22 AGENT-BASED INTELLIGENT MEDIA DISTRIBUTION IN ADVERTISEMENT

Raafat Aburukba[1], Hamada Ghenniwa[1], Weiming Shen[1, 2]
[1] *Department of Electrical and Computer Engineering*
University of Western Ontario, London, Ontario, Canada
[2] *Integrated Manufacturing Technologies Institute*
National Research Council Canada, London, Ontario, Canada
roaburuk@engga.uwo.ca; hghenniwa@eng.uwo.ca; weiming.shen@nrc.gc.ca

Networked electronic displays provide unique and effective capabilities for businesses and industries to communicate with their consumers. Today there are many solutions for static distribution of media-contents. However, delivering dynamic media-contents remains a major challenge. The overall goal of this work is to provide an intelligent media-content distribution, by which business intelligence is utilized to strengthen the business-customer relationships and increase profitability. This paper proposes a multi-agent approach to model the dynamic scheduling problem in media-content distribution. The proposed approach is validated through a prototype implementation.

1. INTRODUCTION

Improved consumer addressability, especially in electronic-based markets, allows companies to send focused promotional messages to specific customers, facilitating a targeted advertising approach. Networked electronic displays provide unique and effective capabilities for businesses and industries to communicate with the public.

Today there are many solutions for static media-content distribution in content-based network domains such as advertisement where a human user schedules media-contents to specific media displays at a specific time. The current approach requires a tremendous amount of work specially if there exist a large number of media-content and displays distributed across the country or the world. Also, if we consider the dynamic behavior of the content-based domain such as, media displays breakdown, the arrival of new media-contents, introduction of new media displays, and media-contents removal, then previous schedules might become infeasible due to the unforeseen changing situations. Therefore, finding coherent schedules in the content-based network environment where disturbances may occur at anytime is a major challenge.

In this work, we look at the content-based network problem as a scheduling problem, where scheduling can be described as the allocation of tasks to capable resources. Most scheduling models can fit in a 4-element structure, which consists of resources, tasks, constraints, and objectives:

- **Resources**: physical/logical devices capable of processing tasks;
- **Tasks:** a set of operations which are physical/logical processes, to be executed by a resource;
- **Constraints:** a set of conditions which must be satisfied. Constraints may be operation-based, task-based, resource-based, or a combination of both;
- **Objectives:** decision criteria which can be modeled as to maximize or/and minimize some function.

To illuminate confusion we will use tasks and resources for media-contents and media displays respectively in the rest of this paper.

In this work, we focus on an instance problem in content-based networks where specific locations of resources are used to display third party contents. By third party, we mean that a business rents a specific time slot from a content-based network provider to display their contents. This paper models the scheduling in the distributed content-based network environment and deal with unforeseen events such as resource addition and removal, and task additions and removal. We present an architecture that provides intelligent scheduling and handles changes in the media-content distribution environment. Previous work on dynamic scheduling has been investigated, and we strongly believe that agent-orientation is an appropriate design paradigm for distributed scheduling in the content-based network environment. The application of the agent for dynamic scheduling integrates the following:

- A negotiation protocol for scheduling to allow agents to cooperate and coordinate their actions in order to define a feasible schedule that satisfies both resource and task objectives and to handle the presence of real-time events.
- Flexibility and scalability where it is possible to dynamically introduce new resources and tasks, or remove existing resources and tasks without interrupting currently scheduled tasks in the environment.

The remainder of this paper is organized as follows: Section 2 describes related work; Section 3 introduces the scheduling problem model; Section 4 provides our dynamic scheduling model and approach; Section 5 introduces our implementation, and Section 6 provides a brief conclusion.

2. RELATED WORK

Previous work is done to transport multimedia contents under heterogeneous clients and network connections. The work in 7 is based on an approach that provides a descriptor schemes. Those descriptor schemes are used for matching multimedia contents with respect to the user profile for personalized video delivery on the web. The matching determines which parts of the video to include and what hotspots and hyperlinks to provide to the user. Their application supports personalization of individual sequences within an interactive video presentation, as well as per video-clip payment of interactive video. Another work in 12 combines video indexing techniques to parse TV News recordings into stories, and information filtering techniques to select stories to construct automatically personalized TV news programs. They formalize the selection process as an optimization problem. Experiments showed that a simple heuristic can provide high quality selection with little computation.

Generally, scheduling problems which are concerned with finding optimal schedules are subject to a limited number of constraints, and have a combinatorial explosion of possible solutions and are generally NP-hard 56. Early scheduling approaches involved finding the optimal schedule, but they were typically not efficient when the problem size grew. The majority of the research work in the scheduling area has been devoted to either simplifying the scheduling problem to the point where some algorithm can find optimal solutions, or to devising efficient heuristics for finding approximate solutions. The solution methods form two distinct classes: exact methods and heuristic methods. Exact methods guarantees finding an optimal solution. Such methods include: mathematical programming, dynamic programming, and branch and bound. Heuristic solutions in the other hand do not guarantee the optimal solution, but typically assure some degree of optimality when compared to the optimal. Some of the meta-heuristic techniques include: Simulated Annealing, tabu search, and genetic algorithms. Meta-heuristics are high-level heuristics that guide local search heuristics to escape from local optima.

Dynamic scheduling refers to the ability to adapt the schedule to the new situation by using appropriate actions to handle each event 17. The work in 11 used simulation to investigate the performance of various schedule repair heuristics (such as: no rerouting, queue rerouting, arrival rerouting, all rerouting) for unexpected resource failures in dynamic job shops. The experimental results showed that the proper selection of a good schedule repair heuristic is based not only on the system characteristics (utilization, resource down times, and frequency of resource failures) but also on the material handling system in terms of speed and the number of material handling system devices. Another study in 8 presents a simulation based analysis of different proposed dispatching rules for scheduling in job-shops with resource breakdowns. The results of the simulation study revealed that the relative performance of scheduling rules can be affected by changing the breakdown parameters. A work in 13 presents several rescheduling strategies are proposed for process time variations, resource breakdowns, and new task arrivals in a dynamic environment. Monitoring the environment is performed periodically and either rerouting to alternative resources or order splitting policies are activated in response to unexpected disruptions.

Agent technology has recently been applied to solve distributed scheduling problems including distributed manufacturing scheduling, transportation scheduling and computing load balancing **2019**18. Many agent-based scheduling in distributed systems adapt economic-based approaches such as contract-net **19**3 which is based on the tender economic model. More complex approaches include combinatorial auctions 16 for many-to-many negotiation problems.

In a content-based network environment such as advertisement, it is highly desirable to employ dynamic scheduling. The dynamic nature of scheduling can be viewed as:

- the process of updating an existing schedule in response to a disruption 21
- the process of generating a new feasible schedule upon occurrence of a disruption 1

3. MODELING THE SCHEDULING PROBLEM

3.1 Scheduling Problem in Content-Based Networks

In a content-based networks domain, a system with *m* resources can be denoted by $\mathbf{R} = \{R_1, R_2, ..., R_m\}$. A resource is a physical/logical device that is capable of executing some operations. In this context, a resource can be formally defined as a set of operations $R_i = \{o_i^1, o_i^2, ..., o_i^{nj}\}$, where n_i is the number of operations that R_i can process. An operation is a physical process to be executed by a resource in the content-based domain such as media-contents or tasks. The overall operations that can be performed by the system **R** is equal to $\bigcup_i^m R_i$. Partially Overlapping (PO) systems consist of resources, each of which is capable of performing a specific set of operations that may overlap with those of other resources. For these systems, the following two conditions must hold:

1) There exist at least two resources $R_i, R_j \in \mathbf{R}$, $1 \le i, j \le m, i \ne j$, such that $R_i \cap R_j \ne \phi$;
2) There exist at least two resources $R_k, R_l \in \mathbf{R}$, $1 \le k, l \le m, k \ne l$, such that $R_k \ne R_l$.

Given a set of *n* tasks in the content-based networks, denoted by $\mathbf{T} = \{T_1, T_2, ..., T_n\}$, where task $T_j \in \mathbf{T}$ (*j=1…n*) is formulated as a set of operations $T_j = \{o_j^1, o_j^2, ..., o_j^{nj}\}$, where n_j is the number of operations belonging to T_j. The tasks need to be performed by **R**. The scheduling process involves the allocation of tasks among resources at specific times. Each task T_j may have its operations processed by a single resource R_i from a group of resources, in which each resource can process all operations of T_j. In partially overlapping systems structure 10, the instance problem in content-based network domain focused on this paper falls under the OR-structure where a task is to be processed by any resource in **R**. The content-based network scheduling problem in PO systems has the following structure:

- Resource capabilities are sufficient. All operations of the tasks can be processed in the system, that is $\bigcup_{j=1}^n T_j \subseteq \bigcup_{i=1}^m R_i$
- Task T_j is to be processed in a resource R_i that has the capabilities to process all the task's operations (PO system OR-structure)
- Tasks have objectives to be achieved in the domain
- Resources have objectives to be achieved in the domain
- No pre-emption. Each task, once started, must be completed before another task may be started on a resource
- No resource may process more than one task at a time
- Each task has a hard start time, and end time
- tasks can have sequence constraints
- tasks can have location constraints

3.2 Modeling Content-Based Scheduling Problem

In this section, we model a content-based instance of scheduling tasks to resources owned by a content-based network provider to meet the tasks' and resources' objectives. In this instance, we focus on two objectives, maximizing the profit and minimizing the cost. These objectives are useful for the content-based network business owners to maximize their profit, and for the tasks' business owners to find the minimum cost to display their tasks.

3.2.1 Maximizing the Profit

The basis of modeling the profit is based on the formula: *Profit=Revenue – Cost*. The cost in this formula is based on the price it costs the business to provide the media-content distribution scheduling service. This cost is created by the media-content network provider based on an analysis to their expenses. Here, we will focus on increasing the revenue since the cost in the profit function is presented as a constant value. To increase the revenue, we maximize the resource utilization, since by having the resource idle, the content-based network business provider will not have any revenue, and therefore, the profit will decrease. In this instance, each time period to promote a task differ from other times which is set by the content-based network providers. For example: the cost to display a task during rush hour in busy streets is more expensive than displaying a task after mid-night. The mathematical formulation to maximize the resource utilization is:

$$\max\left\{\sum_{j=1}^{n} x_{ij} p_{ij}\right\} \quad (1)$$

s.t

$$\sum_{j=1}^{n} x_{ij} p_{ij} \le T_i \quad (1.2)$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad (1.3)$$

$$x_{ij} \in \{0,1\} \quad (1.4)$$

In the above formulation, *n* presents the number of tasks, and *m* presents the number of resources. T_i is the completion time of tasks scheduled on R_i, and $\sum_{j=1}^{n} p_{ij}$ is the total processing time of *n* number of tasks in resource R_i. x_{ij} is a variable that is set by constraint 1.3 and it can be either 1 or 0. If $x_{ij} = 1$, task *j* is to be processed by resource *i*.

3.2.2 Minimize the Cost

The second objective we are looking at in this work is to minimize the cost. The mathematical formulation to minimize the cost is:

$$\min\left\{\sum_{i=1}^{m} c_{ij} x_{ij}\right\} \quad (2)$$

s.t.

$$\sum_{i=1}^{m} x_{ij} = 1 \quad (2.1)$$

$$x_{ij} \in \{0,1\} \quad (2.2)$$

In this objective function, *m* represents the number of resources, c_{ij} is the cost price to execute task *j* in resource *i*. x_{ij} is a variable that is set by constraint 2.1 and it can be either 1 or 0. If $x_{ij} = 1$, task *j* is to be processed by resource *i*.

4. CONTENT-BASED SCHEDULING APPROACH

4.1. Business Rules

The Business Rules Group 2 describes a business rule as a statement that defines or constrains some aspect of the business; a business rule is intended to assert business structure or to control or influence the business's behavior. A rule engine evaluates and executes rules, which are expressed as if-then statements. The power of business rules lies in their ability both to separate knowledge from its implementation logic and to be changed without changing source code 14. A rule is composed of two parts, a condition and an action: When the condition is met, the action is executed 14.

In this work, we use the concept of business rules to capture precise business logic in resources, tasks, and time to govern the behavior of the scheduler. We propose three types of business rules:

- **Resource Business Rule**s are related to the resource location such as marketing zone. In our proposed solution, each resource and task has a business rules associated with it. Assigning business rules to both tasks and resources will govern the scheduler to find the right location for the task.
- **Tasks Business Rules** are related to tasks such as baked products. In our proposed solution, each task is associated with tasks business rules. This association governs the scheduler to produce tasks sequence or order.
- **Time Business Rules** are related to time such as breakfast time. In our proposed solution, time intervals, and tasks are associated with time business rules. This association governs the scheduler to assign the task at a specific time interval.

4.2. Proposed Solution

In a dynamic environment, scheduling usually involves complex and non-deterministic interactions between different participating tasks and resources. We strongly believe that agent-orientation is an appropriate design paradigm for scheduling in a dynamic environment. Indeed, such a paradigm is essential to model:

- The dynamic structure of the environment where it is possible to dynamically integrate new tasks, and resources, or remove tasks and resources without disrupting schedules previously established and re-initialising the environment.
- Multiple and distributed objectives and constraints: each resource can be geographically distributed, where each resource can have its objective and constraints, and solve the scheduling problem locally. Hence, distributed and autonomous scheduling systems are more appropriate than centralized and non-autonomous scheduling approaches.

A key aspect of agent-orientation is the ability to design artefacts that are able to perceive, reason, interact and act in a coordinated fashion. Therefore, unlike the traditional way of having a centralized scheduler, an agent based scheduling system supports distributed scheduling where each resource agent can find a schedule that satisfies its own objective. However, agents collectively find a schedule that satisfies both task and resource objectives. In our proposed approach, we model scheduling as distributed, autonomous and collaborative problem.

- **Distributed:** By distributed we mean that each resource is geographically distributed and responsible for finding a solution of the scheduling problem based on its objective of maximizing the profit.
- **Autonomous:** Each resource is independent from other resources. By independent we mean that when a resource finds a schedule that satisfies its objective to maximize the profit, the schedule does not depend on other resources.
- **Collaborative:** Multi-objectives between tasks and resources

Our proposed solution models the scheduling problem as a multi-agent system. We introduce three types of agents:

- **Interface Agent:** this agent gathers the information of the tasks and sends a request to the manager agent. The information gathered includes: tasks features, associated business rules, and the objective to be achieved.
- **Manager Agents**: represent the resources within its domain. Manager agents are capable of generating a local solution to the schedule based on its objective.
- **Broker Agent:** provides the scheduling functionality to produce a global solution that satisfies the tasks' objective function.

Given the nature of the content-based environment of being distributed and independent, we divide finding a solution to the multiple objective scheduling problem into local and global solutions.

- **Local Solution:** this includes the manager agent view of the solution based on the resource based objective to maximize the profit.
- **Global Solution:** this includes the integral solutions generated by the broker agent in the proposed architecture within the context of the task based objective to minimize the cost.

To govern the scheduling search, we associated resources, tasks, and time to business rules. The resources are associated with resource related business rules. Time intervals are associated with time related business rules. Tasks are associated to resource, time, and task business rules. Having these business rules allow tasks to request for service at specific desired locations, at a specific time, and at a specific order or sequence.

4.3. Proposed System Architecture

The proposed architecture for distributed dynamic scheduling in content-based networks shown in Figure 1 involves the following agents: Interface Agents, Manager Agents, and the Broker Agent. The proposed multi-agent architecture allows manager agents to achieve their objective in scheduling locally. In addition, the manager agent and the broker agent cooperate using the contract-net negotiation approach 3 in order to find a global schedule. Focusing on the objective of minimizing the cost, and maximizing the profit, we present a solution to find a schedule that satisfies both task and resource objectives.

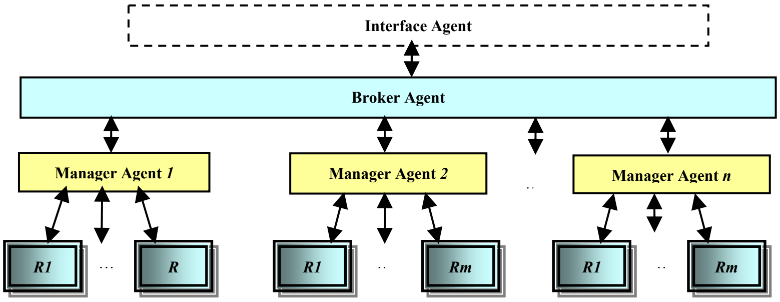

Figure 1 - High-Level System Architecture

Upon an existence of a resource within a domain, the manager agent is notified. The manager agent stores the resource existence and the resource capabilities in its knowledge base. The manager agent informs the broker agent with any updates. The information sent to the broker agent includes: the resource capabilities, and business rules of all the resources attached with the manager agent. The broker agent stores the information for the manager agents in its knowledge.

In case of a resource removal, the manager agent has knowledge of all scheduled tasks in the resources, and therefore,

- The manager agent removes the resource from its knowledge.
- The manager agent reschedules unprocessed tasks in the removed resource within its domain based on tasks' business rules.
- In the case of the manager agent can not find a schedule within its domain, it returns unscheduled tasks back to the broker agent to reschedule the specific tasks. The process of rescheduling these tasks is the same as scheduling new tasks mentioned below.

Upon receiving the tasks by the broker agent, the broker agent generates a match based on the task, time and resource business rules, as well as on the manager agents' capabilities. The match results include the resources capable to schedule the task, and the time interval to schedule the task. The broker agent interacts with the capable manager agents and the cost to execute the tasks in the resources at the specific time slots. Manager agents have the knowledge of the resources in its domain, as well as the scheduled tasks. The manager agents find the optimal schedule based on its objective to maximize the profit. The interactions between the broker agent and manager agents are as follows:

- The broker agent sends a request to all potential manager agents to schedule each task.
- Each potential manager agent finds a local schedule that maximizes the profit.
- Each potential manager agent submits a price to the broker agent.
- The broker agent collects the prices from manager agents and finds the global schedule based on minimization the cost objective.

In the case to remove a task, the system reacts as follows:

- The interface agent sends a request to remove the task to the broker agent.
- The broker agent sends a request to the manager agent that is scheduled to process the task.

- The manager agent cancels the task from the resource.
- The manager agent sends a confirmation to the broker agent.
- The manager agent reschedules unprocessed tasks based on its objective to maximize the profit.

5. IMPLEMENTATION

A prototype system has been implemented to provide scheduling in a distributed content-based environment. We used the JADE, a FIPA-complaint platform 9 in our agent implementation. Each individual agent (interface, manager, and the broker) in our proposed architecture is modeled using coordinated intelligent, rational agent (CIR-agent) 4. The internal structure of each agent is composed of knowledge and capabilities.

In general, the knowledge of the agent is stored in its local database. Each agent has access to its local relational databases via the Java Database Connectivity (JDBC). This allows the agents to query and access information from its local relational databases through a common platform-independent interface. The interface allows Java programs to interact with any SQL-compliant database. However, the capabilities of all agents in the environment include communication, and reasoning. Generally with all agents the communication capability allows the agents to exchange messages with other elements of the environment, including users, agents and objects. The implementation of this component takes advantage of the existing Agent Message System in JADE. The communication component of the agent's is equipped with an incoming message inbox, and message polling can be both blocking and non-blocking, with an optional timeout. Messages between agents are based on using FIPA ACL, in which different ACL performative are supported; the communication component is implemented as a set of object-oriented classes that inherit the *jade.Core.Agent* and *jade.lang.acl.ACLMessage* existing class of the JADE platform. The agent's reasoning capabilities include the problem solver, and interaction devices such as the assignment device. The problem solver in both the broker and the manager agents include the algorithm of the revised simplex optimization technique that finds the exact optimal value of the objective under the linear inequality constraints 15. As mentioned in Section 2, exact solutions are not efficient to solve large problems. However, the focus of this work is not to implement an efficient optimization algorithm. We have implemented the revised simplex optimization algorithm as a proof of concept in both the broker agent to find the global solution and the manager agent to find the local solution.

6. CONCLUTION

This paper presents an agent-based approach for intelligent media-content distribution. It considers the dynamic behavior of the system such as resource additions or breakdown, and task additions or removal. An important advantage of using the broker agent is that it allows the system to operate robustly when confronted with a resource or task agent's appearance or disappearance. Also, in a dynamic environment, the proposed broker agent can help tasks to locate distributed resources in an open environment. The proposed system can monitor any possible changes within the environment and performs scheduling based on these changes.

7. REFERENCES

1. Abumaizar, RJ, Svestka, JA. “Rescheduling job shops under random disruptions”. International Journal of Production Research, 1997; 35 (7): 2065-2082.
2. Business Rules Group. URL: http://www.businessrulesgroup.org/
3. Davis, R, Smith, R. “Negotiation as a Metaphor for Distributed Problem Solving”. Artificial Intelligence, 1983; 20: 63-109.
4. Ghenniwa, H. and Kamel, M. “Interaction Devices for Coordinating Cooperative Distributed Systems", Journal of Intelligent Automation and Soft Computing, 2000.
5. Gordon V, Proth J-M, Chu C. A survey of the state-of-the-art of common due date assignment and scheduling research. European Journal of Operational Research, 2002;139:1-25.
6. Graham, R.L., Lawler, E.L., Lenstra, J.K., and Rinnooy Kan, A.H.G., Optimization and approximation in deterministic sequencing and scheduling: A survey, Annals of Discrete Mathematics 5, 1979; pp. 287-326.
7. Hjelsvold Rune,Vdaygiri Subu,Léauté Yves, “Web-based Personalization and Management of Interactive Video”, Proceedings of the 10th International Conference on World Wide Web, Hong Kong, 2001, pp. 129-139.
8. Holthaus, O. Scheduling in job shops with machine breakdowns: an experimental study. Computers & Industrial Engineering,1999; 36 (1), 137-162.
9. Java Agent DEvelopment (JADE) Framework. URL: http://jade.tilab.com/
10. Kamel, M, Ghenniwa, H. “Partially-Overlapped Systems: The Scheduling Problem”. In Design and Implementation of Intelligent Manufacturing Systems, H. Parsaei and M. Jamshidi, Eds., Prentice-Hall, 1995, pp. 241-274.
11. Kutanoglu, E. and Sabuncuoglu, I.. Routing-based reactive scheduling policies for machine failures in dynamic job shops. International Journal of Production Research, 2001;39 (14), 3141-3158.
12. Merialdo, B, Lee, K. T, Luparello D, and Roudaire J. “Automatic Construction of Personalized TV News Programs,” ACM Multimedia, 1999.
13. Nof, S.Y., Grant, F.H., Adaptive/predictive scheduling: review and a general framework. Production Planning and Control 2 (4), 298-312. Society,1991; 41 (6), 539-552.
14. Qusay H. Mahmoud. Getting Started with the Java Rule Engine API (JSR 94): Toward Rule-Based Applications, 2005. http://java.sun.com/developer/technicalArticles/J2SE/JavaRule.html
15. Ronald L. Rardin. “Optimization in Operations Research”, pp. 236
16. Sandholm T. “Algorithm for optimal winner determination in combinatorial auctions”, Artificial Intelligence, 2002; No. 135, pp. I-54.
17. Saucer, J. “Knowledge-Based Systems Techniques and Applications in Scheduling”. In Knowledge-Based Systems Techniques and Applications, TL Leondes, ed., San Diego, Academic Press, 1999.
18. Shen W. “Distributed Manufacturing Scheduling Using Intelligent Agents”, IEEE Intelligent Systems, 2002, Vol.17, No. 1, pp. 88-94.
19. Sousa, P. and Ramos, C. A distributed architecture and negotiation protocol for scheduling in manufacturing systems. Computers in Industry, 1999;38 (2), 103-113.
20. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. and Peeters, P. Reference architecture for holonic manufacturing system: PROSA. Computers in Industry, 1998; 37 (3), 255-274.
21. Vieira, G. E., Hermann, J. W. and Lin, E. Rescheduling manufacturing systems: a framework of strategies, policies and methods. Journal of Scheduling, 2003, 6 (1), 36-92.