Ubiquitous Sensors based Human Behavior Modeling and Recognition using a Spatio-Temporal Representation of User States

Yoshinori ISODA, Shoji KURAKAKE, Hirotaka NAKANO
Multimedia Laboratories, NTT DoCoMo, Inc.
E-mail:{isoda, kurakake, nakano}@mml.yrp.nttdocomo.co.jp

Abstract

There has been much research on user activity assistance applications using the location of users and objects as context. However people's activities are described in terms of time sequence aspect in addition to location aspect. Therefore, it is important for enhanced user activity support systems to consider the user's context in terms of spatio-temporal constraints. In this paper, we propose a user activity assistance system that employs a state sequence description scheme to describe the user's contexts. In this scheme, each state is described as a spatio-temporal relationship between the user and objects in the real world. Typical sets of states are stored as models of tasks performed by a user. To try out this system, we have developed an experimental house containing various embedded sensors and RFID-tagged objects. Each state is detected by a decision tree constructed by a C4.5 algorithm using the output of the sensors and the RFID tags. The user's context is obtained by matching the detected state series to a task model. Having evaluated the performance of the proposed system in this experimental house, we conclude that our system is an effective way of acquiring the user's spatio-temporal context.

1. Introduction

Over the last few years, there has been growing interest in ubiquitous computing and networking environments in which users are surrounded by many items of equipment and sensors. It is envisioned that these environments enable any people to use computing and communication services with the minimum of fuss at any time wherever they may be.

To assist a user in finding and using suitable services, it is essential to ascertain the user's context. User activity assistance systems that take the user's context into account typically consider the spatial relationships between humans and objects in the environment (including ordinary household objects, domestic appliances, buildings and so on). This spatial information is an important aspect of the user's context with regard to the provision of assistance functions, and a number of functions and applications that use this information have already been implemented [1,2,3]. However, temporal information also has an important aspect of user's context in the real world. By ascertaining and analyzing both the spatial and temporal relationships between the user and objects, it is possible to support the user's activities in a more sophisticated manner.

The use of Hidden Markov Models has been proposed as a means of recognizing a user's state from time-series information [4,5]. However, to recognize the state of a user performing actions in real-world situations, it is necessary to deal with time scales ranging from brief transitions to long-term changes that may take place over several hours. To adapt flexibly to the time scale of the user's state that is to be recognized, it is necessary to use a mechanism in which the time-axis resolution can be changed adaptively according to the user's activity.

This paper describes the use of RFID tags and floor-mounted weight sensors to detect the spatio-temporal relationship between a human user and various objects, and discusses a method for representing the user's state based on information obtained from these devices with multi-resolution as spatio-temporal attributes. We propose a user activity assistance system that performs robust state decision by learning which attributes are valid for discriminating between the user's states based on information obtained from these sensors. The results obtained with a prototype system constructed inside an experimental house are then discussed.

2. Using ubiquitous sensor information for state decision and user support

In the real world, users perform a wide variety of tasks. In this study, a task refers to work performed by a user that is completed by performing specific state transitions, such as checking that doors are locked in multiple locations, or preparing a meal by following a series of procedures. When a user performs a specific task, it is not always necessary for the states associated with the task to occur consecutively. For example, during the execution of a task it is possible that a state may arise that is unrelated to the task. Furthermore, there are some tasks for which there is no inherent sequential relationship of states. To support users in tasks of this sort, it is necessary to recognize states of a discontinuous time series.

Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04)
0-7695-2051-0/04 $ 20.00 © 2004 IEEE

For this study, we prepare task models consisting of a set of user states for each task and support a user by adapting these models to a user's state series that are detected by using ubiquitous sensors information. Fig.1 shows an overview of a user assistance system. In a state recognition module, sensor information is transformed to a spatio-temporal representation of a user's state. A user's state is discriminated by a decision tree constructed in a teaching phase. Attributes that are valid for discriminating between user's states are learned as nodes of the decision tree. The state recognition module provides a series of user's states to a user assistance module. A user assistance manager in the user assistance module executes user assistance by matching the series of user states to a task model.

In the following subsection we explain the details of each module.

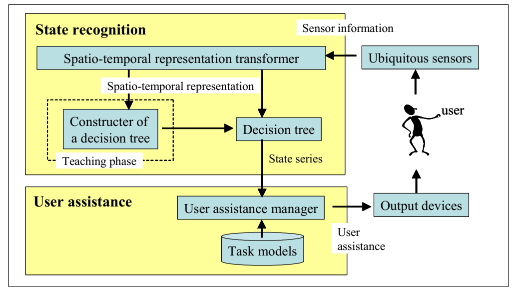


Fig.1: An overview of a user assistance system

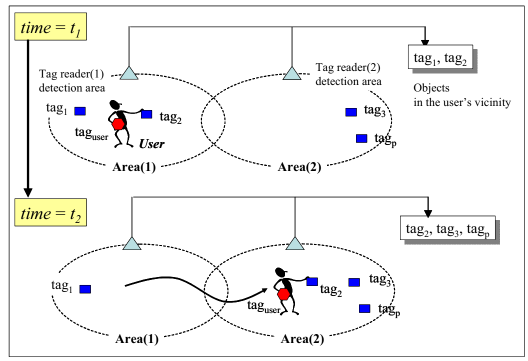


Fig.2: Changes of RFID tags associated with movement of the user

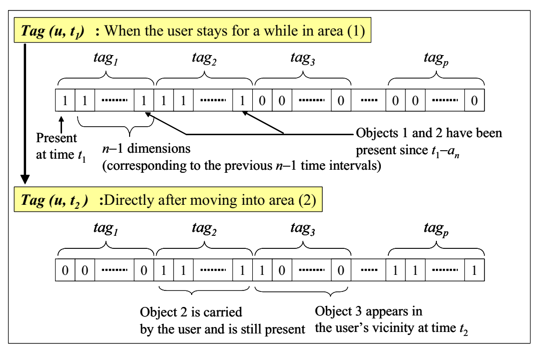


Fig.3. Representation of attributes relating to objects

2.1. The spatio-temporal representation of user states

RFID tags have recently attracted interest as a means of detecting objects in the vicinity of users. In this study, the attributes representing a user's state is derived from information obtained from RFID tags carried by the users and attached to objects.

As shown in Fig.2, the RFID tag information detected by each RFID tag reader is collected for each user. This forms attribute relating to the state of each user with regard to the objects in the user's vicinity. In addition to information on the presence or absence of objects in each user's vicinity, the temporal continuity of presence of these objects is also incorporated as attributes of the user's state. In other words, the user's state is represented by attributes that include information on how long each object has existed in the user's vicinity due to the user's movements and whether or not the objects in question are carried around by the user. The overall set of attributes describing the state of a user u in relation to these objects at time t is as follows:

$$
\begin{aligned}
Tag(u,t) = [&(tag_1(t,a_1), tag_1(t,a_2), ..., tag_1(t,a_n)),\\
&...,\\
&(tag_p(t,a_1), tag_p(t,a_2), ..., tag_p(t,a_n))]\\
(a_1 = 0,\ a_1 < a_2 < ... < a_n)&
\end{aligned}
$$

Here, $tag_i(t,a_k)$ is a binary value indicating whether or not object i has existed continuously in the vicinity of user u during the interval between time $t–a_k$ and time t (1=true; 0=false). Here, the user's vicinity corresponds to the detection ranges of the tag readers that detect the RFID tag carried by the user, so all the objects detected by the same tag readers that detect the tag carried by the user are regarded as being in the user's vicinity. In this way, the user's state is represented by attributes expressing which objects are in the user's vicinity, and if so, for how long. For example, Fig.3 shows the attributes relating to the user's objects at times t_1 and t_2 in Fig.2.

In addition to information on the relative positional relationships between the user and objects as detected by the RFID tags, the user's absolute position is also used as attributes representing the user state. As Fig. shows, the floor is partitioned into cells. Further attributes are used to express whether or not the user is present in each cell,

Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04)
0-7695-2051-0/04 $ 20.00 © 2004 IEEE

and if so, for how long. The attributes relating to the position of user u at time t are as follows.

$$\begin{aligned} Loc(u,t) = {} & [\,(cell_1(t,b_1), cell_1(t,b_2), ..., cell_1(t,b_n)), \\ & ..., \\ & (cell_q(t,b_1), cell_q(t,b_2), ..., cell_q(t,b_n))] \\ (b_1 = 0, & \ b_1 < b_2 < ... < b_n) \end{aligned}$$

Here, $cell_j(t,b_k)$ is a binary value indicating whether or not the user has been continuously detected in cell j during the interval between time $t-b_k$ and time t (1=true; 0=false).

Thus the entire set of attributes $A(u,t)$ representing the state of user u at time t is as follows:

$$A(u,t) = [\,Tag(u,t),\ Loc(u,t)\,]$$

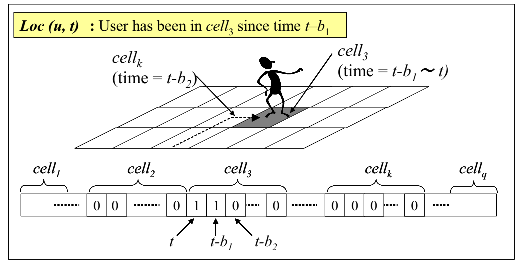


Fig.4: Attribute representation of user position

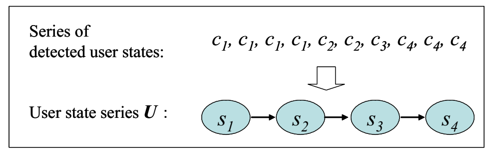


Fig.5: Example of a user state series

2.2. User state decision

In most state decision methods, the developer first has to prepare a model that takes the characteristics of actions into account. The specific states are then discriminated by matching this model to the incoming sensor information. However, to discriminate between the diverse states of users in everyday activities, it is difficult to determine what sort of characteristics should be extracted from the sensor information and how the model should be constructed to distinguish successfully between each of these states. Moreover, to discriminate between diverse everyday activities, it is necessary to use a large number of sensors. This means that large costs are incurred in calibrating every single sensor to match its output to the model.

Therefore in this study we employ a learning-based behavior modeling in which information obtained from RFID tags and sensors is directly mapped to classes of states to be discriminated. Specifically, a C4.5 algorithm [6] is used to learn the attributes that are valid for discriminating between the classes of states. A C4.5 algorithm constructs a decision tree by calculating the information gain ratio of each attribute from a training data set having multiple attributes, and successively employing the attributes with the highest values (e.g., $tag_i(t,a_j)$) as nodes of a decision tree. In a decision tree constructed in this way, attributes are only selected if they correspond to a spatio-temporal resolution that is suitable for discriminating between these user states.

When a decision tree has been constructed that discriminates between m different state classes c_j ($1 \leq j \leq m$), the state class c_j is detected by applying the entire set of attributes $A(u,t)$ obtained from RFID tags and sensors at fixed time intervals to the decision tree. In the resulting series of state classes, the parts $[c_j,...,c_j]$ in which the same state class exists continuously are treated as individual elements s_j, and these individual series form a user state series U. For example, in the output series of user state classes shown in Fig.5, the user state series U is $[s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4]$.

2.3. Supporting users by task modules

As mentioned above, we prepared task models consisting of a set of user states for each task, and supported the user by adapting these models to the discriminated user state series U.

A model of each task was pre-registered as a module describing (a) the task start-up conditions, (b) the task execution conditions, (c) the task execution state, (d) the user support method, and (e) the task completion conditions. Each of these modules (referred to as task modules below) is executed and managed separately. In this way, it is possible to provide users with appropriate support even when performing different tasks in parallel. It is also possible to adjust each module separately based on factors such as their user state decision performance.

The flowchart in Fig.6 shows how the task modules are started up, how their execution states are managed, and how user support is provided. At time t, the user state class c_j is discriminated by applying the information obtained from the RFID tags and sensors to the previously prepared decision tree. If c_j differs from the current state class output at time $t-1$, the user state series U is updated, and s_j is added to the series. The updated user state series U is then compared with the start-up conditions of the registered task modules to determine if any of the modules should be started up. The task module start-up conditions take the form of a regular expression[7] of a series of user states relating to the task, and are matched by tracing back through the series from the state s_j most recently added to the user state series U.

Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04)
0-7695-2051-0/04 $ 20.00 © 2004 IEEE

For example, suppose a user is involved in the preparation of food in the kitchen but has to leave the kitchen temporarily and returns after an interval. A user support module could be used to remind the user of the state that had been reached before the user left the room. The start-up conditions of this task module are declared in a regular expression requiring that (i) the current state is the state s_k that food preparation is in progress, (ii) this is preceded by states that are unrelated to the task, and (iii) these unrelated states are preceded by another state s_k. By performing matching in this way based on conditions expressed as a time-reversed series of states, it is possible to avoid starting up the task module when the user is in a state unrelated to the task in question.

When users perform real tasks, they may sometimes be temporarily interrupted, and may sometimes abandon the task altogether. In the latter case, the completion conditions of a task module that has been started up may never be met, causing the module to remain permanently active. However, when recognizing user actions over prolonged periods, it is difficult for the system to decide whether a task has been temporarily suspended or stopped altogether. To deal with this problem, the system occasionally asks the user to identify the task modules that are still active and manually terminate the modules of tasks that have been abandoned.

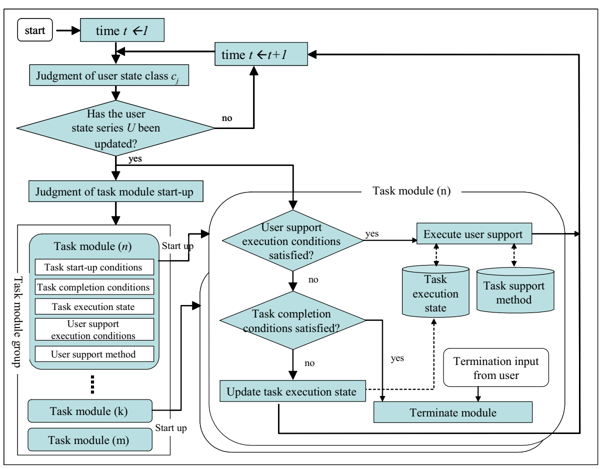


Fig.6: User support flowchart

3. Ubiquitous experimental house

Fig.7 shows the configuration of the prototype system constructed inside the experimental house.

3.1. RFID tag system

We installed an active type RFID tag system consisting of RFID tag readers and RFID tags which were attached

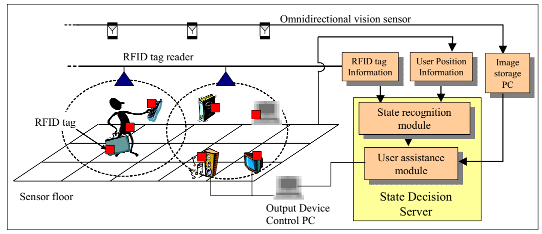


Fig.7: Test system configuration

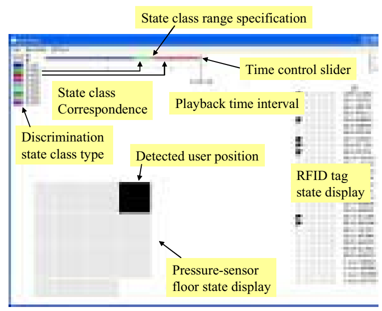


Fig.8: State class teaching screen

to the users and to various objects in the house. This tag system has low directionality in the tag detection ranges, and by attaching attenuators we were able to set the tag detection areas in the range from a few tens of centimeters to a few meters. Each reader communicates with a PC by TCP/IP via a protocol converter. The RFID tags transmit at intervals of 0.4 seconds.

3.2. Sensor floor

To detect the user's position, the floor was covered with a large number of pressure sensors. The detector units of each pressure sensor formed unit cells of 18×18 cm, and the binary value output by each cell was read into the PC via a serial port at 0.4 second intervals. In the prototype system, these detector cells were grouped into 5×5 blocks, so the user position was detected with a fundamental resolution of 90×90 cm.

3.3. State decision server

The state decision server operates in either of two phases. One is a teaching phase in which it acquires information from sensors and RFID tags and uses this information as teaching data to construct a state decision tree. In the other phase, it uses this decision tree to perform state decisions based on the current information provided by the sensors and RFID tags, and provides user support by updating the user state series and executing and managing task modules. In the learning phase, the RFID tag and sensor information is recorded when performing the user actions shown in the window in Fig.

Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04)
0-7695-2051-0/04 $ 20.00 © 2004 IEEE

8. This recorded information is played back using a time control slider, and the decision tree is constructed by using a mouse to specify specific time regions and allocate the state classes.

In the state decision/user support phase, the resulting decision tree and the current information from the sensors and RFID tags are used to discriminate the current user state, and the user is supported based on the resulting series of discriminated user state classes.

4. Experimental evaluation

4.1. Test conditions

To evaluate the performance of this system at making state decisions to provide user support, we performed tests in the environment shown in Fig. 9. These tests involved four types of discriminated user state classes — s_o: not in kitchen, s_m: in the kitchen, but only momentarily (less than 2 seconds), s_p: in the kitchen continuously for a brief period (less than 10 seconds), and s_l: in the kitchen continuously for a long period (at least 10 seconds).

An example of a situation considered in this test is the provision of support to a user involved in cooking who briefly moves away from the kitchen area and then returns. To remind the user of the stage that had been reached in the cooking process, support is provided by presenting an image representing the state that had been reached just before the user left the kitchen. This image is captured and stored by an omnidirectional visual sensor installed in the wall of the experimental house, and is shown on a display situated close to the user.

If this sort of user support is provided based only on the current instantaneous user position, then the image would be presented even when the user simply approaches the kitchen and then moves away again straight away. Therefore, this user support should only be performed when a match is obtained to a state series [$s_l \rightarrow s_o \rightarrow s_m \rightarrow s_p$] where the user has returned to the kitchen and a certain period of time has elapsed.

We used 21 RFID tags attached to objects (one of which was continuously carried by the user), and these were attached to objects associated with particular places (e.g., kitchen articles, a fixed telephone, stationery articles, etc.) and objects carried by the user (cellphone, books, bags, etc.). The outer limits of the RFID tag reader detection ranges are shown by the dotted lines in Fig. 9. Four binary attributes were used to represent the temporal continuity of the RFID tags and the user position detector cells. These were as follows: current value, ≥80% of the previous 10 frames, ≥80% of the previous 50 frames, and ≥80% of the previous 100 frames. The attributes representing the user state consist of 80 dimensions for the RFID tags (20 tags × 4), and the attributes relating to the user's position consist of 60 dimensions (15 detection units × 4), making a total of 140 dimensions. The state decision server read in the RFID tags and sensor information at an acquisition rate of 2 frames/second.

To acquire teaching data, the user performs actions such as standing in the kitchen and moving to another place for a few minutes, and the information provided by the sensors and RFID tags are recorded as a single teaching data set. Five sets of teaching data were obtained, and in each case the teaching screen shown in Fig. 8 was used to play back the information and allocate state classes to each time interval. The number of teaching data items allocated to the state classes in each set of teaching data is shown in the second column of Table 1.

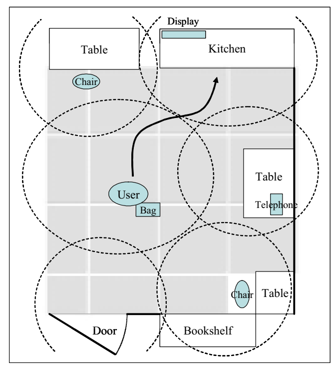


Fig.9: Test environment

Table 1: The number of teaching data items used to generate the decision trees

Teaching data acquisition	No. of teaching data items acquired	No. of teaching data items used for decision trees	Resulting decision tree
1st time	269	269	A
2nd time	638	907	B
3rd time	652	1559	C
4th time	1122	2681	D
5th time	895	3576	E

4.2. Test results

As shown in column 3 of Table 1, decision trees were constructed by successively adding together the 1st through 5th sets of teaching data, thereby resulting in the generation of five different decision tree A, B, C, D and E. The state decision performance of these trees was evaluated by applying each of them to a series of user action data acquired as evaluation data separately from the teaching data.

Fig.10 shows the recognition rates of each state class $\{s_o, s_m, s_p, s_l\}$ according as the amount of teaching data. These recognition rates were calculated by having a human allocate the correct state class to the state at each time interval in the evaluation data, and calculating the

Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04)
0-7695-2051-0/04 $ 20.00 © 2004 IEEE

proportion of cases where the state class output by the decision tree matched the correct state class. As the graph shows, the results produced by decision tree E (made with the largest quantity of teaching data) were more accurate than those of the initial decision tree A in the discrimination of all the state classes. Increasing the amount of teaching data caused a particularly large increase in accuracy in state class s_o where the state that can be adopted by the user are diverse.

However, as in state class s_p, there were also cases where the recognition rate decreased as the amount of teaching data was increased. This could be because of a bias in the teaching data or because the teaching data included states in which the RFID tags were not detected correctly when the user was performing some action.

Fig.11(a) shows the discrimination results for each time interval when the evaluation data was applied to decision trees A through E, and Fig. 11(b) shows an enlarged view of part of these results. The bottom parts of these figures show the series of the correct state class according to a human. In cases where the decision tree was constructed from a small amount of teaching data, the state classes were discriminated incorrectly for short periods as shown in Fig. 11(b). Increasing the amount of teaching data results in more suitable user state attributes being used in the decision tree, and with decision tree E the discrimination of incorrect state classes for short periods was eliminated. By adequately increasing the amount of teaching data, it is thus possible to provide the sort of kitchen user support envisaged above.

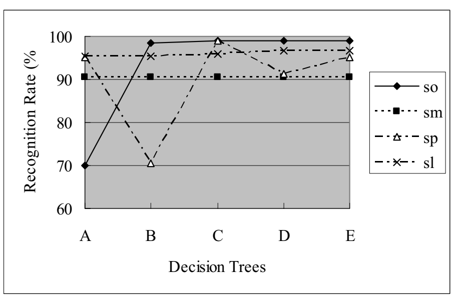
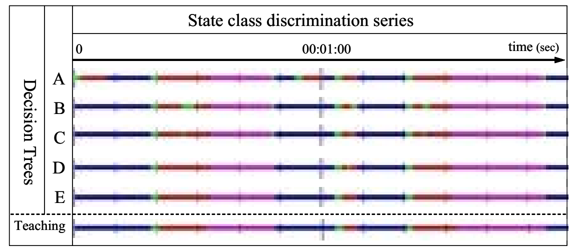


Fig.10: The accuracy of state class discrimination

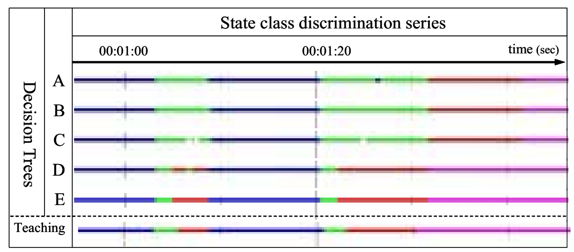
5. Conclusion

We have proposed a system that discriminates between user states by employing user state representations to express both spatial and temporal relationships between humans and objects in an environment, and provides user support by matching the result to a task model that consists of a set of states. We have also conducted tests in which user states were discriminated by using a prototype system constructed in an experimental house, obtaining favorable discrimination results.

Issues for further study include evaluating the performance of state decisions with respect to more complex user states, and modeling the state series of complex tasks performed by users. It would also be worth evaluating how the discrimination performance is affected by varying the time range of the attributes used to represent the user's state.



(a) Overall state class discrimination series



(b) An enlarged view of the above (a)

Fig.11: State decision series obtained with the evaluation data

6. References

[1] A. Ward, A. Jones, A. and A. Hopper, "A New Location Technique for the Active Office", IEEE Personal Communication, Vol.4, No.5, 1997.

[2] A. Harter, et al, "The Anatomy of a Context-Aware Application", Proc. of the MOBICOM'99, pp.59-68, 1999.

[3] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Hhafer, "EasyLiving: Technologies for Intelligent Environments", Handheld and Ubiquitous Computing, 2000.

[4] J.Yamato, J. Ohya, and K. Ishii, "Recognizing Human Action in Time-Sequential Images using Hidden Markov Models", Proc. of CVPR'92, 1992, pp.379-387.

[5] B. Clarkson, and A. Pentland, "Predicting Daily Behavior via Wearable Sensors", TR540, July, 2001.

[6] R. Quinlan, C4.5: PROGRAMS FOR MACHINE LEARNING, Morgan Kaufmann Publishers Inc., 1993.

[7] J.E.F. Friedl, Mastaring Regular Expressions, O'Reilly & Associcates, Inc.

Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA'04)
0-7695-2051-0/04 $ 20.00 © 2004 IEEE