{1}{1+dist_{sp,c_i}}$
  choose top-n c_is with $max(sim_{sp,c_i})$
end for
for all $e_j \in D$ **do**
  $dist(pr_{sp}, pr_{e_j}) = (d_{j1}, ..., d_{jn})$ /* distance vector
  normalize $d_{j1}, ..., d_{jn}$
  add priority weights (pw_{sp}) to $dist(pr_{sp}, pr_{e_j})$
  calculate $dist_{sp,e_j} = \sqrt{\sum_{k=1}^{n} d_{jk}}$
  set $sim_{e_j} = \frac{1}{1+dist_{e_j}}$
  choose top-m candidates e_js with $max(sim_{e_j})$
end for
initialize M_{pa-va}, M_{va-va} with $\{c_i\}$ and $\{e_j\}$ collection

Algorithm 2 Filtering

let R be the set of recommended vas
if $t \neq 0$, *set* $R = \emptyset$
set $PA_t = \{pa_k, 0 \leq k \leq |PA|\}$ /* at time t
for all $pa_i \in PA_t$ **do**
  find $top - k$ values for m_{ij} in M_{pa-va}, $m_{ij} = w_{pa_i|va_j}$
  add va_js to R
end for
In M_{va-va} *find* $top - k$ values for n_{ij}, $n_{ij} = w_{va_i,va_j}$
if $va_i, va_j \notin R$ **then**
  add va_i and va_j to R
else
  goto 8
end if
suggest R on the surface

Updating Matrices

The process of updating two matrices M_{pa-va} and M_{va-va} is as follow (algorithm 4): Let $PA_t = \{pa_k, 0 \leq k \leq |PA|\}$ be the set of physical actions being performed at time t and $VA_t = \{va_s, 0 \leq s \leq |VA|\}$ be the set of virtual actions suggested at time t.

While updating M_{pa-va}, for each $pa_i \in PA_t$ and $va_j \in VA_t$, $m_{ij} = m_{ij} + \alpha$, where α is an aggregate value or in the simplest case a constant, e.g., $\alpha = 1$. In M_{va-va}, $\forall va_i, va_j \in VA_t$, and $w_{va_i}, w_{va_j} > 0$,

$w_{va_i,va_j} = min(w_{va_i}, w_{va_j})$ and $n_{ij} = max(w_{va_i,va_j}, n_{ij})$

Decay

After reaching the end of the sequence, the weights of items in the M_{pa-va} matrix are decreased by multiplying them by a decay value, e.g., 0.9 (algorithm 5). This is done in order to bound the weights in the matrix, so that new feedback can

Algorithm 3 Implicit Feedback Incorporation

while $PA_t \neq \emptyset$ **do**
  on $f_{va_i} > 0$, /* feedback value given to va_i
  $w_{va_i} = w_{va_i} + f_{va_i}$
end while
if $PA_{t+1} \neq PA_t$ **then**
  update M_{pa-va} and M_{va-va}
end if

Algorithm 4 Updating Matrices

set $VA_t = \{va_s, 0 \leq s \leq |VA|\}$ /* at time t
for all $pa_i \in PA_t$ and $va_j \in VA_t$ **do**
  $m_{ij} = m_{ij} + \alpha$
end for
for all $va_i, va_j \in VA_t$, and $w_{va_i}, w_{va_j} > 0$ **do**
  $w_{va_i,va_j} = min(w_{va_i}, w_{va_j})$
  $n_{ij} = max(w_{va_i,va_j}, n_{ij})$
end for

have a practical effect even after a large number of update cycles.

ACTIVITY RECOMMENDATION IN THE OPERATING ROOM

We illustrate the approach in recommending relevant virtual actions with an example of surgical procedures in an operating room. We consider the case where a set of suggested virtual actions are presented on a large public display inside the OR. Based on our qualitative study, the following four types of virtual actions can be presented in a complete operation scenario: *First*, virtual actions that appear on the screen during the entire procedure from start to the end and remain relevant. For example, the va associated with the patient record, or the va presenting relevant guidelines. *Second*, virtual actions that are triggered once based on the physical actions and remain relevant during the rest of procedure. Changes in the context may modify the relevance weight of these virtual actions, which causes an update in the state of the va on the surface. An example is that the va associated with the surgical instrument list is mostly used while preparing or gathering the instruments, but it stays visible on the display during the operation. *Third*, virtual actions that are presented only in specific stages and disappear as the operation moves on. For example, the va related to notifying the surgeon about the status of preparation is only needed when the patient is anesthetized and ready for the surgery. *Finally*, virtual actions that appear on and off on the screen as the situation requires, e.g., emergency call to the anesthetist can become relevant in several phases of the surgery including pre- and post-operation phase.

To illustrate how the display adapts to changes in relevance, consider the following situation:

The preparation for the operation is finished; the patient is anesthetized; and the surgical instruments are ready for use. In this stage, the anesthesia checklist is no longer relevant. Even though the operation instrument checklist is less relevant, it is not totally irrelevant as its relevancy increases with triggering of context events, such as picking up an instrument

Algorithm 5 Decay

if $PA_t = \emptyset$ **then**
  for all $m_{ij} \in M_{pa-va}$ **do**
    $m_{ij} = m_{ij} * d$, $d < 1$ (e.g., 0.9)
  end for
end if

from the tray. The weight of the va named 'calling the anesthetist' changes if, for example, a nurse clicks on this button. This interaction will be considered as a positive feedback for the va.

EXPERIMENTAL EVALUATION

To evaluate our proposed approach, we implemented the algorithm in a simple user interface (figure 9) and conducted an experiment with a group of 16 clinicians in 3 different hospitals.

Hypothesis

Our main hypothesis was that with the proposed recommendation approach, the system can find a collection of relevant vas based on a set of pas as well as adjust and improve the relevance model over time based on the users' interactions. Adjustment includes creating new vas, changing the relevance weight of the existing vas, and updating the list of vas on the surface.

Experimental Setup

We used the data streams of 10 real-world surgical operations that were annotated and used for activity recognition. The operations were of the same type (open hernia), but they slightly varied in the procedure and the set of physical actions being performed. The total collection of virtual actions used during the experiment was made in cooperation with the medical staff. The system was designed to apply the recommendation algorithms on streams of physical actions and recommend the set of top 6 most relevant virtual actions on the surface. We also used a schema (figure 8) where we listed the flow of that typical operation and asked 6 participants (out of 16) with different specializations to rate the relevance of each va to each pa as well as add new vas if they were missing in the schema. The ranking was between 0 and 4, where 0 means that the va is irrelevant to current pa and 4 means it is highly relevant. For example, for the physical action of "intubation", one could rank the virtual action "call the porter" as irrelevant but rank the "intubation guideline" as highly relevant. We then used those ratings as a baseline to compare with the real time results.

Ten out of 16 clinicians (other than those who filled the schemas) volunteered to rate the system suggestions for relevant vas to current pas. The experiment started with null training, so all items in the matrices were set to 0. The system suggested a random top-k item to the user in the first operation, but the weights were adjusted as the procedure proceeded. Each operation, used the updated matrices from previous operations. The users were asked to put themselves in the situation of that particular operation and imagine what kind of information would be used throughout the surgery as well as how

irrelevant	mindre relevant	relevant	meget relevant	totalt relevant
0	1	2	3	4

dit speciale: operationstype:

Operation + Anæstesi Hændelser / Visninger			Forberedelse af operationinstrumenter			lejring	afspritning	afdækning (covering)	timeout	Incision	Indgreb (execution)	Lukning	forbinding	Oprydning på stue		
												instrumentoprydning				
	Tjek af anæstesi aparat	Anæstesi forberedelse (medicine + instrumenter)	Patient forberedelse for anæstesi	Bedøvelse	Intubering	patientovervågning						patientovervågning vækning			Ekstubering (extubation)	Klar til afgang
Vis patientjournal																
Vis anæstesiskema (præmedicinering)																
Vis patientværdier																
Vis guidelines for anæstesi																
Vis anæstesi tjekliste																
Vis guidelines for intubering																
Vis guidelines for operationsprocedure																
Vis patient safety tjeklist																
Vis operationinstrument tjeklist																
Kald anæstesilæge																
Send besked til kirurg																
Order blod																
Tjek patient info																
Send besked til portør																
Notificere opvågning om status																
kald rengøring																
anæstesimontering																

Figure 8. The schema used for collecting initial rates for relevance between physical actions and information resources.

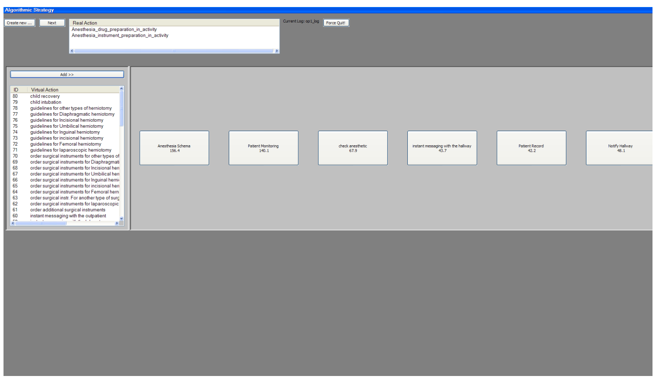

Figure 9. A screen shot of the implemented interface used for the evaluation. With change in the set of physical actions (the list on the top of the screen), a new set of 6 most relevant virtual actions appears on the right side of the screen. If a relevant virtual action is missing, it can be added to the screen from the list of actions on the left.

relevant or necessary those pieces of information would be. Each click on the va increased its relevance weight in that particular state of the surgery given the presented collection of physical actions being carried out in the operating rooms.

Results

For testing the hypothesis, the experiment was directed towards showing whether or not the number of relevant recommended items would stabilize as we got close to the last operation in the test (number 10). We logged users' interactions and choices of vas in a file whose rows contained 1) the set of physical actions 2) the initial set of suggested actions by the system and their weights and 3) the set of actions and their weights after the user interacted with the screen. The weight of each virtual action was a real number which we then normalized to a category of three ratings, namely totally relevant, relevant, and irrelevant. The latter was associated with those vas that were not used by the users. The expectation was that after going through each operation, the number

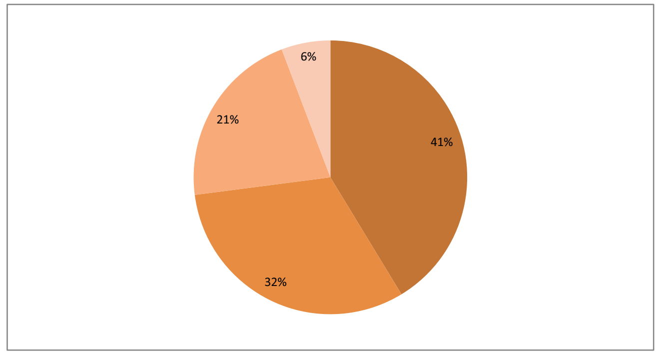

Figure 11. The diagram shows the difference in relevance weight between physical and virtual actions at the end of experiment compared to average data gathered from schemas as base. For example, 41% of the relevance weights after the experiment are identical with the values obtained from the schemas.

of totally relevant and relevant vas suggested by the system would increase resulting in decrease in the number of irrelevant items as well as reduction in the number of new items added by the user.

We calculated the average weights given by the users to the vas suggested by the system during each operation. We also computed the number of new items added by clinicians as relevant items to current pas. The results are summarized in figure 10. Despite the variations, the average number of totally relevant items seems to stabilize, the average number of relevant vas shows a slight increase resulting in reduction of irrelevant items. The number of new added items (lower diagram) is also decreased from ~ 4.5 items in the first operation to ~ 1.8 in the end, implying that more of the relevant actions were already part of the recommendation set in the later operations compared to the earlier operations.

We also computed the average weights given to pairs of virtual-physical actions by clinicians using the schemas (figure 8) and used it as the baseline to compare with the weights obtained at the end of the online experiment. The relevance rates in the schemas were between 0-4 (0 for irrelevant and 4 for totally relevant) while corresponding rates in the online experiment were real values. Hence, in order to compare the results, we first normalized the final weights for each virtual-physical action in the log file to a value between 0-4 using the min-max normalization [**?**]. Hence, considering min_{vp} and max_{vp} as the minimum and maximum values of each virtual-physical action pair (vp) in the matrix (M_{pa-va}), we mapped a value, v of vp to v' in the range $[newMin_{vp}, newMax_{vp}]$ by computing

$$v' = \frac{v - min_{vp}}{max_{vp} - min_{vp}}(newMax_{vp} - newMin_{vp}) + newMin_{vp}$$

We then computed the difference in the values of pairs in the schemas and in the matrix to see how close the weights obtained by the system were to the base rates. The diagram in figure 11 shows that 41% of the relevance weights after the experiment are identical with the values obtained from the schemas. In 32% of the pairs there is a difference of 1 in the

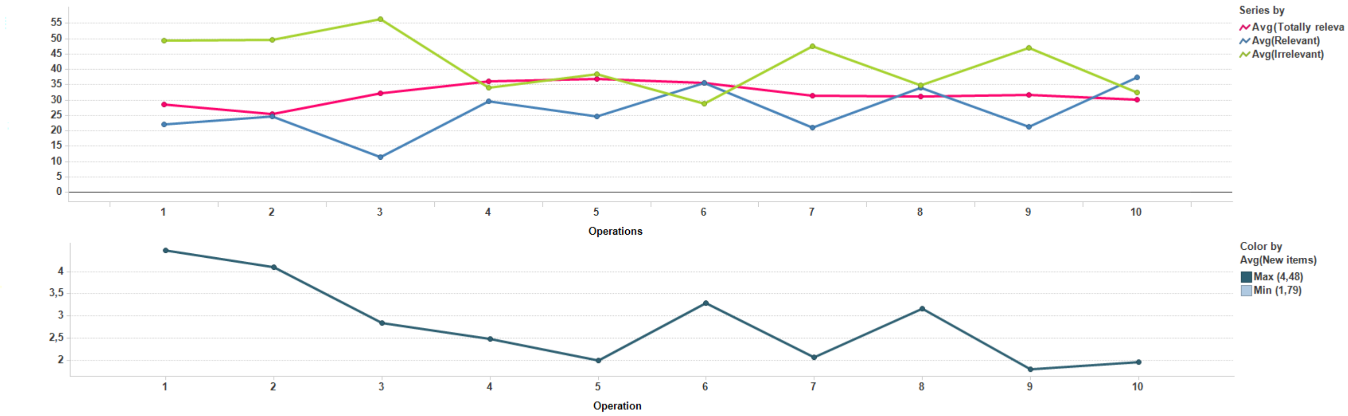

Figure 10. The upper diagram shows the development in presenting relevant virtual actions from operation 1 to 10. The lower diagram presents the average number of new items added by users in different operations.

scale. This means, for example, the relevance rate of a virtual action to a physical action based on the schemas is 4 whereas it is 3 in the normalized matrix M_{pa-va}. The percentage becomes less in the difference rates of 2 and 3 (21% and 6%) which means that relatively few pairs of virtual-physical actions have obtained a quite different relevance weight in the system after the experiment compared to what clinicians had rated in the schemas.

DISCUSSION AND CONCLUSION

We have presented our approach to supporting context-aware information retrieval and filtering in shared environments by developing a recommender system based on collaborative filtering and implicit feedback techniques. The method was employed in an activity-aware recommender system inside an operating room. Data collected from real surgeries was used as context to which the display adapts. The results of an experiment with clinicians showed a stabilization tendency in the set of relevant recommended items based on current physical actions being performed during surgical operations. Overall, 73% of the relevance rates between virtual and physical actions obtained by the system at the end of the experiment were identical or close to the rates given by clinicians in the schemas. This shows that the algorithm is able to incorporate the users' interactions with virtual actions as feedback and gradually obtain the same knowledge as intended by the users.

In our opinion, the preliminary user test indicates that the system effectively adapts to clinical professionals in order to display the most relevant information for a particular stage of an operation. However, we acknowledge that more thorough testing should be carried out. One future step is to conduct a study by testing the adaptive recommendation against a non-adaptive approach where all or a fixed set of virtual actions are presented to the users. The aim is measuring users' satisfaction in terms of e.g., access to relevant information and time consumption. Such testing, however, is rather complicated and resource-intensive due to the limited access to clinicians and ethical and other frameworks limiting access to operating rooms.

We tested our approach using data from the same type of surgical operation, i.e., open hernia and assumed that filtering and clustering of similar operations is done (the initialization process). We concentrated on evaluating the online part which is recommending relevant actions as an operation proceeds. In order to apply our approach to other types of operations, we must be able to handle very different procedures or (in general) different behavior patterns. An approach can be to cluster entities based on their activity patterns and then learn a generic model for each cluster. For new instances of that entity, we determine which cluster the entity belongs to and then apply the model of that cluster to infer the activities. We have determined some metrics that can be used to find similarities between the entities. For example, in case of surgical operations, the information about type, disease, duration, and patient's meta info can be used for clustering similar operations. Implementation and evaluation of this model is the next future step.

The current approach lets the participants in a task to initiate an interaction with the suggested information in order to trigger virtual actions instead of allowing automatic actions in the system. For example, when sending notifications is relevant, a personalized message with information about e.g., the participating surgeon is provided, but sending the message is done once a clinician in the OR confirms the action in from of e.g., a click. Giving the participants control over the system's actions has two advantages; first, as mentioned, it works as implicit feedback which helps retaining and updating the relevance model; second, it prevents triggering wrong actions that can cause consequences in critical settings such as operating rooms. However, we realize that in order to sustain this control, different interaction mechanisms such as speech recognition might be more appropriate in an e.g., OR, due to the hygiene issues.

We emphasize that the focus of this study has been on the collaborative aspect of the surgical procedures rather than concentrating on patient safety aspect and critical working environment in the OR. However, we believe many of those issues can already be addressed in our approach, and more capabilities can be added with only minor modifications.

REFERENCES

1. Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R., and Pinkerton, M. Cyberguide: A mobile context-aware tour guide. *Wireless Networks* (1997).

2. Adomavicius, G., and Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng. 17* (June 2005), 734–749.

3. Afsaneh Doryab. *Context-aware Information Adaptation In Collaborative Settings*. PhD thesis, IT University of Copenhagen, Denmark, October 2011.

4. Allen, J. F. Maintaining knowledge about temporal intervals. *Commun. ACM 26* (November 1983), 832–843.

5. Ardissono, L., Goy, A., Petrone, G., Segnan, M., and Torasso, P. Intrigue: Personalized recommendation of tourist attractions for desktop and hand held devices. *Applied Artificial Intelligence 17*, 8-9 (2003), 687–714.

6. Bardram, J. E., Bunde-Pedersen, J., and Soegaard, M. Support for activity-based computing in a personal computing operating system. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM Press (New York, NY, USA, 2006), 211–220.

7. Bardram, J. E., and Christensen, H. B. Pervasive computing support for hospitals: An overview of the activity-based computing project. *Pervasive Computing, IEEE 6*, 1 (2007), 44–51.

8. Cantador, I., and Castells, P. Semantic Contextualisation in a News Recommender System. In *Context-aware Recommender Systems 2009* (2009).

9. Cheverst, K., Davies, N., Mitchell, K., Friday, A., and Efstratiou, C. Developing a context-aware electronic tourist guide: some issues and experiences. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '00, ACM (New York, NY, USA, 2000), 17–24.

10. Dragunov, A. N., Dietterich, T. G., Johnsrude, K., McLaughlin, M., Li, L., and Herlocker, J. L. Tasktracer: a desktop environment to support multi-tasking knowledge workers. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, ACM (New York, NY, USA, 2005), 75–82.

11. Han, J., Kamber, M., and Pei, J. *Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems)*, 2 ed. Morgan Kaufmann, Jan. 2006.

12. McCarthy, J. F., and Anagnost, T. D. Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, CSCW '98, ACM (New York, NY, USA, 1998), 363–372.

13. Meyer, S., and Rakotonirainy, A. A survey of research on context-aware homes. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21*, ACSW Frontiers '03, Australian Computer Society, Inc. (Darlinghurst, Australia, Australia, 2003), 159–168.

14. Rattenbury, T., and Canny, J. Caad: an automatic task support system. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (New York, NY, USA, 2007), 687–696.

15. Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. *Recommender Systems Handbook*, 1st ed. Springer-Verlag New York, Inc., New York, NY, USA, 2010.

16. Sørensen, C.-F., Wu, M., Sivaharan, T., Blair, G. S., Okanda, P., Friday, A., and Duran-Limon, H. A context-aware middleware for applications in mobile ad hoc environments. In *Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing*, MPAC '04, ACM (New York, NY, USA, 2004), 107–110.

17. van Setten, M., Pokraev, S., and Koolwaaij, J. Context-aware recommendations in the mobile tourist application compass. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, P. De Bra and W. Nejdl, Eds., vol. 3137 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2004, 515–548. 10.1007/978-3-540-27780-4$_2$7.