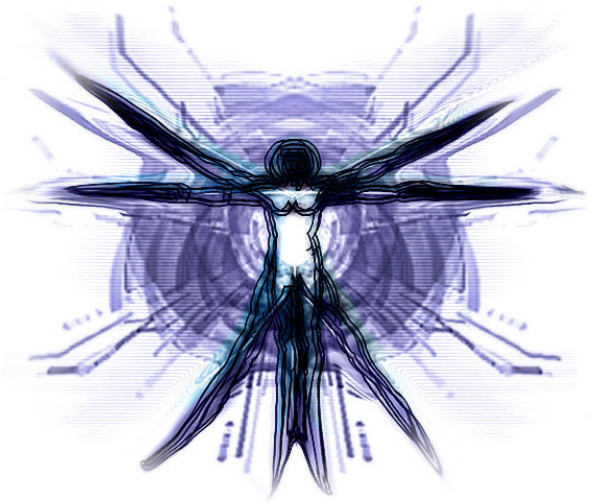

IS-ACTIVE

Inertial Sensing System for Advanced Chronic Condition Monitoring and Risk Prevention

D4.3 – Learning and adaptation techniques

Document ID:	IS-ACTIVE - D4.3
Document title:	Learning and adaptation techniques
Document type:	Deliverable
Contributors:	RRD, INE, UT

Abbreviations

Below is a list of abbreviations used in this document.

ADTree	Alternating Decision tree classifier
BFTree	Best First decision tree classifer
BMI	Body-Mass Index
CFS	Chronic Fatigue Syndrome
CLBP	Chronic Low Back Pain
COPD	Chronic Obstructive Pulmonary Disease
D	Deliverable
FB	Feedback
GA	Genetic Algorithm
IMA	time Integral of the Modulus of Accelerometer ouput
INE	Inertia Technology
J48	Pruned C4.5 decision tree classifier
J48graft	Grafted, pruned C4.5 decision tree classifier
JRip	Java implementation of Repeated Incremental Pruning to Produce Error Reduction (RIPPER) classifier
LOO	Leave-One-Out
NBTree	Decision tree with Naive Bayes classifiers at the leaves
PART	PART decision list, partial C4.5 decision tree classifer
PDA	Personal Digital Assistant
REPTree	Reduced Error Pruning tree classifier
Ridor	RIpple-DOwn Rule learner classifier
RRD	Roessingh Research and Development
WEKA	Java based machine learning toolkit

1. Introduction

The objective of IS-ACTIVE is to devise a person-centric healthcare solution for elderly with chronic conditions – especially people with COPD – based on the recent advances in wireless inertial sensing systems. The project emphasizes the role of the home as care environment, by providing real-time support to patients in order to monitor, self-manage and improve their physical condition according to their specific situation.

IS-ACTIVE focuses on simple and ubiquitous feedback interfaces, such as PDA, Tablet, TV and the inertial sensors themselves. Part of the research done in the IS-ACTIVE project is focussed on automated learning and adaptation techniques. The implementation of self-learning, adaptive algorithms can be found both in the sensor and in the PDA feedback device.

The research and development done in the area of learning and adaptation techniques is described in this deliverable (D4.3). The document focuses on two specific areas in which these techniques are applied: the assessment of physical activity levels using duty-cycled accelerometer data and the self-learning, adaptive generation of feedback messages for motivation of physical activity.

Section 2 describes an improvement of the IMA accelerometer-based method for estimating the level of physical activity of a person by adapting the duty-cycle to meet the accuracy requirements while reducing the energy consumption.

Section 3 explains the methods and algorithms used within the IS-ACTIVE PDA feedback application that automatically determine the right timing for presenting feedback messages to the user.

2. The Assessment of Physical Activity Level Using Duty-Cycled Accelerometer Data

This section describes an energy efficiency improvement of the IMA accelerometer-based method for estimating the level of physical activity of a person [Bosch et al., 2011]. The sensor sampling and data processing requirements are significantly reduced by duty-cycling sensor sampling and adapting the duty-cycle to meet the resource constraints of the sensor nodes and thus making the implementation and long-lasting operation possible on these resource-constrained devices. By duty-cycling, the system maintains adequate bandwidth, while still reducing the effective number of samples taken per unit of time. We analyze in detail the impact of duty-cycling on the accuracy of the method and show that we can reduce the duty-cycle to as little as 10%, incurring a mean error of only about 4%. This translates into energy saving of up to 60% on the sensor node.

2.1. Energy-efficient IMA

We aim to provide an efficiency improvement on the IMA accelerometer-based energy expenditure estimation method by Bouten et al. [Bouten et al., 1997], as described by the IMA formula of Equation 1 (see also D4.1). We focus on reducing the sampling frequency fs, thereby lowering the system's processing requirements and energy consumption.

$$IMA = \int_{t=t_0}^{t_0+T} |a_x|\, dt + \int_{t=t_0}^{t_0+T} |a_y|\, dt + \int_{t=t_0}^{t_0+T} |a_z|\, dt$$

Equation 1 – the IMA formula

However, a certain minimum sampling frequency is necessary to obtain a reliable result. This means that at some point, reducing the sample frequency further would impair the reliability and accuracy of the IMA output too much. This is mostly due to the fact that a certain minimum bandwidth is required to capture the most significant spectral components of the movement signal, which according to Bouten et al. are located roughly between 1 Hz and 20 Hz.

However, we could still improve efficiency in the time domain: it is probably not strictly necessary to keep sampling continuously. If we duty cycle the accelerometer sampling, we could achieve a higher level of efficiency, while maintaining sufficient bandwidth. Assuming that the level of physical activity does not change much in the inactive periods of the duty cycle, i.e. when the system is not sampling, the impact on performance would be minimal. Also, during the inactive periods, the system could enter a sleep mode to conserve energy. For this optimization, the IMA algorithm itself is not altered in any way and its band-pass filter is not reset at any point in time.

The defining parameters of a duty cycling scheme are the duty cycle *D*, i.e. what fraction of time the system is active, and the duty cycle period (T_D) or frequency ($f_D = 1/T_D$), which dictates how often the system switches between activity and inactivity per unit of time. Figure 1 schematically shows the proposed duty cycled sampling procedure for one activity value period T. Two alternatives are displayed. The top plot shows a duty cycle period that matches the period T, meaning that all samples are collected at the beginning of that period. The duty cycle is set to 20 %. The bottom plot shows a similar scheme, but the period T is devided in

multiple duty cycle periods (three in this case). The bottom alternative has the advantage that the samples are spread more evenly over the period T, which makes the chance of missing short but important significant activity smaller. The IMA output is probably going to be less reliable when the duty cycle period is very long, simply because the chance that the activity level changes significantly in the inactive period is higher. However, choosing a very short duty cycle period is also not beneficial, since the act of switching to and from sleep mode will claim system resources as well. The length of the duty cycle period is therefore an important design choice.

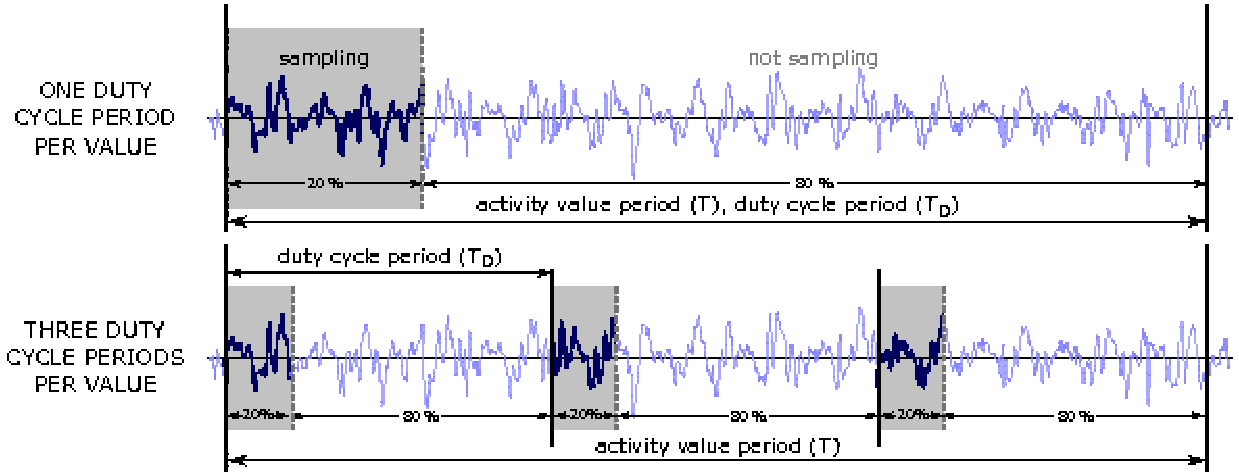

Figure 1 – Overview of duty cycle sampling scheme

Note that when f_D is very high relative to the sampling frequency at moderate duty cycle, the net effect will be very similar to just reducing the sampling frequency. As an extreme example, when f_D is equal to half the sample frequency and the duty cycle is 50 %, every other sample is dropped, which means that the effective sample frequency is equal to half the sample frequency. Even when f_D is set to a more useful value, we can still calculate the effective sample frequency $f_{s,eff}$, which is equal to the actual amount of samples collected per second ($f_{s,eff} = Df_s$). We use this as a measure for the amount of effort involved in the IMA calculation. Obviously, when the duty cycle D is 100 %, f_s equals $f_{s,eff}$.

Summarizing, the following parameters are important for our implementation of the IMA algorithm:

- Activity value period (T): the interval between successive IMA activity values.
- Sample frequency (f_s): the sampling rate of the accelerometer at the active periods of the duty cycle. This is what the hardware is configured to and the true rate at which samples enter the system when active.
- Duty cycle (D): The fraction of time the system is sampling. The effective sample frequency, which indicates how many samples are actually collected per unit of time, depends on this parameter and is equal to $f_{s,eff} = D_{fs}$.
- Duty cycle frequency (f_D): The frequency at which the system switches between active and inactive within the duty cycle scheme. The activity value period (T) should be an integer multiple of the duty cycle period ($T_D = 1/ f_D$).

The effect that these parameters have on the system's performance and effciency are evaluated in the next section. Note that when the bare output from integral formula of Equation 1 is used, its magnitude does not depend only on the level of activity. Parameters like the sample frequency f_s, the activity value period T and the duty cycle D also have a significant

influence on the resulting value. Therefore, we always normalize the activity values before any comparison is made, scaling it such that the theoretical (and practically unattainable) maximum IMA activity level will just fit into a 16 bit unsigned integer (= 216 − 1).

2.2. Experiments and results

To assess the impact of our efficiency improvement on the accuracy and reliability of the IMA activity value, we perform a series of experiments. We compare the output of the original unoptimized system at the highest sample frequency with the output of the system at various different optimized configurations. To be able to simulate different algorithm configurations using exactly the same conditions, we need to collect raw accelerometer data. This means that the actual experiments need to be performed just once and that the algorithm itself is executed offine in the simulation. Because the output is only compared between different algorithm configurations, and not between different activities or users, the absolute values of thesystem output are of little importance. This means that we can suffice with only one user, provided that a sufficiently broad set of activities is performed.

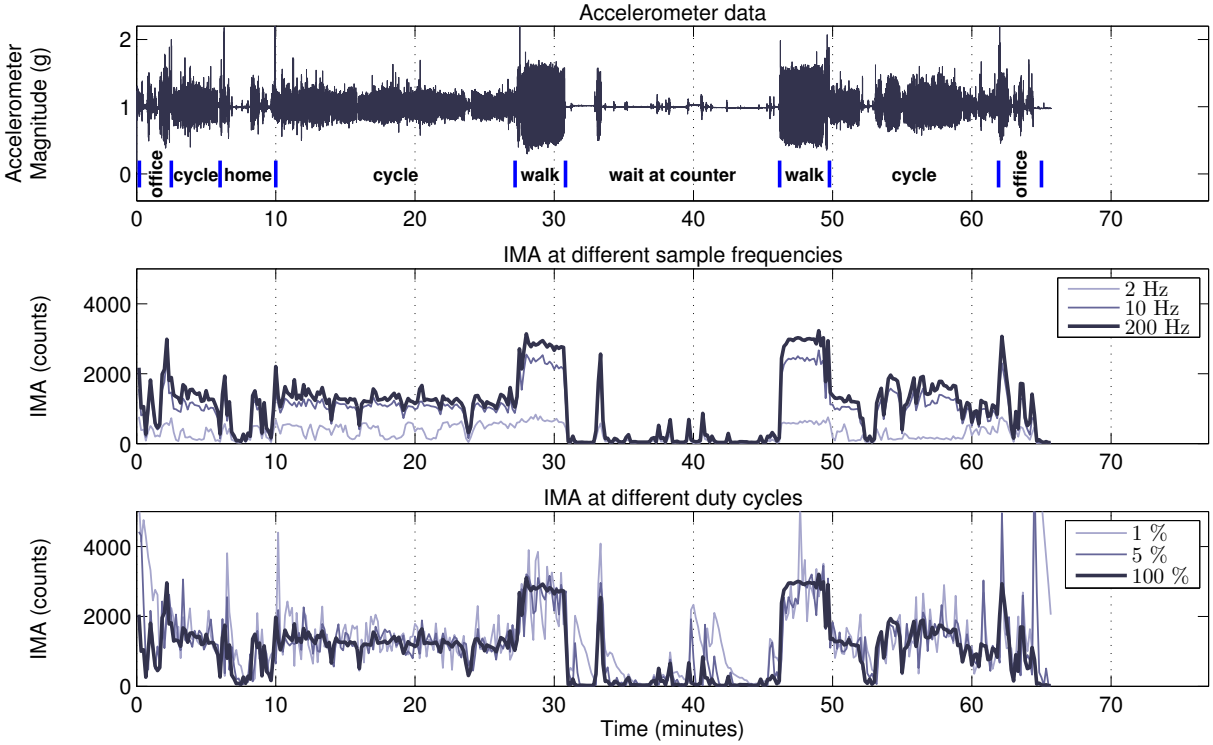

Figure 2 - IMA calculated from accelerometer data at various configurations

For this experiment, we use the ProMove inertial sensor node platform [ProMove]. We collect raw three dimensional accelerometer data at 200 Hz from one person performing daily activities. The sensor range is configured at ±6g. We collect approximately one hour of data involving activities like cycling, walking, standing and sitting. The sensor node is mounted at the user's waist on his belt. The data is logged to the node's on-board flash memory and downloaded wirelessly at the end of the experiment. The accelerometer data from the experiment is shown in the top plot of Figure 2. The data is logged from a bicycle errand to the local city. The first 2.5 minutes involve sitting at a desk, walking down some stairs and out of the building. After that time, a short cycling trip is started, which ends at the person's home.

At round the 6 minute mark, the trip ends and the person's activities involve walking around his home and standing still to talk to people. At minute 10, the actual bicycle trip to the city is started, which ends at the minute 27. At minute 24, the person needs to stop at a traffic light. Starting at minute 27, the person walks a significant distance which ends at the 31 minute mark at the counter of a shop. Up until minute 46, the person waits, which mostly involved standing and short walking activity. After that time the person walks back to his bike. Close to the 50 minute mark, the person reaches his bike and cycles back to the office. This trip involved a long stop at a tra□c light starting at minute 52 and the person arrives at the office building at the 62 minute mark. The log ends with the person walking back into the building and taking a seat behind his desk.

The middle plot of Figure 2 shows the resulting IMA activity values for three different simulated sample frequencies. The IMA values are produced at an activity value interval (T) of 10 s and normalized as explained in Section 3. The sample frequency is reduced from 200 Hz to lower frequencies by down-sampling the raw accelerometer data; this also involves low-pass filtering to prevent aliasing effects. At the full 200 Hz, the IMA values are the most accurate. When the sample frequency is reduced to 100 Hz or even 50 Hz, the IMA values do not differ very much from the 200 Hz case (not shown in the plot). When the sample frequency is reduced further to 10 Hz, the difference first becomes significant, especially for the more intense activities, such as walking. Notice that the produced IMA value is consistently lower than the equivalent at 200 Hz, which is to be expected since some spectral components that contribute to the signal energy are lost at lower sample frequencies. However, even though the IMA values differ significantly from the 200 Hz case, the overall trend is mostly maintained, which means that the activity intensities retain similar relative magnitudes. Finally, for a sample frequency of 2 Hz, this trend is lost for the most part. This is evident from the fact that walking (e.g. at minute 30) and riding a bicycle (e.g. at minute 25) become indistinguishable in terms of activity level. Apparently, some important frequency components for distinguishing physical activity levels are located above 1 Hz.

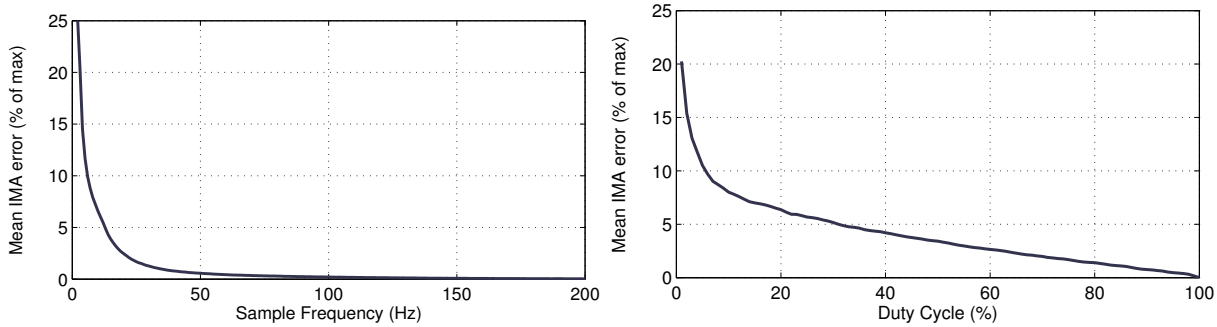

Figure 3 - Frequency and duty cycle performance statistics

Figure 3-left shows error statistics for a whole range of sample frequencies. The plot shows the mean error, expressed as the difference with the 'ideal' IMA output at 200 Hz. The error magnitude is shown as a fraction of the maximum IMA value encountered in the 200 Hz simulation. We notice that the error is relatively small when the sample frequency is above 40 Hz and becomes very significant when it drops below 10 Hz.

In the next simulations, we explore what happens when the duty cycle is varied. We fix the simulated sample frequency at an appropriate level of 50 Hz. The activity values are again produced with an interval T of 10 s, which is now also the duty cycle period (T_D). The normalization of the IMA value also accounts for the effect of the duty cycle. The bottom plot of Figure 2 shows the resulting IMA activity values for three different simulated duty cycles. As the plot indicates, the main effect of duty cycling is that the IMA output is less stable. This is to

be expected, since the system is sampling only at a fraction of the time, and at those instances the amount of motion may be significantly higher or lower than the motion that is skipped and thus not included in the result. The largest deviations happen at those instances where a brief dip or peak in the activity level happens just in the sample period of the sensor. Obviously, this effect is reduced when the duty cycle is higher and less samples are skipped. This effect is visible in Figure 2 for example at minute 40. An important difference with the reduction of the sampling frequency, as explored in the earlier simulations, is that the overall magnitude of the IMA output does not change, indicating that the system retains sufficient bandwidth. Figure 3(b) shows error statistics for a whole range of duty cycles 0.01 % and 100 %. As shown, between 10 % and 100 %, the impact of duty cycle on performance seems close to linear. However, when the duty cycle drops below 10 %, the difference grows much faster.

For the results in Figure 3 - right, the duty cycle period T_D was set equal to the activity value period T of 10 s. As explained in the previous section, this may not be optimal since 10 s is a long time. Therefore, we investigate what value for T_D would be optimal. To extend the range of possible values a little, we increase T to one minute. We investigate duty cycle periods T_D that are integer divisors of T. The results are shown in Figure 4. The sample frequency f_s is still set to 50 Hz. The horizontal axis of the plot shows the effective sample frequency, which directly maps to the chosen duty cycle ($f_{s,eff} = Df_s$), as explained in the previous section. Clearly, reducing T_D is beneficial, as the error drops consistently for almost the whole spectrum of effective sample frequencies. However, reducing T_D below 2 s shows no more clear improvement (not shown in the plot), which suggests that this is the optimum value in this case.

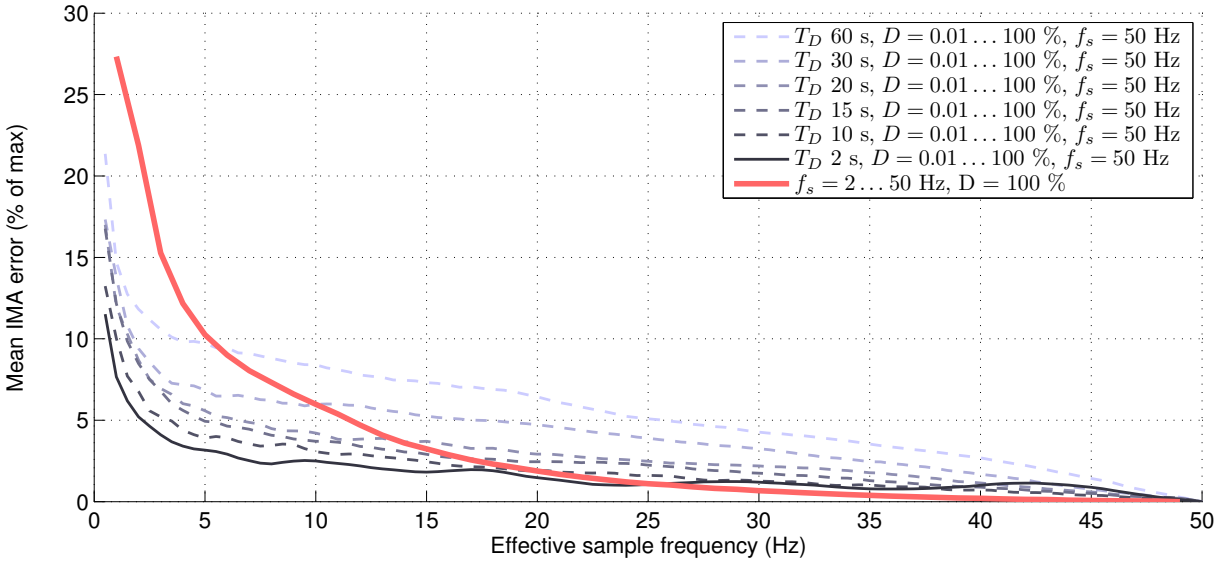

Figure 4 - Duty cycle performance statistics at varying duty cycle periods

It is now interesting to check how the reduction of the duty cycle - and thereby the reduction of the effective sample frequency - compares to the reduction of the real sample frequency shown in Figure 3-left. Figure 4 also includes a plot for the performance of varying the sample frequency directly, while keeping the duty cycle at 100 %. The comparative performance impact of reducing only the duty cycle versus reducing only the real sample frequency is clearly visible in the plot. Particularly to attain a low effective sample frequency (well below 25 Hz) to minimize processing effort, the use of duty cycling can be beneficial.

For example, we could aim to achieve an effective sample frequency of 5 Hz. We can do this by setting a duty cycle of 10 % at a real sample frequency of 50 Hz, which incurs an error of only about 3-5 % at a duty cycle period below 15 s. In contrast, reducing the sample frequency directly to 5 Hz incurs an error of more than 10 %, which is up to three times worse. Conversely, approximately the same 3 % error is incurred at a real sample frequency of 15 Hz, which means that, at that same level of error, the amount of samples that needs to be taken and processed can be decreased by about three times (i.e. 66 % less) using a proper duty cycling scheme. Also, at a duty cycle of 10 % the system can enter sleep mode up to 90 % of the time and, at a moderate duty cycle period of a few seconds, it can sleep for long consecutive periods, avoiding the burden of frequent sleep-wakeup cycles. This amounts to a very significant efficiency improvement, yielding estimated energy savings of around 60 % (including some additional overhead) when compared to reducing the sample frequency directly.

2.3. Conclusion

In this chapter we investigated improving the IMA algorithm in terms of its efficiency. We achieve this by effectively reducing the amount of accelerometer samples taken per unit of time. This is done by duty-cycling the sensor sampling instead of reducing the (hardware) sampling frequency directly. This preserves the system's bandwidth, while still reducing the sampling and processing needs. We perform simulations with the IMA algorithm for various possible configurations using data collected from a real-life experiment. The simulations show that a reduction in the energy consumption of up to 60 % is feasible using the duty-cycling method, with only a minor (up to 5 %) difference from the results produced at very high sample frequencies with no duty cycling. This work has been published in [Bosch et al., 2011].

3. Self-Learning, Adaptive Feedback on Physical Activity

In the IS-ACTIVE project, one of the goals is to help our patients achieve and maintain a healthy lifestyle. Patients with Chronic Obstructive Pulmonary Disease (COPD) need to remain physically active to prevent physical deconditioning. Increasing physical participation in everyday activities is therefore among the key goals of rehabilitation treatment in patients with COPD. Research on the daily activities of COPD patients has already shown that they are significantly less active overall than healthy controls (863±244 counts per minute, versus 1189±320 counts per minute) [Tabak et al., 2011].

2.2. Introduction

In IS-ACTIVE, we use a 3D-accelerometer (ProMove-3D) to assess the patient's daily activity pattern in IMA values, combined with a PDA for providing feedback. By comparing his activity to some predetermined reference activity pattern the patient is provided with feedback messages at regular intervals advising them to be more or less active.

A similar, earlier version of this system has been successfully used in studies to balance the daily activity patterns of chronic low back pain (CLBP) patients [van Weering et al., 2009], chronic fatigue syndrome (CFS) patients [van Weering et al., 2007], [Evering et al., 2011] and people suffering from obesity (BMI > 30). In the case of CLBP and CFS patient populations, the goal of the feedback is to spread activity over the day, while for obesity patients, the goal is to encourage them to increase activity over all.

The data collected during the aforementioned studies has shown that the approach is successful in general. The ability of the patient to see his/her activity in a graph (see Figure 5) gives insight into the patient's own level and spread of daily activities; and the addition of regular feedback messages to remind the patient of his/her status increases the patient's ability to balance or improve his own activity pattern further.

Figure 5: *the PDA activity monitoring and feedback application, running on an HTC Desire (Android 2.2), showing the activity graph of the patient (blue), plotted over the desired reference activity graph (green). The status panel shows the current deviation from the reference line in percentage points.*

Although providing feedback messages to the patient has proven to be efficient, compliance to the individual feedback messages remains relatively low. Table 1 shows some statistics about data collected from 5 individual data sets of collected activity and feedback data.

	CLBP	CFS	Obesity	CFS2	COPD	Total
Subjects	17	38	40	26	17	138
Days	322	675	308	331	262	1898
FB subjects	17	11	17	26	15	86
FB days	212	267	109	323	165	1076
FB given	1771	1300	1006	1440	1394	6911
FB usable	526	455	531	502	360	2374
Compliance	**61.98%**	**64.84%**	**47.27%**	**64.34%**	**58.06%**	**59.14%**

Table 1: *Data statistics from 5 corpora of activity and feedback data: CLBP (Chronic Low Back Pain), CFS (Chronic Fatigue Syndrome), Obesity, CFS2 (Chronic Fatigue Syndrome, Prognostic Cohort), COPD (Chronic Obstructive Pulmonary Disease).The* ***FB subjects*** *row indicates the number of subjects that received feedback,* ***FB days*** *is the number of days on which feedback was received,* ***FB given*** *the total number of feedback messages given, and* ***FB usable*** *the total number of feedback messages for which compliance could be calculated.*

The table shows that compliance to feedback messages is on average only 59%, meaning that almost half of the messages given to patients are ignored. The compliance measure we use compares the amount of activity performed in the 30 minute interval before the feedback event (Δ1), with the amount of activity performed in the 30 minute interval after the feedback event (Δ2). By comparing these values we can see if a subject was more active after an encouraging feedback message ($E_{ncouraging}$) or less after a discouraging message ($F_{discouraging}$). We refer to messages that suggest increasing activity as encouraging, and those that suggest decreasing activity as discouraging. An example of an encouraging messages is “you should go for a walk” and an example of a discouraging message is “read the newspaper”. If no more than three data points (3 minutes of measurement) are missing in Δ1 and Δ2, we can reliably determine the compliance and differentiate between the following cases:

1. Message Type ‘$F_{encouraging}$’ and Δ1 < Δ2.
2. Message Type ‘$F_{encouraging}$’ and Δ1 >= Δ2.
3. Message Type ‘$F_{discouraging}$’ and Δ1 <= Δ2.
4. Message Type ‘$F_{discouraging}$’ and Δ1 > Δ2.

In cases (1) and (4), we determine that the subject complied with the message, while in cases (2) and (3) we say that the subject did not comply with the message.

The old version of the activity monitoring and feedback system gives feedback on fixed time intervals (e.g. very hour), which, intuitively does not seem optimal. Therefore we aimed to create a feedback module that automatically determines an optimal timing of feedback messages. We want to find out why patients sometimes comply with feedback and sometimes not. To do this we try to predict the compliance of a feedback messages by looking at its context. After calculating the compliance for every feedback message in our datasets, the next step is to define this context in terms of features (see Section 2.2). After enriching our datasets with features, we use a statistical machine learning approach to find the relationships between context features and the compliance to the feedback message in Section 2.3. The work described here is adapted from [Akker et al., 2010].

2.3. Contextual Features

The context of each feedback message instance is captured in a set of features related to that specific feedback message instance. These features primarily should contain information that might be relevant for the patient's (unconscious) decision to either ignore or follow the given advice. Also, these features should be available to the system quickly and automatically. There are many conceivable reasons for a patient not to follow the advice that was given. For example "I don't feel like walking now", "I am tired" or "I am in too much pain right now" are all perfectly valid reasons, but they are difficult (if not impossible) to measure automatically. Not only would these factors be difficult to measure automatically, in this case they were not recorded in the corpora used here. That limits us to the data that was gathered during the feedback experiments and data that can be added retrospectively. The features that we defined fall roughly into five categories: (2.2.1) time related, (2.2.2) message related, (2.2.3) weather related, (2.2.4) history related and (2.2.5) activity related. We shall discuss these in detail now.

2.3.1. Time related features

The time related features that are calculated for each feedback message instance concern the time at which the message was generated.

- **dayOfWeek:** which day of the week it is.
- **weekDay:** whether this is a weekday or not.
- **dayPart:** whether the message was given in the morning (<12:00), afternoon (12:00-18:00) or evening (>18:00).
- **hourOfDay:** the hour of the day on which the message was given (rounded off).

The rationale behind these features is that people have both a weekly and a daily rhythm. This means that on some days (e.g. Sundays) people might want to relax and hence may be less motivated to be active. In case of daily rhythm, people might be bound to a sedentary job during certain fixed hours of
the day. Adding these features can, given sufficient data, help to detect individual's patterns and enable adaptive behaviour of the system concerting timing of feedback.

2.3.2. Message related features

The following message related features contain information about the message itself.

- **feedbackType:** whether this is an encouraging or discouraging message.
- **feedbackMessage:** the exact textual content of the feedback message.
- **messageGoOutside:** whether the feedback message advises the patient to go outside.
- **messageIsAQuestion:** whether the feedback message was phrased as a question.
- **messageSuggestIdle:** whether the feedback message suggests that the patient sits idle for a while.

The first two of these message related features are self-explanatory, but the last three might require some background. The reason for including the [messageGoOutside] feature is that if the system advises a patient to go out for a walk when the weather is bad, the patient might be inclined to ignore the message. This feature is thus related to the weather features (Section 2.2.3). Whether or not a message is phrased as a question (**messageIsAQuestion**) might influence the patient's willingness to react. Some people might prefer a system that is more

strict and issues its feedback as “commands”, while others might be more stubborn and dislike a commanding tone, thus preferring a more suggestive style of message. In the case of the **messageSuggestIdle** feature, some patients might prefer a more detailed suggestion (e.g. read the newspaper), while others might react more favourably to a message such as “take it easy”.

2.3.3. Weather related features

The following features contain information about the weather on the day the feedback message was generated. They are taken from the Royal Netherlands Meteorological Institute[1] and were added to the corpora retrospectively.

- **meanTemperature**
- **minimumTemperature**
- **maximumTemperature**
- **cloudScale**
- **precipitationSum**
- **precipitationDuration**

The reason for using weather data as features is that the weather conditions might influence the patient’s willingness to respond, especially when the message tells them to go outside.

2.3.4. History related features

The following features are related to the history of usage of the feedback system.

- **dayOfUsage:** a count of how many days the subject has been receiving feedback from the system.
- **totalMSGSToday:** the total number of messages received so far this day.
- **encouragingMSGSToday:** the total number of encouraging messages received so far today.
- **discouragingMSGSToday:** the total number of discouraging messages received so far today.
- **neutralMSGSToday:** the total number of neutral messages received so far today.
- **averageCompliance:** the average numerical compliance of all previous feedback messages this day.
- **sameMessageTodayCount:** the number of times the exact same message (as this message) was received today.
- **sameMessageOverallCount:** the number of times the exact same message was received in the total history.
- **sameTypeMessageTodayCount:** the number of times the same type of message ($F_{encouraging}$ or $F_{discouraging}$) was received today.
- **sameTypeMessageOverallCount:** the number of times the same type of message was received in the total history of usage.

These 10 features are designed to capture the state of previous interactions with the system. For example, receiving the same message several times on the same day might cause a habituation effect or even irritation leading to non-compliance. The same kind of reasoning lies behind the rest of the history related features.

[1] http://www.knmi.nl/klimatologie/daggegevens/index.cgi

2.3.5. Activity related features

The last two features are calculated from the measured activity level of the subject.

- **distanceFromReference:** the distance from the reference line at the time of the feedback message instance.
- **approachingReference:** whether or not the patient is approaching the reference line, calculated as the difference between the time of the feedback message instance T and T - 30 minutes.

A patient whose current activity level is much lower than his reference line (**distanceFromReference**) is possibly harder to motivate than someone who is closer to his optimal activity level.

2.4. Experiments

The goal of the machine learning experiments is twofold: to determine a suitable classification algorithm and to find the set of features that result in the best performance. Performance in this case is measured by accuracy of the classifier, defined as the number of correctly classified instances divided by the total number of instances in the dataset. Because the goal is to have a patient specific classification method, we choose to perform the experiments on the datasets of individual patients. Some patients in our datasets were given so few feedback messages that there was not enough data to perform the machine learning experiments on them, so they were excluded, leaving a total of 38 patients: 12 CLBP, 11 CFS and 15 Obese patients.

2.4.1. Baseline

In order to judge the accuracy of a certain machine learner outcome, a baseline is required. This was calculated for every patient by using a ZeroR, or most occurring class, classifier. The ZeroR classification scheme calculates the relative occurrence of classes in the training set, then, in the test phase it assigns to each unseen instance the most occurring class. In our two-class classification problem, if e.g. 60% of the instances are in the ‘yes’ class and 40% in the ‘no’ class, the ZeroR classifier will assign to each instance the ‘yes’ class and will achieve 60% accuracy. The average baseline performance over the 38 patients was 60.74% (σ = 7.01).

2.4.2. Genetic Algorithm

For all further experiments we use a genetic algorithm (GA) to search for good combinations of features.

“Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm [...]. In every generation, a new set of artificial creatures (strings) is generated using bits and pieces of the fittest of the old [...]”[Goldberg, 1989].

In our case, we use chromosomes (or binary strings) that represent subsets of the complete set of features: each position in the string maps to a specific feature. If the string contains a ‘1’ at a certain position, the feature to which that position is mapped is selected for the feature set, otherwise it is not. The fitness of chromosomes is calculated by filtering out all the features that are mapped to a ‘0’ in the bit string, and performing Leave-One-Out (LOO) classification with the remaining set using a certain machine learning scheme. The population size (numbers of chromosomes in each generation) is set to 100. In the selection step we use

tournament selection (see e.g. [Miller and Goldberg, 1995]) with a tournament size of 2. In the case that the two chromosomes selected in the tournament have the exact same fitness, the chromosomes with the lowest number of 1's is selected for reproduction. This favouring of smaller feature sets has been shown not to negatively influence the results of the search procedure, and it makes practical sense to do so since it results in classifiers that are faster to train and faster to test. The crossover- and mutation rates were fixed to 0.8 and 0.001 respectively. These values are close to the ones often cited in literature and have been experimentally determined to provide decent convergence rates. To reduce the probability of finding local optima from a GA run, all experiments were repeated 200 times, storing the global optimal results along the way. With these settings, in the feature selection experiments (see Section 2.3.4) on average 5036 feature sets were tested (σ = 270) per patient, which is 0.0038% of the total search space in an average time of 85 minutes (σ = 100) per patient.

2.4.3. Classifier Selection

The first part of the experiments deal with the selection of a suitable classification method. We applied a set of 35 different machine learning schemes from the WEKA Machine Learning toolkit [Witten and Frank, 2000]. In the first step we selected the patient with the highest instances count and ran the genetic machine learner for all classifiers. This resulted in an initial selection of the 10 best scoring classifiers: Ridor, part, ADTree, JRip, J48graft, J48, REPTree, NBTree, RandomForest and BFTree. In the next step we repeated the experiments with a larger number of patients. Because of the time constraint associated with running the genetic machine learning algorithm for all patients, we chose a set of 12 patients: 6 with relatively high instance counts (υ = 74, σ = 19) and 6 with relatively low instance counts (υ = 26, σ = 5). To determine the overall best performing classifier from the set of 10, we chose to implement a voting system, whereby for each run, the best performing classifier receives 3 points, the second-best receives 2 points and the third-best 1 point. Overall the Ridor classifier received the highest number of points: 27 out of a possible 36 (runner-ups had 22 points or less) and thus was selected as the most suitable classifier. The Ridor, or RIpple-DOwn Rule learner, is a simple rule learning scheme whereby first a most general rule is derived from the data and subsequently exception rules are generated in a cascading manner [Gaines and Compton, 1995].

2.4.4. Feature Selection

For the feature selection experiments we used the Ridor classification scheme with the aim of determining for all patients how the different features influence the classification of compliance performance. The genetic algorithm settings that were used are described above in Section 2.3.2 and will not be repeated here. Table 2 shows the average baseline, performance and relative improvement over the baseline per corpus as well as the overall average values. Numbers in brackets indicate the standard deviations. These results represent the best recorded scores during the genetic search over the feature space.

Corpus	Baseline	Score	Improvement
CLBP	61.94 (5.91)	86.16 (3.49)	+63.64%
CFS	63.70 (9.05)	87.24 (3.55)	+64.85%
Obesity	57.59 (5.00)	85.05 (4.83)	+64.75%
Total	60.74 (7.01)	86.03 (4.09)	+64.42%

Table 2: *Classification results per corpus. The improvement is calculated by placing the score on a scale from baseline to 100; the theoretical performance limit.*

All scores are significant improvements over the baseline with p-values less than 0.0001, computed using a paired t-test over individual patient results. For each patient, we look at the

feature sets that achieved the highest score using the least number of features. Figure 6 shows the list of features ranked by number of occurrences in the top-scoring, minimal feature set solutions. If e.g. for one patient there were 3 unique solutions and a feature was selected in 2 out of the 3 solutions, that feature's weight is increased by 2/3.

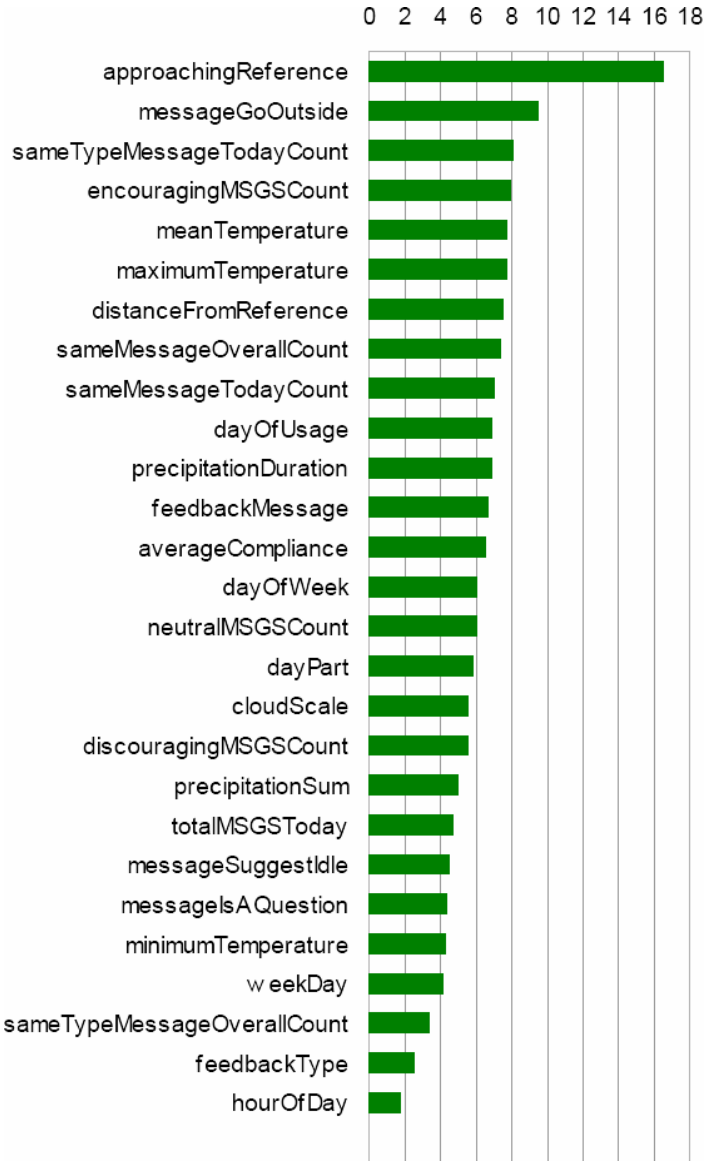

Figure 6: *Features weighted by their occurrence in the best-scoring, minimal feature sets.*

On average each feature was selected 6.28 times (σ = 2:73). This is observable from Figure 6 where only the feature 'approachingReference' really stands out. This means that for each patient, the Ridor classifiers that were trained and showed to have the highest performance each rely on very variable feature sets. This diversity in feature selection among the different subjects is most likely caused by the small datasets used for training the classifiers. Figure 7 shows the average improvement over baseline over all subjects plotted against the number of instances used for training. When ignoring the data point for 10 instances (υ = 32:5, σ = 37:1), a trend of rising improvement over baseline can not be seen. Usually when plotting instances versus performance a rise in performance is expected when using more training data, which is not the case here. This could mean that either all results are random (which they have proven not to be) or overall performance will only start increasing with much more data (more likely).

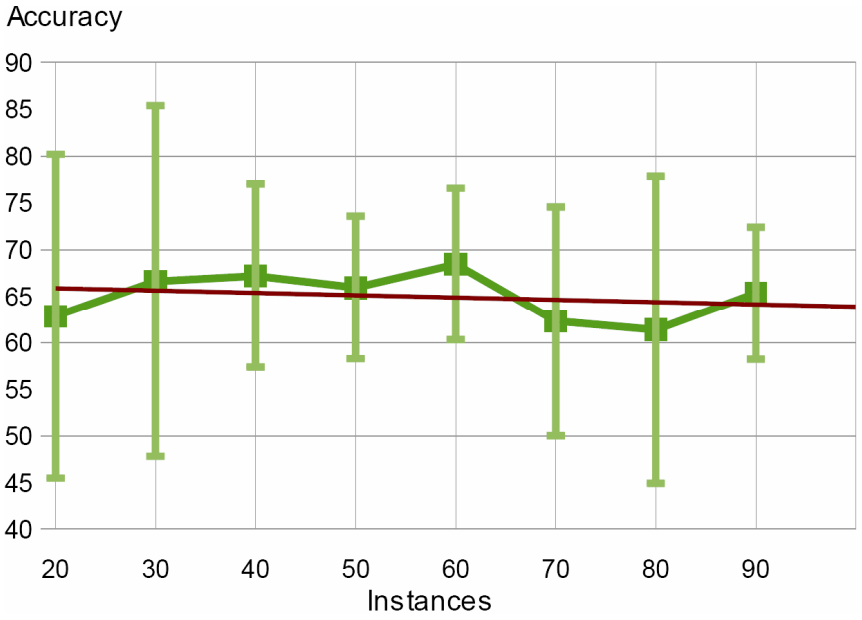

Figure 7: *Average improvement over baseline for all subjects when using increasing numbers of instances for training. In red is the linear trend line: -0.25 x + 66.08.*

2.5. Implementation

The work described in Section 2.2 and 2.3 has been implemented into the PDA application developed for IS-ACTIVE to do real-time prediction of optimal timing for feedback messages [Akker et al., 2011].

Figure 8 shows an overall architecture of the PDA software framework in the old configuration without the self-learning, adaptive feedback agent. The architecture is set up in a modular fashion, whereby each module handles a specific part of the application. The BluetoothModule handles communication with the ProMove-3D sensor, by reading and writing the data streams. These data streams are send to the ProMove3DReader which reads out the IMA values from the sensor and sends them through to the IMADataModule. The GUIModule, besides handling the Graphical User Interface, receives periodic requests for feedback from the UserInputModule. The GUIModule then passes the request on to the BasicFeedbackModule, which returns a random feedback message based on the patient's current deviation from the reference line (e.g. "Please go for a walk", if the patient is too far below its reference value).

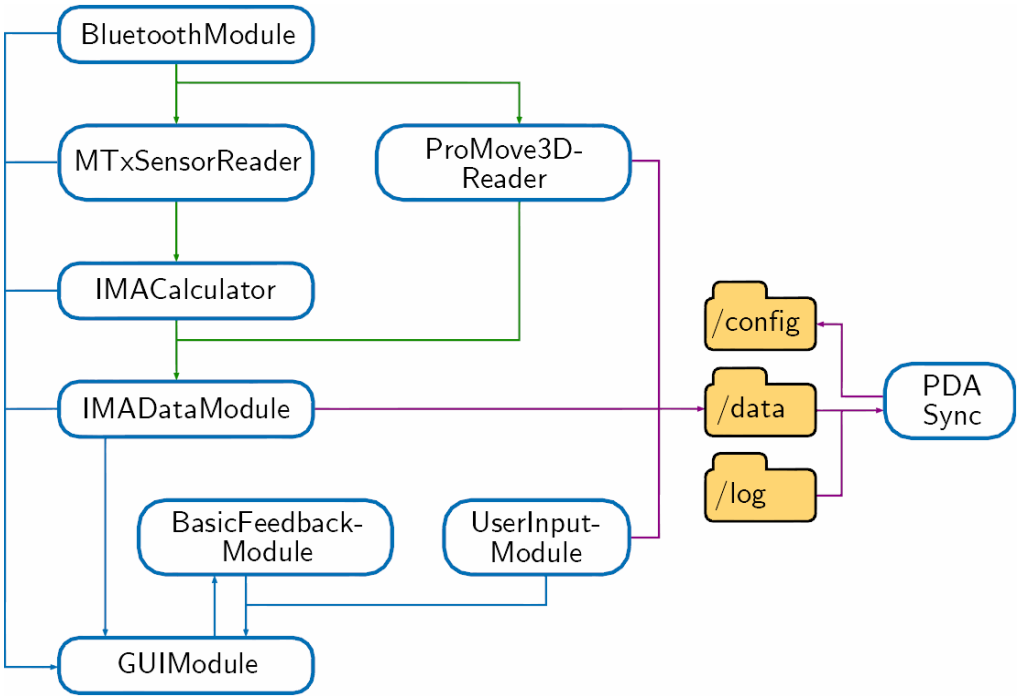

Figure 8: *Overall architecture of the PDA activity monitoring and feedback application, running on Android. This figure shows the modular set up and the specific modules used in the original application for providing feedback at regular time intervals.*

Figure 9, below, shows a detailed view of the modules involved in the implementation of the new FeedbackAgent, that automatically determines optimal timing for feedback messages. The FeedbackAgent has access to the IMA values from the ProMove-3D sensor via a direct stream from the IMADataModule. As the BasicFeedbackModule, it receives the feedback requests from the GUIModule (via the UserInputModule). These are now sent "constantly" (i.e. once every minute), and it is the FeedbackAgent's task to decide whether to honour the request or not based on its internal prediction methods. New modules introduced are the LocationModule, which provides up-to-date location information in the form of {*City, Country*} parameters and the WeatherModule which, based on the patients current location gives information on the current weather (temperature, condition, minimum- and maximum temperatures).

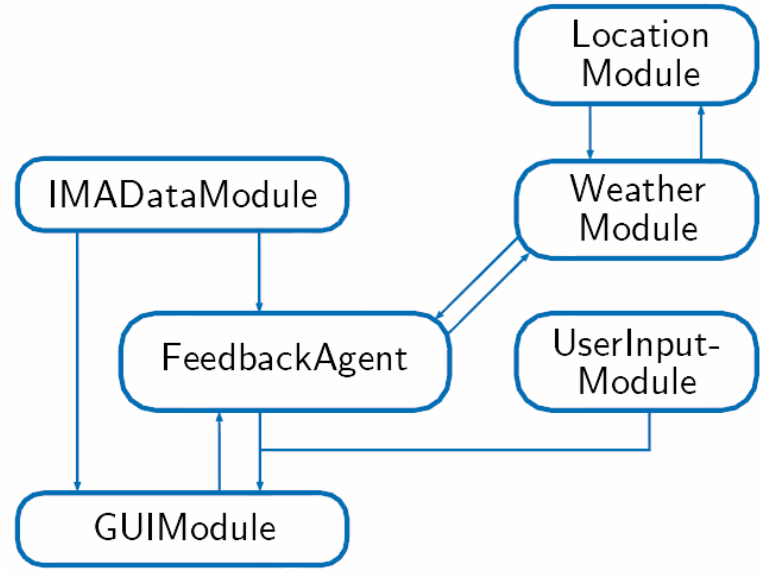

Figure 9: *Detailed overview of the modules involved in the implementation of the FeedbackAgent, showing also the new LocationModule and WeatherModule.*

Figure 10 shows the inner architecture of the FeedbackAgent module. Handling of a feedback request, coming in from the GUIModule every minute works as follows:

1. A feedbackRequest enters the FeedbackAgent.
2. The FeedbackAgent requests a list of *possible* feedbackEvents from the EventGenerator
    a. The EventGenerator creates a feedbackEvent: a feedback message with all its relevant features (parameters).
    b. The content of the feedbackEvent is provided by the MessageContentManager, which delivers the actual feedback message (e.g. "Please go for a walk outside") and the associated parameters (e.g. MessageGoOutside=true, MessageIsAQuestion=false).
    c. The subject's parameters are added to the feedbackEvent by the SubjectContext, which queries the MessageHistoryDatabase for history related features (e.g. NumberOfMessageReceivedToday, EncouragingMessages-ReceivedToday)
3. The *hypothetical* feedbackEvents are passed on to the Classifier.
4. The Classifier uses the WEKAInstanceGenerator to convert the feedbackEvent into a WEKA-compatible feedbackInstance.
5. The Classifier uses its internal classifier (a weka.rules.Ridor classifier) to predict compliance of the feedbackInstance.
6. The Classifier sends the classification result back to the FeedbackAgent (main component).
7. If the predicted compliance of the feedbackInstance is "false", the feedback request coming from the GUIModule is ignored.
8. If the predicted compliance of the feedbackInstance is "true", the feedback request is honoured, and the feedback message is sent to the GUIModule.

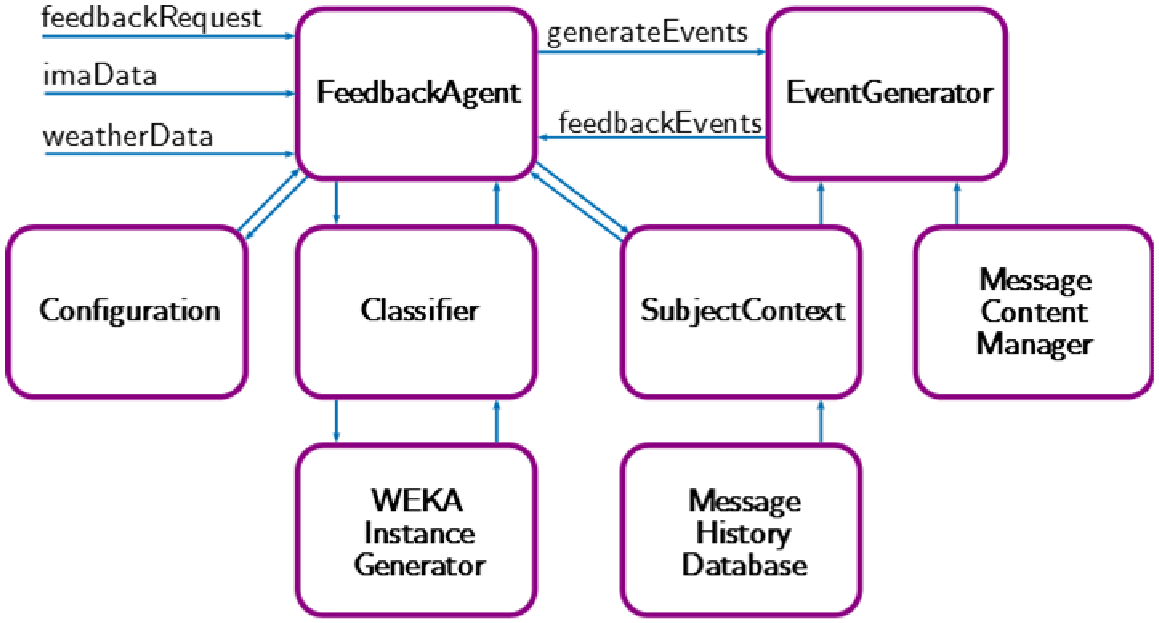

Figure 10: *Internal architecture of the FeedbackAgent module and all of its sub-components.*

If the FeedbackAgent decides the time is right for presenting the patient with a feedback message, the message is displayed as usual on the PDA's interface (see Figure 11). Then after 30 minutes of the patient viewing the feedback message, the FeedbackAgent calculates the actual compliance for the message that it sent. This information is then used to update the internal Ridor classifier to improve performance, and adapt the classifier slowly to the individual patient's preferences.

Figure 11: *The PDA activity monitoring and feedback application displaying a feedback message "Take a nice walk!" to the patient. From the user interface point-of-view, there is no difference with earlier versions of the activity monitoring application.*

2.6. Conclusion

The Self-Learning, Adaptive feedback module, or Feedback Agent, automatically predicts the right timing for presenting the patient with feedback messages on their daily activity levels. As a patient receives more feedback, the system updates itself by learning from the patient's reaction (compliance) to the given feedback. In this way, the system is self-learning and adaptive to the patient's personal preferences regarding feedback. The theoretical work behind the prediction of feedback compliance has been published in [Akker et al., 2010], while the implementation of the algorithms on a PDA application is discussed in [Akker et al., 2011a].

Current work in progress on self-adaptive feedback message content is published in [Wieringa et al., 2011], while future work in the area of smart, personalized feedback on physical activity in the terms of the combination of timing, content and representation is published in [Akker et al., 2011b].

References

[Tabak et al, 2011] M. Tabak, M. M. R. Vollenbroek-Hutten, P. van der Valk, J. A. M. van der Palen, T. M. Tönis, and H. J. Hermens, “How do COPD patients distribute their daily activities?,” in *Proceedings of ICAMPAM 2011: the 2nd International Conference on Ambulatory Monitoring of Physical Activity and Movement*, 2011.

[van Weering et al., 2009] M. G. H. Van Weering, M. M. R. Vollenbroek-Hutten, T. M. Tönis, and H. J. Hermens, “Daily physical activities in chronic lower back pain patients assessed with accelerometry.,” *European journal of pain London England*, vol. 13, no. 6, pp. 649-654, 2009.

[van Weering et al., 2007] M. G. H. Van Weering, M. M. R. Vollenbroek-Hutten, E. M. Kotte, and H. J. Hermens, “Daily physical activities of patients with chronic pain or fatigue versus asymptomatic controls. A systematic review.,” *Clinical Rehabilitation*, vol. 21, no. 11, pp. 1007-1023, 2007.

[Evering et al., 2011] R. M. H. Evering, M. G. H. van Weering, K. C. G. M. Groothuis-Oudshoorn, and M. M. R. Vollenbroek-Hutten, “Daily physical activity of patients with the chronic fatigue syndrome: a systematic review.,” *Clinical rehabilitation*, vol. 25, no. 2, pp. 112-133, Feb. 2011.

[Goldberg, 1989] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989, p. 432.

[Miller and Goldberg, 1995] B. L. Miller and D. E. Goldberg, “Genetic algorithms, tournament selection, and the effects of noise,” *Complex Systems*, vol. 9, pp. 193-212, 1995.

[Witten and Frank, 2000] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, no. San Francisco. Morgan Kaufmann, 2000, pp. 0-371.

[Gaines and Compton, 1995] B. R. Gaines and P. Compton, "Induction of ripple-down rules applied to modeling large databases," *Journal of Intelligent Information Systems*, vol. 5, no. 3, pp. 211-228, 1995.

[Akker et al., 2010] H. op den Akker, V.M. Jones, and H.J. Hermens, "Predicting Feedback Compliance in a Teletreatment Application," *Proceedings of ISABEL 2010: the 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Rome, Italy: 2010.

[Akker et al., 2011] H. op den Akker, H.J. Hermens, and V.M. Jones, "A Context-Aware Adaptive Feedback System for Activity Monitoring," *Proceedings of ICAMPAM 2011: the 2nd International Conference on Ambulatory Monitoring of Physical Activity and Movement*, 2011.

[Bouten et al., 1997] C. Bouten, K. Koekkoek, M. Verduin, R. Kodde, J. Janssen, A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity, Biomedical Engineering, IEEE Transactions on 44 (3) (1997) 136–147. doi:10.1109/10.554760.

[Bosch et al., 2011] Bosch, S. and Marin-Perianu, R.S. and Havinga, W.K. and Marin-Perianu, M. (2011) "*Energy-Efficient Assessment of Physical Activity Level Using Duty-Cycled Accelerometer Data."* In: The 2nd International Conference on Ambient Systems, Networks and Technologies, ANT 2011, 19-21 Sept 2011, Niagara Falls, Canada. pp. 328-335. Procedia Computer Science 5. Elsevier. ISSN 1877-0509

[ProMove] ProMove inertial sensor platform, www.inertia-technology.com