MULTI-AGENT BASED SURVEILLANCE OF HUMAN WORKFLOWS

Emmanuel Sardis, Vasileios Anagnostopoulos and Theodora Varvarigou
National Technical University of Athens - NTUA, Athens, Greece

Keywords: Multi-agent systems, Workflows, Recognition, Human workflows, JADE.

Abstract: Workflow recognition through processing of humans and objects in a camera sensor network, presents a significant challenge recently. Human action recognition and sequence of actions manipulation, that construct a workflow situation/rule, has many practical applications in many different real human application environments. This article presents a multi agent based real time infrastructure, for recognizing humans workflows, by evaluating and processing computer vision signals, from multiple cameras sensors. The system architecture, the related agents' infrastructure of the distributed environment of sensors, are presented together with the algorithmic modules, that evaluate sensors signals into workflow events, and related alarms' outputs from the system. The article presents a full functional system that integrates the distributed functionality of the multi agents' infrastructure into real working environments, using the JADE agents' technologies. The evaluations of system simulation results are conclude this work, giving related feedback for possible future architecture and implementation extensions.

1 INTRODUCTION

The problem of task recognition is of paramount importance during industrial workflows. Accidents and abnormal behavior happen very often and for this reason it makes sense to develop automatic visual surveillance systems that take advantage of the recent progress of computer vision techniques to monitor activity areas. Thus, increased safety of the workers reduces operational costs and productivity in some cases can also be improved. There is generally a big gap between the perceived abnormal behavior by humans, and the one perceived by computer vision systems. The discrepancy is mainly accounted for by the coarse information provided by automatic systems. Moreover, a huge amount of rules learned by experience helps humans identify accidents and abnormal behavior.

Another big problem is the lack of flexibility. Sub-tasks comprising a workflow do not happen in a timely and a well-ordered manner. As a consequence, we need relaxed rules that can define a normal workflow execution. Instead of imposing artificial constraints that could hinder the productivity (like walking on a coloured line with a tolerance of some centi-meters), which usually are not well suited to the working environment and can break down under unexpected circumstances, we choose to construct generic systems that can encode human experience in a high-level way. Such systems are more amenable to be re-used, since application specific systems are more expensive. By allowing for high-level input we hope to decouple the problem, from the requirement of specially trained personnel with computer vision knowledge. Our approach from a non-technical standpoint amounts to partitioning the space in regions of interest and monitor interesting activities there. The usage of a multi-agent based architecture provides the required infra-structure to achieve this goal. Its suitability will be analyzed and proved in the following sections.

Next sections present the status of the workflow systems, section three suggests the decomposition in micro-tasks and its justification, section four analyzes the multi-agent underpinnings of the proposed solution, and section five presents the main system architecture and its technical analysis. In section six we present the workbench environment that was created in order to test our system with the related results and evaluation notes. Finally, in section seven, we conclude the article by discussing future architecture extensions and research activities for enhancing our proposed solution.

Sardis E., Anagnostopoulos V. and Varvarigou T..
MULTI-AGENT BASED SURVEILLANCE OF HUMAN WORKFLOWS.
DOI: 10.5220/0003305305240530
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 524-530
ISBN: 978-989-8425-40-9

2 STATUS OF WORKFLOW SYSTEMS

A workflow is defined as the computerized facilitation or automation of a business process (WfMS, 1999). It is composed of tasks, each of which corresponds to a unit business task or another business sub–process. The main steps of a workflow execution is a repletion of (a) scheduling tasks, (b) assigning tasks, and (c) obtaining results from the environment until the entre workflow process completes. Related workflow works propose the usage of agents and web services like (Purvis et al., 2004), (Vital et al., 2004), (Fleurke et al., 2003), (Savarimuthu et al., 2004) and Service Oriented Architectures like (Poggi, et al., 2007), (Mok et al, 2006), (Buhler et al., 2005), (Negri et al. 2006), where they present a solution by integrating agent technologies and SOA technologies, that is Web services, workflow, rule engine and semantic Web, or Petri Nets. Workflows and Agents technologies are complementary to each other, and although there has been a lot of work on integrating the two, no real time applications and systems have been implemented yet. In our case study and proposed system solution, we are using a multi agent based architecture, implemented and simulated with real time camera sensors and a workbench environment.

3 THE CONCEPT OF WORKFLOW MICRO-TASKS

The lack of regular execution of a workflow due to sub-task ordering or inclusion of unrelated sub-tasks makes the problem of workflow identification an ill-posed problem.

The first step is to split the workflow into more elementary tasks that can be combined to create the workflow. There can be a multitude of ways to split a workflow into subtasks. However, a good rule of thumb is that, since we usually recognize more than one workflow, we generally select the minimum subset of possible micro-tasks that can give the whole workflow. Moreover, we require that a task be atomic, in other words we require that it cannot be executed in different ways.

We will see now that there can be other constraints that must be taken into account. Take as an example the workflow where a security officer moves through a corridor. We can assume the whole movement to be a micro-task with some fixed duration and tolerances. If however an accident or an abnormal behavior happens and intervention is to be expected, we should allow for some granularity in the micro task splitting for efficiency reasons. Suppose that the corridor can be accessed from two doors at the beginning or the end of the corridor. It makes sense to split the micro-task in two smaller. The first is the movement from the beginning to the middle of the corridor, and the second is the movement from the end to the middle of the corridor. In this way, we allowed for ad hoc rules that subdivide the workflow. We can re-formulate the above procedure as the definition of three checkpoints, one at the beginning, one in the middle, and one at the end. For this reason, we allow the following micro-tasks.

Now let us try to generalize our approach, when two security officers are sweeping the corridor back and forth into opposite directions. We can require a rule table like the following Table 1. We see that the allowed tasks and micro-tasks are the same in number. However a micro task of the form *<O1 Approaching Ch1, O2 Approaching Ch1>* during execution of task 2 can happen (the two officers are talking about sports), and triggers an abnormal behavior alarm.

Table 1: Security Officer Workflow.

Task	Check point	Micro-task	Timing (sequence number)
1	1	Approaching Check point 1 (Ch1)	1
2	1	Leaving Ch1	1
2	2	Approaching Ch2	2
3	2	Leaving Ch2	1
3	3	Approaching Ch3	2
4	3	Leaving Ch3	1

We see that the allowed micro tasks are in reality 12, 6 per officer and for this reason the micro-tasks should not only describe normal, but also abnormal behavior.

In more complex environments, the number of micro-tasks can become overwhelming. The big number of micro-tasks hints towards a distributed implementation, while the synergy in sequencing them in order to infer task execution hints towards an agent based approach.

The solution we described above can be scaled to industrial environments as a coarse way to identify useful workflows with the help of computer vision. It is true though that there are limitations on the size and type of objects that can be efficiently identified.

Table 2: Workflow with two Security Officers.

Task	Check point	Micro-task	Timing (sequence number)
1	1	O1 Approaching Ch1, O2 Approaching Ch3	2
2	1	O1 Leaving Ch1, O2 LeavingCh3	1
2	2	O1 Approaching Ch2, O2 Approaching Ch2	2
3	2	O1 Leaving Ch2, O2 LeavingCh2	1
3	3	O1 Approaching Ch3, O2 Approaching Ch1	2
4	3	O1 Leaving Ch3, O2 Leaving Ch1	1

For this reason, we expect that a real solution would also need the input from various other sensors. As a result, we offer a uniform approach. This has the welcome side effect that our system is decoupled by the inaccuracies injected by the various sensors, especially the computer vision provided information. It is our feeling, that these inaccuracies can be improved by use of more inputs (eg. more cameras and related sensors), something that can be done in practice. In our approach, we concentrate especially on the computer vision sensory input provided by cameras and processed by computer vision algorithms like (Zhu et al., 2006) and (Viola and Jones, 2001). The resulting outputs of the sensors are usually in the form of bounding boxes and some ids. Our goal first is to create check points which act as “visual traps”. We put visual checkpoints in places that are relevant to the workflow and the accidents. This listing strongly depends on the workspace, but can be calibrated by the administrator without having specific knowledge of the computer vision algorithms used. Objects, check points and the attitudes of the objects with respect to checkpoints (Leave/Approach) are the input to our multi-agent system.

4 MULTI-AGENT BASED SOLUTION

As we argued previously, the problem at hand hints towards a multi-agent based solution. The synergy between checkpoints and their autonomy in making inferences shows that they can be implemented by multi-agents. Moreover, the structure of them is clearly the same and from the software point of view they are processes that can be created or deleted by the administrator, they can be migrated to adapt to the computational resources, but more importantly they are identical. They run the same algorithm, having the same memory but their decisions and actions rely on the actions of the other processes and their data. We cannot think of a better solution other than one provided by multi-agent systems. One can argue that the problem is generic and has no particular relation to an industrial environment. This could be true, however industrial environments are more constrained, have clearer workflow definition and the earnings of the approach translate to significant operational cost reduction with increased productivity. For this reason, though our approach has not special industrial characteristics at first sight, it is suited and could potentially be reliable.

But there is another reason why checkpoints (agents) are particularly useful for industrial environment. Usual approaches rely on scene separation algorithms, in order to identify tasks (Coros, 2009) or to improve human detection. This separation is too coarse to be of any value in an industrial environment where scene separation is very tight and the system reliability depends on it. Moreover, the checkpoints are visual points that can be synchronized between cameras, in an offline phase or online (Yun and Park, 2006), without interference to workflow execution. Use of RFID tags is usually not an option in an industrial environment and excessive use of other types of sensors can be dangerous. Finally, since they are visual, the administrator can manage them easily because they have better semantic content. So we reformulate the problem to adapt to the industrial environment and the new formulation translates immediately to agent architecture.

5 SYSTEM ARCHITECTURE

As communication architecture, we select a bidirectional ring (Figure 1), where agents can send and receive information between them, in other words the communication topology is a broadcast medium. The filtering of the messages is based on their content. The messages are of the form of Table 3.

Table 3: Agent message format.

Field	Work-flow Id	Check point Id	Micro-task Id	Alarm	Dest.
Value	Integer (Int)	Int	Int	Boolean	Int

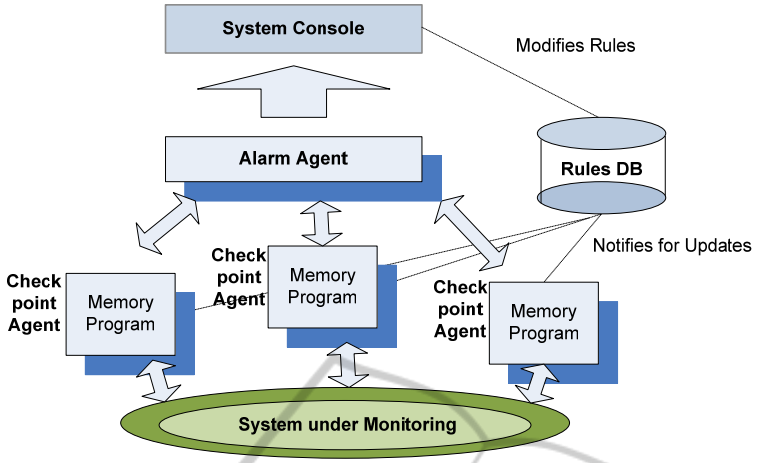


Figure 1: System Architecture.

It may happen that the workflow can have repeated object motion with respect to the checkpoint. As we can see from the above message, this can be dealt because the micro-task *id* is the sequence number of the micro-task in the workflow (see Table 4). This is possible because the system has two designated components (Figure 1).

The first is the workflow (rule) database that is loaded to the agents at system startup/reset. This is necessary since the administrator can create or destroy checkpoints. For this reason, at system startup or reset, cached agents are destroyed or new one are created according to the surveillance necessities. After the agent fixation process, each one is loaded the rule table via broadcast.

The second designated component is the alarm generator (Figure 1). We are now ready to describe the agent algorithm and memory after initialization.

Table 4: Workflow entries in workflow list.

Field	Work-flow Id	Sequence number	Check-point Id	Micro-task Id	Time out
Value	Integer (Int)	Int	Int	Int	Time

Table 5: Micro-task entry into rules.

Field	Micro-task Id	Object Id	Object Motion
Value	Integer	Integer	Leave/Enter

A micro-task (Table 5) is a list of pairs of the form (*Object Id*, *Object Motion*) with respect to Checkpoint. Different objects can create different micro-tasks, the role of the *Object Id* is to create a binding of objects.

We say that *O1/Leave-O2/Enter* and *O3/Leave-O4/Enter* realize the same micro-task, *1/Leave-2/Enter* with the binding (1:O1, 2:O2) and (1:O3, 2:O4). The binding happens at the first node which activates the workflow for monitoring. The algorithm description makes this clear (Figure 2). It can be easily proved that this algorithm that manipulates the workflow recognition system is correct and its results are presented in the following sections of the article through the simulation screenshots of the Sniffer JADE tool.

Moreover, it encodes the intention of the administrator to resume a workflow, if the alarm is not fatal or to issue an alarm. We can observe that the above algorithm does not interfere with disjoint workflows, because of the aforementioned binding. Also, it should be noticed that the algorithm takes advantage of the synergistic nature of the agents to solve the problem.

6 IMPLEMENTATION & WORKBENCH

The algorithm we have presented is generic and easily implementable using a standard agent platform like JADE. However, we have to specialize the micro-task description in the cases under consideration. In constraint environments micro-tasks can be comprised from lists of many people since tasks are very specific. In other scenarios, like market surveillance systems, multi-object micro-tasks are more difficult to obtain, because there are questions on bindings. This should be either manually determined by the administrator, or automatically by extra information. We decided to emulate a scenario depicted in the well-known Caviar dataset (EC Funded CAVIAR web site).

In specific, the scenario *"Person stops outside store, goes inside and out of store. Five groups of*

people walking along the corridor." and *"Person goes inside and out of a store twice. Visible on corridor view only: Group of 4 people comes out store"*. Since this scenario uses fixed cameras we decided to implement it at our laboratory with suitable camera placement. We also wanted to be able to use as many people as possible, for future experiments and extensive debugging. We used two cameras like in the original scenario, and we emulated the event of a human object entering the shop, with the use of the laboratory door.

One camera faces the door from 5.5 meters, and the other camera monitors the entering and leaving the store events perpendicularly to the first. The second camera was placed at distance 2 meters.
The workflow definition and the related points in the working environment, where workflow triggers are activated, are presented in the following Figure 3, using the colored dots on the figure. We have used two cameras that trigger the steps and the related events in our multi agent platform, where the recognized persons in scenes are giving input signals to our algorithms, for evaluating and checking the related workflow rules and the workflow recognition label. Figure 3 shows the configured check points, while Figure 4 shows an emulated customer entering and leaving the store and micro-task events, while passing the check points.

Figure 5 shows a sample of the exchanged messages for this workflow, this is a normal workflow. In order to decouple the problem from computer vision, we used manual annotation of the events. Our implementation is done in JAVA with the JADE platform.

Algorithm (agent software)
Trigger: Agent N senses a realized micro-task.
Action: The agent searches the list of activated workflows to find those that wait on the micro-task. Those found are now waiting for the next entry in their list if the corresponding checkpoint is the same. Otherwise they are removed and create an active set. For each member of the active set a message is sent with alarm false, workflow Id equal to the member workflow Id, checkpoint and micro task id the next in the workflow. If the sensed micro-task is not pending for some workflow, we activate the workflows that have it as a first micro-task happening at this checkpoint. The workflow waits on the realized micro-task. The agent is self-retriggered for this micro-task. If nothing is found a message is sent to the alarm agent.
Trigger: Agent N senses a timeout for an active workflow.
Action: A message containing the corresponding workflow entries for the previous different checkpoint in the workflow list and alarm true, is forwarded to the previous different checkpoint. The workflow is deleted.
Trigger: Agent N receives a relevant message with alarm true from a check point.
Action: The agent searches the active sets for pending messages from the corresponding checkpoint. The members are deleted. A message containing the corresponding workflow entries for the previous different checkpoint in the workflow list and alarm true, is forwarded to the previous different checkpoint. If deletion empties the active set an alarm is issued towards the alarm agent.
Trigger: Agent N receives a relevant message with alarm false from a check point.
Action: The agent activates the workflow contained in the message and the activated workflow waits for the entry in the message. The object binding is copied from the message.

Figure 2: Algorithm for manipulating workflow events.

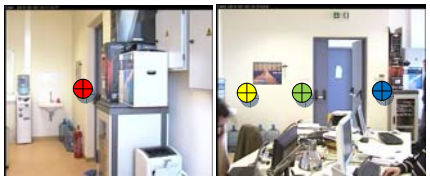


Figure 3: Camera 205 (*Trap point T4*, red dot), Camera 206 (*T1*, blue dot), (*T2*, green dot), (*T3*, yellow dot).

Empty micro-task micro-task Empty micro-task Empty micro-task

Figure 4: Scene frames presenting related workflow micro-tasks per camera.

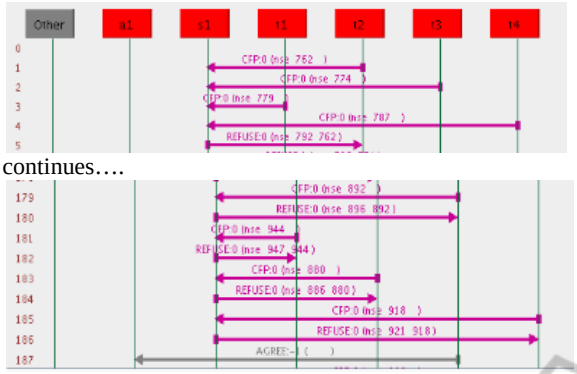
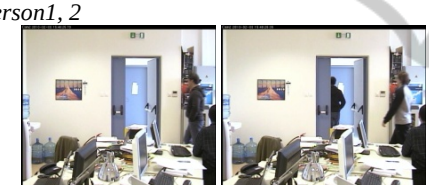


Figure 5: A normal workflow announcement to the *Alarm Agent* using JADE Sniffer tool.

Figure 6 shows a typical sequence of events for four people entering the shop. Of course, the applicability and the related workbench tests can be performed in any computer vision video, but for the paper purposes and tests of our algorithms, the normal enter – leave sequence of events in parallel with the computer vision human/objects detection was used.

Enter Person1, 2



LeavePerson1, 2

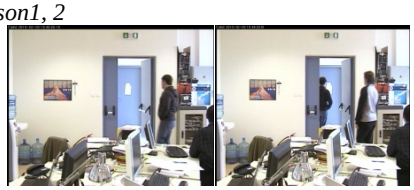


Figure 6: Enter-Leave micro-tasks for two human objects at checkpoint 1.

Figure 7 shows the message exchange. This is a normal workflow. In the first part of
Figure 7 7, enter events are shown between the *Sensor* agent which multiplexes sensor readings. In a real life implementation there is no *Sensor* agent, since it is part of the *Checkpoint* agent. In the second part of
Figure 7 7, the checkpoint 1 detects a leave event and forwards the request to checkpoint 4 (the perpendicular camera) according to case 4 of our algorithm.

For the purposes of display, we have implemented a small set of rules part of them are shown in Table 6. The xml file used as a rule engine, was constructed based on a simple format of object id plus object behaviour (enter or leave).

Table 6: Sample workflow rules.

```
   <?xml version="1.0"?>
   <?xml-stylesheet type="text/css"
href="nutrition.css"?>
   <listOfRules>
   <rule><cluster rulePos="1" trapId="1">
    <utask objectId="1"
objectBehavior="TRUE"/>
    <utask objectId="1"
objectBehavior="FALSE"/></cluster>
    <cluster rulePos="2" trapId="4">
    <utask objectId="1"
objectBehavior="TRUE"/>
    <utask objectId="1"
objectBehavior="FALSE"/></cluster>
    ...
```

Both Figure 5 and Figure 7 are presenting the agents simulation results in JADE. The last message on the agents simulation creates a ‘normal’ or ‘abnormal’ output, something that later can drive any extension of our system for safety and security purposes using multi agents implementations.

In the agents’ simulation windows used in this article, we used four check points in our application which are described as *t1*, *t2*, *t3*, *t4*. The alarming agent is described as *a1* and we describe the emulated *Sensor* agent as *s1* (it is only one since it multiplexes the sensor readings). The scope of the *Sensor* agent is to control and furthermore drive the signals in system output or outputs according to the connected on our system external modules. As we have already explained above the system scope is to drive safety and security external modules.

In the following figures, that describe message exchanges between agents, we re-used existing messages from JADE framework. Figure 8 shows the scalability properties of our architecture. We plot RAM consumption with increasing number of check points (traps).

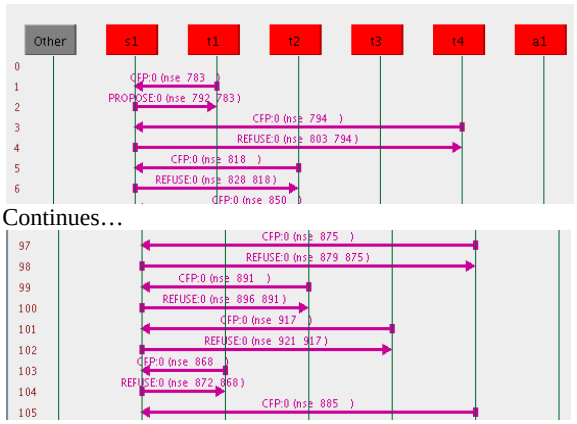


Figure 7: A normal workflow with 4 human objects. First part shows enter phase and second part leave phase with message propagation to the next agent.

As it is expected, it is almost linear which gives us predictable performance in case we need a mapping to the available resources. Our system was tested using between 3-15 trap points in the related lab videos. The system performance in multiple agents usage (multi camera control) and under more trap points per camera (agent) for a workflow of humans, is an interesting test case.

Test case measurements are very complicated and system performance is based on many different type criteria. For our simple cases the system used for the lab test followed the Figure 8 results.

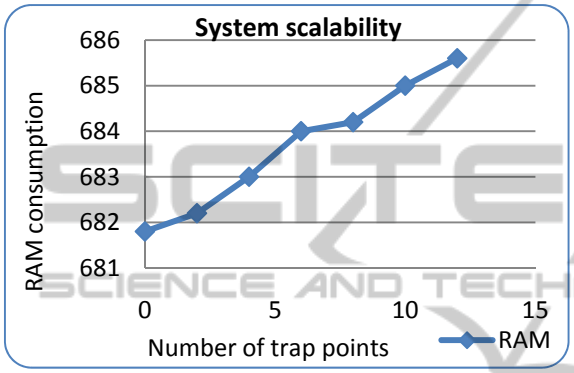


Figure 8: Scalability of the proposed architecture.

7 CONCLUSIONS

Through this paper, we touched upon what seems to be an appropriate framework for human workflow analysis. We presented a few test cases where our approach gives satisfactory answers using multi agent systems architecture. We are in the start of analyzing the challenges of workflow recognition using “traps”. We see vast potential in recognising single-object workflows by applying an agent for controlling the related trap points. The multi-object/human workflows are also common and they are the main target of our current and future research.

ACKNOWLEDGEMENTS

European Research project FP7 SCOVIS (ga. 216465).

REFERENCES

WfMS: Workflow Management Coalition Terminology and Glossary – WfMS Specification, 1999.

Purvis, M. A.; Savarimuthu,; Purvis, 2004. A Multi-agent Based Workflow System Embedded with Web Services. In Proceedings of the second international workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments (COLA 2004); Ghorbani, A.; Marsh, S., Eds.; IEEE/WIC Press: Beijing, China.

Vital, J., M., Buhler, P., and Stahl, C., 2004. Multiagent systems with workflows. IEEE Internet Computing, 8(1): pp. 76-82, Jan/Feb 2004.

Fleurke, M., Ehrler, L., Purvis, M. A., 2003. JBees – An Adaptive and Distributed Agent Based Workflow System. In Proceedings of the International Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments (COLA 2003), Halifax, Canada. IEEE/WIC Press. Ghorbani, A. and Marsh, S., Ed.

Savarimuthu, B.T.R., Purvis, M.A., and Fleurke, M. 2004. Monitoring and Controlling of a Multi-Agent Based Workflow System. In Proceedings of the Australian Workshop on Data Mining and Web Intelligence (DMWI2004), Conferences in Research and Practice in Information Technology, Vol. 32, Australian Computer Society, Bedford Park, Australia, pp. 127-132.

Poggi, A., Tomaiuolo, M., and Turci, P., 2007. An Agent-Based Service Oriented Architecture. WOA 2007, pp. 157-165.

Mok, W.Y., Palvia, P., and Paper, D., 2006. On the computability of agent-based workflows. In Decision Support Systems, Volume 42 , Issue 3 (December 2006), pp. 1239 – 1253, Elsevier Publishers B. V.

Buhler, P.A., Vidal, J.M., 2005. Towards Adaptive Workflow Enactment Using Multiagent Systems. Information Technology Management 6(1), pp. 61-87.

Negri, A., Poggi, A., Tomaiuolo, M., Turci, P., 2006. Agents for e-Business Applications, in Proc. AAMAS 2006, Hakodate, Japan.

Zhu Q., Yeh M.C,,Cheng K.T. and Avidan S. 2006 Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2006), Vol. 2, IEEE, pp. 1491-1498.

Viola, P. and Jones, M. 2001 Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2001), Vol. 1, IEEE, pp. 511-518.

Coros K. 2009 Video Shot Selection and Content-Based Scene Detection for Automatic Classification of TV Sports News, Advances in Soft Computing, Vol. 64, Springer, pp. 73-80, 2009.

Yun J.H. and Park R.H. 2006 Self-Calibration with Two Views Using the Scale-Invariant Feature Transform. Lecture Notes in Computer Science Vol. 4291 Springer , pp. 589-598.

EC Funded CAVIAR project/IST 2001 37540, http://homepages.inf.ed.ac.uk/rbf/CAVIAR/