Event-Based Activity Tracking in Work Environments

Thomas Stiefmeier[1], Clemens Lombriser[1], Daniel Roggen[1], Holger Junker[2], Georg Ogris[2] and Gerhard Tröster[1]

[1]Wearable Computing Lab, ETH Zurich, Switzerland
{stiefmeier,lombriser,droggen,troester}@ife.ee.ethz.ch
[2]Institute for Computer Systems and Networks, UMIT Innsbruck, Austria
{holger.junker,georg.ogris}@umit.at

Abstract. Wearable computers aim to empower people by providing relevant information at appropriate time. This context-based information delivery helps to perform intricate, tedious or critical tasks and improves productivity, decreases error rates, and thus results in a reduction of labor cost.

To evaluate the usability of wearable computing in a work environment, we have chosen a car production scenario in which new employees are trained for mechanical assembly tasks. In this paper we describe the implementation of an activity tracking system that allows to detect the individual steps of assembling a front lamp into a car body. Our approach is to break down these steps of the assembly task into smaller units, so called discrete events. Body-worn and environmental sensors are employed to create these events which trigger transitions in a Finite State Machine (FSM). The FSM states represent user activities which correspond to the assembly steps.

1 Introduction

Wearable computers can support the user with context-sensitive information at appropriate time. The applications range from context-based information delivery [1] to pro-active health care monitoring [2].

The basis for wearable computing is the detection of the user's context. With the appropriate sensors and algorithms, context-based information delivery becomes feasible: the user receives adequate information automatically when he needs it, without having to explicitly ask for it.

Context-based information delivery can improve productivity in some work environments, for example in a maintenance scenario by highlighting parts of a complex schematic on a head mounted display (HMD). On the other hand, it decreases error rates, for instance in computer-assisted surgery. By validating the surgeon's actions online, he can be made aware of facts that may not be visible. Overall, context-based information delivery has the potential to reduce the time workers need to perform complex tasks. This is of great importance in such countries where the cost of labor is comparatively high. Table 1 describes some applications of context-based information delivery in work environments.

This paper describes the first results of an ongoing project that seeks to develop a system to help workers perform critical and complex assembly tasks in a car production environment. In order to foster user acceptance and wide applicability in various work environments, such a system must meet the following key requirements:

- **Unobtrusiveness:** Previous research has shown that a wide range of factors play a role in the user-acceptance of wearable systems. To promote user-acceptance, the system should be small, light, and must not hamper user motions [3]. The wearable system should present information the user needs in a convenient way, while user input is minimized, so that the user can concentrate on his work rather than on the wearable system.
- **Reliability:** The system must be reliable and robust in order to support the worker even for critical tasks.
- **Minimal instrumentation:** Depending on the application, adding sensors to the work environment may be impractical and/or costly. For instance in

Training:	The process of training unskilled workers for assembly tasks can be accelerated considerably. For example, presenting relevant feedback on a HMD supports the trainee with information when needed. Recording the training process enables the training supervisors on the other hand to get a precise overview of the trainees' present skills.
Maintenance:	In critical maintenance (e.g. aerospace industry), workers generally rely on paper checklists. Missing a verification step is however always possible and could be avoided by tracking the user's activity. Furthermore, an updated electronic checklist could be down-loaded specifically for each apparatus undergoing maintenance.
Surgery:	The location where the surgeon is to perform his operation can be enriched by overlaying x-ray images of the patient. In case of a problem, pictograms or literature relevant to the situation can be quickly presented to the surgeon.
Firefighter:	Presenting maps or blueprints of parts of the building can help a firefighter to orient in an unknown environment with poor visibility due to fire, smoke or steam. Group collaboration among firefighters could be enhanced by showing team members on a map of the firefighter's proximity.
Restaurant:	A waiter in a restaurant could benefit from using a wearable computing system that tracks his activities. The orders would be processed autonomously, leaving the waiter's hands free. Here, the activity tracking is based on the waiter's location and his hand movements.

Table 1. Applications of context-based information delivery in a work environment

a maintenance scenario, attaching sensors to each single screw would not be feasible.
- **Low Power:** Wearable computing components are battery powered. To ensure long run-times, the system needs to be power efficient.

Assembly and maintenance tasks are often composed of sub-tasks that typically need to be completed in a predefined sequence. In order to support the worker with helpful information, the wearable system must be able to track these tasks. We refer to this problem as *activity tracking*.

With this work, we demonstrate how user activity tracking can be implemented in a mechanical assembly scenario in which new employees are trained on the process of assembling the front lamp into a car body. The main contribution of this paper is the application of wearable computing based on simple sensors to this assembly task in order to track the worker's assembly progress. To speed up the learning process, the worker is supported with appropriate feedback, such as 'next steps' or 'dependency violations'. The deployed sensors comprise body-worn, car-mounted and tool-mounted sensors. Our activity tracking mechanism is based on *discrete events*, which trigger transitions in a finite state machine (FSM) whose states represent the sub-tasks. A graphical user interface continuously displays the list of sub-tasks that must be completed in order to assemble the lamp in the car. Based on the sensor readings and the state of the FSM, the sub-tasks which have been completed are marked as done and the upcoming task is highlighted.

This paper is organized as follows: In Section 2, we describe how we approach the problem of activity tracking based on discrete events. We show a preliminary demonstration of these ideas for a car-assembly scenario in Section 3. The results and limitations of the current approach are discussed in Section 4. In Section 5 we sketch improvements to the activity tracking. Finally, Section 6 concludes this paper.

2 Discrete Event-based activity tracking

Manufacturing, assembly and maintenance tasks often consist of a number of sub-tasks that must be completed in a predefined order. Some sub-tasks depend on others and must be completed sequentially while some others can be carried out in parallel. Our objective is to devise a system that allows to track the user activity within such a task lists based on readings of sensors placed on the body or in the work environment.

From the continuous sensor readings, we extract discrete events. Operating on discrete events is a more scalable approach than operating on continuous data streams, since events comprise the essential information of the raw sensor data but require less bandwidth for transmission. Furthermore, powerful string processing techniques are available when discrete events are considered as symbols [4, 5].

In order to achieve this objective we will thus: 1) segment continuous data streams into events, 2) associate sub-tasks with event sequences, 3) keep track of the current sub-tasks in the graph of sub-tasks, and 4) indicate completed as well as upcoming sub-tasks to the user with an appropriate user interface.

The entire approach and the implemented algorithms are detailed in the following section.

3 Tracking of the assembly workflow in a car manufacturing environment

As a first step of tracking the assembly workflow we focus on a single assembly task of the entire car assembly: mounting and adjusting the front lamp of the car. Figure 1 illustrates the front part of the car body where the lamp is to be mounted.

Fig. 1. Front part of the car body where the lamp is assembled

This front lamp task itself is composed of a number of dedicated sub-tasks indicated in Figure 4. First the lamp is inserted into the car body. Afterwards, a supporting plastic bar is installed and tightened with three screws. The lamp itself is then fixed to the car body by two screws. Finally, the alignment of the lamp with the car body is checked against a reference for quality control purposes.

3.1 Setup

In order to track the progress of the lamp assembly, we placed sensors on distinctive parts of the car body, on the worker and also on the tools such that all sub-tasks relevant for the different steps in the workflow can be detected.

Figure 2 shows the sensors worn by the worker during the assembly. The orientation and the movements of the hands are tracked using inertial sensor modules mounted on the back of the hands (A). The modules contain three-axis accelerometers, gyroscopes and magnetic field sensors allowing the calculation of the module's orientation with respect to an earth fixed reference frame. The

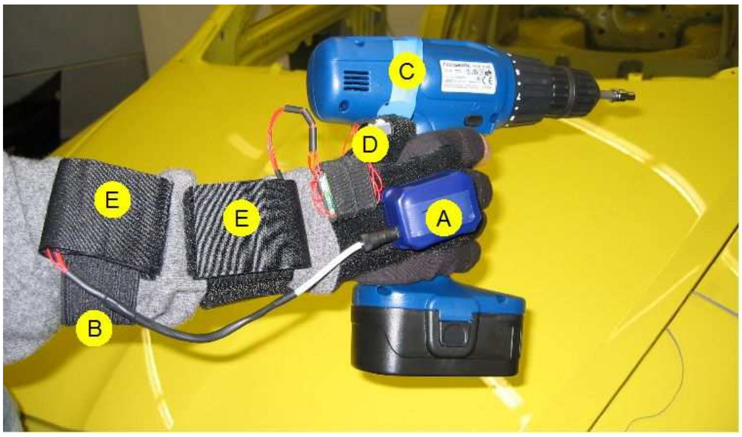

Fig. 2. Wearable sensors attached to the dominant hand of the worker

data streams acquired from these sensors allow to track both hands in three dimensional space. Whether the worker is grasping or holding anything is sensed at the forearm (B) through a band containing Force Sensing Resistors (FSR). The band measures the contraction of the muscle *flexor carpi radialis*. The level of contraction allows to deduce whether the fingers are bent and the hand is grasping an object or not. In order to know what has been grasped, all relevant work tools are equipped with a RFID tag (C), which is recognized by our system through a RFID reader module mounted between thumb and forefinger (D). A microcontroller connects periodically to the RFID reader to determine whether a RFID tag is (still) in the proximity.

As wiring up the worker would be too great an impediment for the user during his work, all data streams from the above mentioned wearable sensors are transmitted in parallel using Bluetooth modules (E).

The correct positioning of the assembled car components is monitored by a set of stationary sensors mounted directly on the car body. Critical locations with permanent contact to the component, e.g. the contact surfaces behind screws, are monitored by measuring the force exerted on FSRs on these surfaces. The FSRs' very low thickness (about 0.5 mm) allow it to be placed on such positions without modifications of the components or the car body. At locations where the assembled components do not touch the car body, we used magnetically triggered reed switches. They also measure the proximity of alignment checking tools used at specific places for quality control.

In our setup we have applied 5 FSR sensors and 4 reed switches around screw positions, the back of the lamp, and the checkpoints for the alignment tools. The sensors are read by a microcontroller mounted inside the engine compartment. It preprocesses the sensor readings and delivers either the current sensor values, or signals state changes.

3.2 Data acquisition

Figure 3 shows the system architecture and the data flow in our system. The data collected by the sensors is gathered on the assembly site by our *Context Recognition Network (CRN) toolbox* [6]. It provides a library of data processing algorithm modules frequently used for context recognition and allows to set up a process network using these modules.

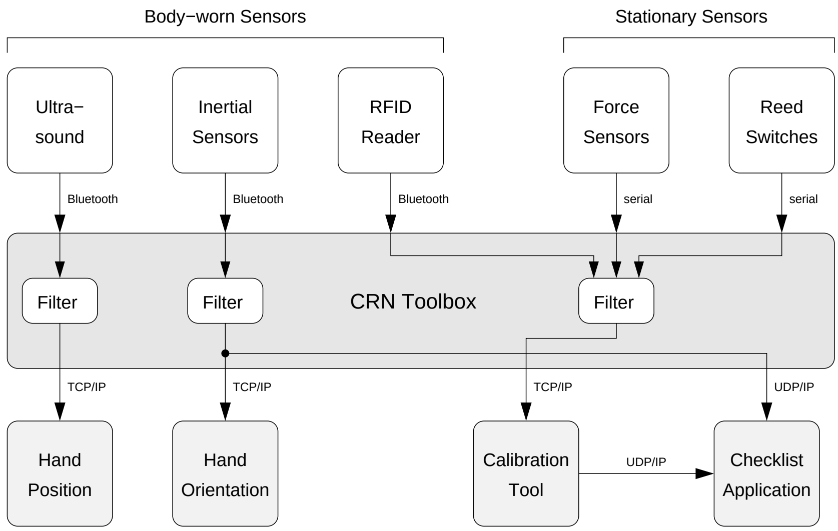

Fig. 3. System architecture displaying the different data processing layers

The sensor data arriving from the wearable and stationary sensors are read by the CRN toolbox and transformed by filters into a more directly usable form for a context recognition application. For instance, it converts the raw hand acceleration data into orientation information of the hand in three dimensional space.

Data computed by the CRN toolbox is made available on TCP/IP ports to the local computer network. The data streams are customized for the application accessing such a port and contain raw or processed data of selected sensors. This architecture allows the applications, which further process the data, to be located anywhere in the computer network, but also on the local computer. For example, the *Calibration Tool* in Figure 3 receives raw sensor data and provides a graphical user interface to adjust thresholds used for filtering the stationary FSR data.

The CRN toolbox also allows to feed data from the computer network into the internal process network, such that activities of additional workers can be taken into account when preprocessing the data from the one at the local assembly site.

3.3 Segmentation of continuous data streams into discrete events

Table 2 lists the continuous data streams that are processed. In order to obtain discrete events from these data streams, different approaches are carried out.

Data Stream	Retrieved Events
RFID data	tool taking/release
Reed switch data	lamp insertion/removal, alignment check
Force sensor data	screw tightened/loosened, object grasp
Inertial sensor data	torque level excess

Table 2. Available continuous data streams and retrieved discrete events

The RFID reader periodically scans its environment for valid RFID tags. The corresponding data stream can be interpreted as a binary sequence of states: either a tag is in proximity of the reader, or it is not. Detecting the changes of these tag states yields an event. For each known RFID tag we define the events *activated* and *deactivated*.

Force sensor data are discretized by applying adjustable thresholds. In case the sensor value exceeds a threshold, the corresponding sensor is on, otherwise it is off. Events are again generated on the occurrence of a transition between these two states.

The data stream acquired from the network of reed switches is inherently discrete because one switch is either on or off at a specific instant in time. Relevant events are the transitions from on to off and vice versa.

The gyroscope data stream originating from the inertial sensor module on the back of the hand is adopted to detect the event of a screw being tightened. The used cordless screwdriver is equipped with a torque limiter which causes vibrations when the actual torque needed to drive the screw exceeds the adjusted torque level. We apply a sliding-window on the gyroscope data stream and calculate the Fast Fourier Transform (FFT) over that window. Thresholding certain characteristic signal frequencies yields the information whether the torque level has been exceeded.

3.4 Finite State Machine implementation

Figure 4 illustrates the task graph of the front lamp assembly task. It is the basis for the implemented FSM.

When the lamp is inserted into the front part of the car body, the discrete event $R1$ based on a reed switch is generated. This event triggers the FSM to proceed from state *Lamp Insertion* to state *Fixing Plastic Bar*. The next transition depends on three aggregated events denoted by $S1$, $S2$ and $S3$ in Figure 4. Each of these events is composed of two atomic events. One event is caused by a force sensor at a screw and the other one is originating from the excess of the torque level. This aggregation of two events, which contain partly redundant information, is a first step in direction of relying less on stationary sensors but more on body-worn and tool-mounted sensors, namely using the torque excess for detecting that a screw has been tightened. However, recognizing the correct screw in case of a torque excess is not possible yet. Therefore, we still rely on the force sensors attached to the surface around the screws. The described transition from state *Fixing Plastic Bar* to state *Fixing Lamp Body* inherently comprises a parallel process in which the three screws are being tightened. The order does not matter. Important is the fact that all three screws are tightened and the torque level is exceeded at appropriate time instants. In Figure 4, we simplified this parallel process using the description $S1 \wedge S2 \wedge S3$. A similar transition is found between state *Fixing Lamp Body* and *Alignment Check and Adjustment* where the tightening of two other screws causing events $S4$ and $S5$ is involved. The last forward transition which completes the lamp assembly task is triggered by the occurrence of events $R2$, $R3$ and $R4$. These events originate from reed sensors which detect the usage of the alignment tool. Again, this transition is not depending on the events' order.

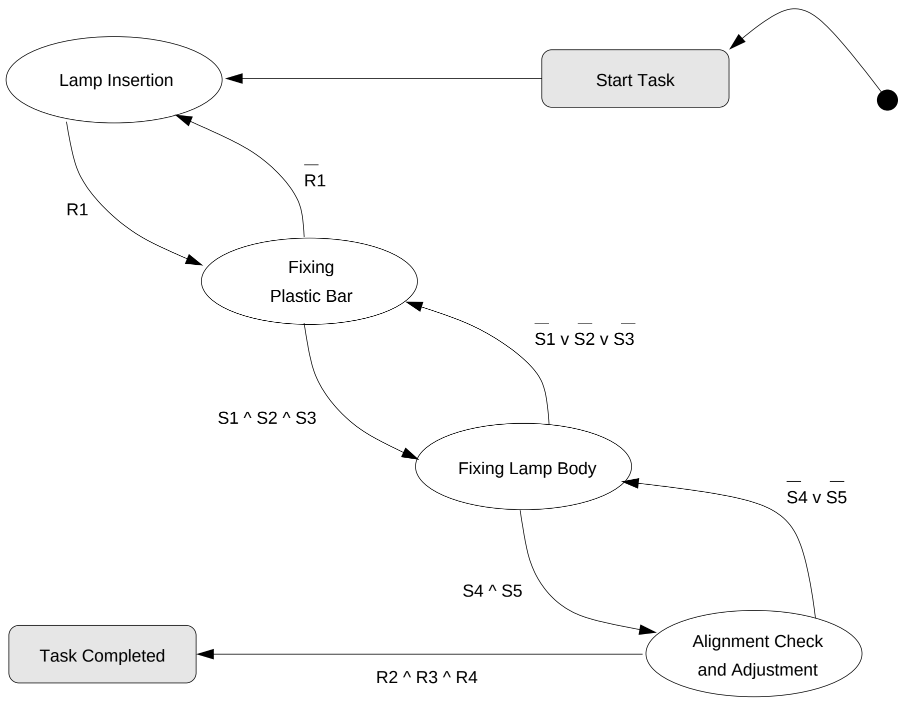

Fig. 4. Task graph of the front lamp assembly

In addition, the FSM implements some backward transitions to enable the worker to redo certain sub-tasks without being forced to completely start over.

4 Results

Our devised activity tracking system is able to track the worker's activity as described by the task graph. It detects the different assembly sub-tasks based on events created from the RFID reader, FSR, reed and inertial sensor data. Using a graphical user interface to configure the thresholds of the FSR sensors provides a quick way to adapt to changing setup configurations. The inertial sensors allow to detect the vibration of the hand, indicating that the screws have been tightened correctly. To allow the computation of different context information at dedicated locations, the data processing steps have been successfully distributed over several computing devices. This permits the assignment of computationally intensive processing steps to more suitable computing devices like servers.

As the system in the current state is a proof-of-concept, we only performed tests comprising four test subjects. We seek to develop a system using wearable sensors only. Currently, we mainly rely on the stationary sensors installed on the car body for the detection of the states. The inertial sensor data allow to compute the hands' movements and their orientation. However, the estimation of position using this information is not accurate enough. To determine that a screw has been tightened when the system recognizes that the screwdriver's torque level has been exceeded, we still need the FSRs on the car body. As an alternative, we have evaluated an ultra-sound system to locate the worker's hands. Such a location system utilizes so called beacons installed at known locations in the environment. Measuring the time-of-flight from these ultrasonic beacons to the receiver on the user's hand allows to determine the distances between the hand and the beacons. Triangulation then yields the hand's position in three dimensional space. Previous work in this field has shown good results [7].

The stationary sensors in the current system allow us to verify tracking results which are obtained only relying on the wearable sensors. This is a required prerequisite toward an activity tracking system based mainly on body-worn and tool-mounted sensors. The current state of the system serves as a basis for our ongoing work, which we outline in the next section.

5 Future work

In the current system we relied on sensors placed on the worker's body and in the environment. In order for the system to be more flexible, we would like to move as much as possible toward body-worn sensors. Still, sensors in the environment are necessary, for instance to detect the location of the worker relative to the work environment.

The FSM has been hand-designed. Although this was relatively straightforward, designing a FSM is likely to become increasingly difficult for more complex tasks. Automatically generating a FSM could be one way to solve this problem. Based on examples of user activity, genetic algorithms [8] could be deployed to define FSMs that correctly associate internal states with sub-tasks.

In the current implementation of activity tracking we relied on simple binary signals, Boolean logic and a FSM. Although this method was adequate here, it may not be general or robust enough for more complex tasks. We plan to examine other approaches such as string techniques [4], sequence matching [9] and learning user behavior [10].

6 Conclusion

In this paper, we have described the implementation of an activity tracking system. It is able to track the mechanical assembly task of mounting a front lamp into a car body. The devised system relies on discrete events which are generated using data from a RFID reader, force, reed and inertial sensors. These events trigger transitions in a FSM whose states correspond to the assembly sub-tasks. Our implementation monitors the progress of the assembly task by displaying an automatically updated checklist to the worker. Therefore, the system can be used as a basis for context-sensitive information delivery in a work environment.

References

1. Piekarski, W., Thomas, B.H.: Tinmith-Metro: New outdoor techniques for creating city models with an augmented reality wearable computer. In: Proc. of the 5th International Symposium on Wearable Computers, Los Alamitos, CA, IEEE Computer Society Press (2001) 31–38
2. Sung, M., Marci, C., Pentland, A.: Wearable feedback systems for rehabilitation. Journal of NeuroEngineering and Rehabilitation **2** (2005)
3. Gemperle, F., Kasabach, C., Stivoric, J., Bauer, M., Martin, R.: Design for wearability. In: Proc. of the 2th International Symposium on Wearable Computers, Los Alamitos, CA, IEEE Computer Society Press (1998) 116–123
4. Gusfield, G.: Algorithms on Strings, Trees, and Sequences. Cambridge University Press, Cambridge (1997)
5. Crochemore, M., Rytter, W.: Jewels of Stringology. World Scientific, Singapore (2002)
6. Bannach, D., Kunze, K., Lukowicz, P., Amft, O.: Distributed modular toolbox for multi-modal context recognition. In: Proceedings of the Architecture of Computing Systems Conference. (2006)
7. Ogris, G., Stiefmeier, T., Junker, H., Lukowicz, P., Tröster, G.: Using ultrasonic hand tracking to augment motion analysis based recognition of manipulative gestures. In: Proceedings of the IEEE International Symposium on Wearable Computing. (2005) 152–159
8. Goldberg, D.E.: Genetic Algorithms in Search Optimization & Machine Learning. Addison-Wesley, Reading, MA (1989)
9. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J Mol Biol **215** (1990) 403–410
10. Manavoglu, E., Pavlov, D., Giles, C.L.: Probabilistic user behavior models. In: Proceedings of the IEEE International Conference on Data Mining. (2003) 203–210