Complex Application Architecture Dynamic Reconfiguration Based on Multi-criteria Decision Making

Vincent Talbot (1) & Ilham Benyahia (2)

Université du Québec en Outaouais, 101 St-Jean-Bosco, Box 1250, Hull Station, Gatineau, Québec, Canada J8X 3X7

vtalbot@telask.com(1)
ilham.benyahia@uqo.ca (2)

ABSTRACT

Intelligent Transportation Systems (ITS) are increasingly important since they aim to bring solutions to crucial problems related to transportation networks such as congestion and various road incidents. Management of ITS, as other complex and distributed applications, has to cope with unforeseeable events and incomplete data while guaranteeing a quality of service (QoS) defined by multiple criteria reflecting real-life needs. To enable applications to adapt to changing environments, we define a methodology of dynamic architecture reconfiguration based on multi-criteria decision making (MCDM) using evolutionary computing (EC) to find the best combination of architecture components. We use the Pareto Evolutionary Algorithm Adapting the Penalty (PEAP), a category of EC, selected in this paper to deal with time-consuming online processing required by basic EC such as genetic algorithms. Our simulation results relating to road safety highlight the benefits of MCDM prior to such reconfiguration. We also address the problem of destabilization which can result from repeated reconfigurations in response to ongoing environment changes.

KEYWORDS

Complex applications, architecture performance optimization, architecture reconfiguration, multi-criteria decision making, Pareto Evolutionary Algorithm Adapting the Penalty (PEAP), road safety application.

1. INTRODUCTION

Emerging technologies in computing and telecommunications have brought valuable new dimensions to the development of complex applications in many domains including intelligent transportation systems. Traffic and routes are continuously monitored, and reports on their current states are transmitted to a central station where all the information is visualized and analyzed. Then, control data are transmitted to different locations within a specified deadline so that traffic systems can be adjusted according to new road conditions to avoid road congestion and to help

DOI : 10.5121/ijsea.2010.1402

drivers restrict their speed to avoid accidents. The software supporting the control processing requires an architecture that can be reconfigured in response to changing road conditions. To ensure that the processing components of a traffic management system are readily adaptive, our approach is built around dynamic reconfiguration of the software architecture. To support such reconfiguration, we set out a new approach to multi-criteria decision making (MCDM) based on the Pareto Evolutionary Algorithm Adapting the Penalty (PEAP) to improve upon the traditional slow genetic algorithm process. To reach appropriate final decisions related to reconfiguration we propose the integration of supervised learning or interactions with a decision maker to find the solution that best satisfies the specific objectives of the application being considered with PEAP. We also address the important problem of destabilization, which is a serious threat when frequent reconfigurations occur.

This paper is organized as follows. Section 2 provides background on related work involving the use of reconfigurable software architecture. Section 3 gives an overview of MCDM incorporating metaheuristics with a focus on transportation applications. Our approach to dynamic reconfiguration based on evolutionary techniques is detailed in Section 4. In Section 5, we present two scenarios specific to road safety illustrating the benefits of dynamic reconfiguration based on multi-criteria decision making. Section 6 presents a conclusion and future directions.

2. Related Work

Much recent research on distributed systems that interact with their environments has focused on dynamic reconfiguration and adaptive resource management [1], [2], [3], as means of optimizing and guaranteeing the required quality of service (QoS). However, premature triggering of reconfiguration may result in system instability and performance degradation because of the uncertainty created by frequent changes in the operating environment. To maintain good operation equilibrium, the tendencies of these systems must therefore be evaluated before the reconfiguration process is launched.

2.1 Dynamic Reconfigurable Frameworks

Dynamically reconfigurable systems typically incorporate component-based frameworks capable of modeling, managing and reorganizing their architecture with little or no human intervention. OpenRec [4] and Fractal [5], [6] are two such frameworks capable of introspection and extensibility. Both have recently been extended with formal specification metalanguages (Alloy and Focal, respectively), used to model systems and to prove dynamically that systems configurations are semantically correct and satisfy functional constraints.

The drawback of these reconfigurable systems is that they lack processing capability to deal with instability associated with frequent reconfigurations. Sophisticated tools such as UML-RT [7] designed for specification validation of real-time distributed systems are of limited value for dynamic validation of adaptive systems that interact with disturbed environments. Simulation frameworks are more appropriate for that purpose since various scenarios can be tested for performance validation. In this paper, we use a real-time, process-driven simulation environment called J-Sim [8] to evaluate the performance of each architecture configuration.

2.2 Management of Architecture Solutions Based on the Reconfiguration

In order to cope with changes in its environment, a self-adapting system must locate, discover or construct alternative configurations and select the most appropriate one for the current environment context according to QoS criteria such as deadline satisfaction and operating rate. The approach proposed in [9] is based on generating reactive plans (configurations) from goals expressed in temporal logic. Its three-layered conceptual model consists of a goal management layer, a change management layer and a component layer. In [10], alternative solutions for a given application domain are captured in a domain repository. Similar approaches are found in the field of cognitive radio [11] where hierarchical management is used to discover and construct dynamically new hardware and software architectures for cognitive radio systems. In [12] a generic cognitive framework is presented for autonomous decision making. Multiple possibly conflicting, and operational objectives are analyzed in a time-varying environment.

Our focus here is on the impact of dynamic reconfiguration on ITS applications, specifically those relating to the comfort, safety and security of motorists. As shown in [13], some traffic models may lead to congestion that can cause crashes. Clearly, one objective of reconfigurations must be to balance traffic loads. Another essential objective in traffic monitoring reconfigurations must be to enhance dynamic emergency vehicle dispatching systems.

3. Multi-criteria Decision Making (MCDM) for Dynamic Reconfiguration

Just as the configuration of architectures intended for complex applications including real-time systems must be validated, so the reconfiguration of these architectures must take into account the QoS parameters to be validated online. When the QoS is defined by a set of criteria and there are numerous alternatives for architecture reconfiguration, MCDM is essential.

3.1 Evolution of MCDM

Pioneer research in multi-criteria decision making (MCDM) dates back to the 1950s. MCDM then evolved in two main methodological directions, the first based on an outranking criteria approach to deal with heterogenous criteria and their associated scales [14], [15] and the second on multi-attribute utility theory [16].

Today, new approaches to MCDM take advantage of advances in information technologies applications to solve both theoretical and applied decision problems such as those encountered in ITS. One frequently used method is to reduce multiple criteria to a single criterion by aggregation. However, there is little research on combining multiple objectives in a more realistic way, especially when there is an absence of dominance or potential conflict between two or more criteria. Recent work on MCDM in the field of transportation has focused on solution research within large spaces of feasible solutions and involves the application of metaheuristics and learning techniques such as supervised learning, a category of evolutionary algorithms (EA) such as genetic programming [17].

3.2 Use of Metaheuristics in MCDM for ITS

Increasingly, multiple-objective metaheuristics is being applied to the solution of complex ITS problems such as vehicle assignment, routing and scheduling or crew assignment and scheduling. In this context, there is a pressing need to overcome major challenges in MCDM techniques for use online to manage highly frequent environment changes.
Many EAs deal with optimization problems complicated by performance requirements. One such example is the Strength Pareto Evolutionary Algorithm (SPEA) [18] which is based on the relative strength of individual solutions within a solution space. However, when used for complex applications with QoS requirements, MCDM optimization must be constrained by performance rules as well as semantic rules (e.g., correctness of the composed architecture).

Little research has addressed MCDM problems in dynamic and constrained applications. The main issue is that problem solving methods such as evolutionary techniques require numerous iterations to find an optimal combination yet there are temporal constraints for making online decisions that must be met. In [19], MCDM is applied in the dynamic reconfiguration of optical networks characterized by very high speed transmissions. An approach based on Pareto Evolutionary Algorithm Adapting the Penalty (PEAP) which is characterized by individual penalties is used in this context. The main purpose of multi-criteria optimization is to find the Pareto border defined by a set of potential solutions. In our work, we have therefore employed a hybrid technique that combines Pareto-optimal sets with penalty functions.

4. An Architecture for Dynamic Reconfiguration Incorporating MCDM

We outline here a methodology of architecture reconfiguration based on evolutionary techniques and incorporating MCDM. While the approach presented in this paper has been developed in the context of ITS, it is broadly applicable to complex applications that must meet QoS requirements while operating in a disturbed environment.

Our network management framework [20] centers around a set of software agents called Complex Agents (CAs) that are equipped with functions to monitor and control their environment. They are autonomous software entities which are capable of reconfiguring their internal processing architecture in order to maintain an optimum level of control and monitoring despite unforeseen changes in their environment. The architecture of a CA is based on a library of software components classified according to the processing cycle of environment events. Different components may serve the same end purpose, but with different performances. For instance, scheduling of real-time events can be carried out by a number of different algorithms such as first deadline, earliest deadline, time laxity, etc. [21].

The CAs rely on a sophisticated architecture reconfiguration management (Fig. 1). Parameters representing the performance criteria of each CA component will always include processing accuracy and the CPU execution time.

4.1 Reconfiguration Management Architecture

The central element of architecture reconfiguration management in our framework is the MCDM mechanism which reacts to significant environment changes and decides whether reconfiguration of the CAs is needed, and what reconfiguration should be applied in order to optimize conflicting objectives of processing performance parameters.

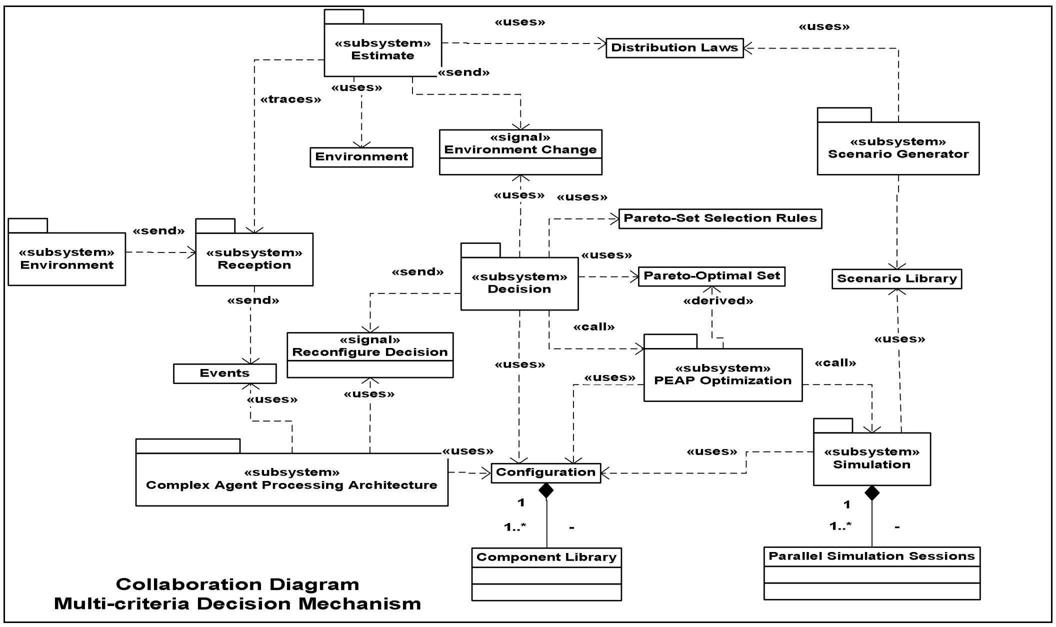

Fig. 1: Reconfiguration Management Architecture.

Under stable *Env1* network conditions, CAs receive a regular flow of monitoring events from their environment via the framework Estimate module. Nevertheless, it happens that the environment is disturbed due to incidents or unforeseen events in the monitored environment.

Fig. 2 illustrates a scenario following events that indicate significant changes in the network environment. At t1 a change from *Env1* to *Env2* is detected and a reconfiguration process is triggered. At t2 the Estimate module generates an estimatation of the type of change (stable or disturbed), its intensity and its duration. These parameters are sent to the MCDM module at t3 and a final decision to reconfigure and how (or not) is made at t4.

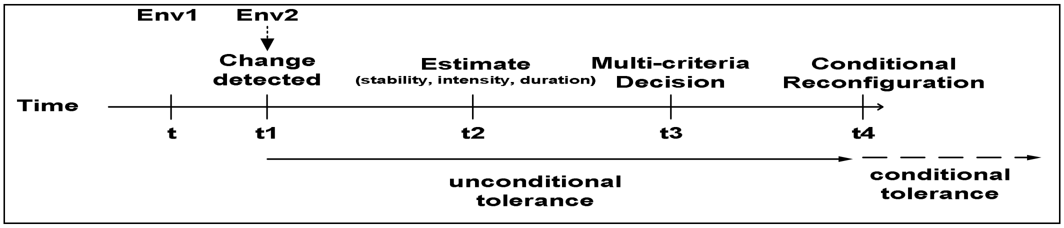

Fig. 2: Reconfiguration Management Steps.

4.2 The MCDM Process

At the core of the decision mechanism of our approach is a multi-criteria optimization process based on PEAP. The extent of valid configurations for a CA is great, and depends on the size of its library of components. PEAP searches for the best configuration by evolving a set of configuration solutions called a Pareto-optimal set (Fig 3).

Genetic algorithm operators in PEAP combine existing solution "individuals" to generate new ones. New individuals are assessed by measuring their fitness (quality of the solution) according to the new ITS environment. Only individuals with a superior fit are selected for the subsequent rounds until a final Pareto-optimal set is obtained.

The fitness for each individual is calculated using a quality value obtained by simulation (see Fig. 1) and a penalty value obtained by counting the number of compositional constraints violated by that individual. In PEAP, the penalty value reduces the fitness of an individual without discarding that solution completely. The decision to reconfigure is often granted since it is sometimes preferable to leave the CA as is if the perturbation has a weak intensity or is expected to be brief. Therefore the final decision is made in two steps:

1. A set of static rules based on the distributed system stability and duration of the environment disruption will determine whether the CA should to be reconfigured or not.

2. If reconfiguration is necessary, the Pareto-optimal set is divided into three sub-sets of equal size. Each subset presenting perturbation rate is matched with a low, moderate or high intensity value (represented by sigma). The individual located in the subset that matches the estimated intensity value is finally selected.

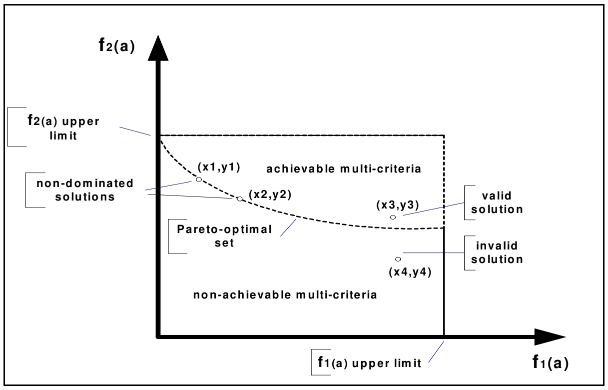

Fig. 3: Generation of the Pareto-optimal set using PEAP.

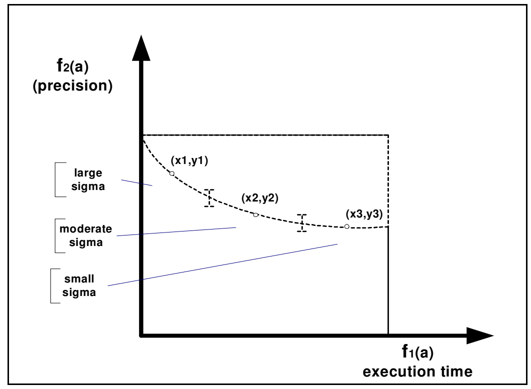

Fig. 4: Selection of a non-dominated solution using static rules based on sigma intensity.

5. Benefits of Decision Making Before Reconfiguration in ITS Applications

In order to validate the requirements of the decision process for dynamic architecture reconfiguration for complex applications, we have considered the application of road safety in the context of ITS applications.

5.1 Introduction to Road Safety

Main roads in cities around the world today are clogged with traffic. The extent of congestion underscores the inability of major road networks to cope with current traffic demands. Transportation planners are increasingly turning to ITS to ensure road safety through transportation network management systems.

Road safety applications of ITS will help drivers to make decisions that will improve traffic balance and avoid road incidents. Drivers can be advised to change their routes as quickly as possible to avoid congestions that may lead to incidents. Also, efficient emergent vehicle dispatching may lead to traffic incident managements and additional incidents avoidance.

5.2 Instantiation of our Framework by Road Safety Application

ITSs applications are made up of software and hardware components also called logical and physical components in the official ITS Architecture for Canada. Physical components, represented by transportation network components such as roads or junctions to be monitored, form the environment to be managed by the logical components. In this paper, vehicles, message signs, sensors and cameras are examples of hardware components, whereas software components are illustrated by traffic assessment, routing algorithms and vehicle dispatching algorithms.

According to the performance characteristics of our framework presented in Section 4, functions that compose each layer of the transportation network management system can be achieved by different algorithms characterized by different performance criteria. Performance is measured in terms of the computation time required to reach the final solution within a given confidence interval. Thus, the same function of an ITS may perform differently according to the environment.

Selecting the best algorithm presents a challenge. The algorithms shown in Table 1 all have the same function of learning (clustering or classification) used for traffic assessment. Two main criteria must be considered when measuring the performance of traffic assessment algorithms: 1) execution time for testing and 2) learning accuracy. Thus, MCDM is required to select the best algorithm among a set of algorithms achieving the same function.

Classification Algorithm	Avg Testing Time	Avg Learning Accuracy
K* algorithm	0.15 s	60 %
1B1 algorithm	0.05 s	48 %
Bagging algorithm	0.01 s	81 %

Table 1: Performance parameters of classification algorithms.

The overall management of a transportation network demands a variety of performance criteria including the accuracy of information communicated for interaction of its components, a short execution time for real-time control, and the reliability of information about the road environment. Since the full processing of information coming from the environment must satisfy two or more criteria at the same time (e.g., minimizing CPU time and maximizing the resulting confidence or accuracy), a multi-criteria analysis comparing alternative architecture configurations is required for an ITS management system as a precursor to dynamic adaptation and reconfiguration of the components in response to environment changes. We now discuss the impact of dynamic adaptation of ITS components by considering two scenarios related to road safety.

5.3 Scenarios Requiring Dynamic Architecture Reconfiguration

We consider the library of components of CAs presented in Section 4. When Estimate module (Fig. 1) indicates the presence of parameters representing congestion of roads, the reconfiguration module proceeds to a pre-processing to decide if the current architecture of CAs needs to be reconfigured and whether components need to be added that are capable of triggering actuators to change traffic lights or displayed messages on roads being monitored. When the behavior of the monitored traffic undergoes frequent major changes possibly due to isolated events, activating the reconfiguration mechanism will not be an adequate response, because the system may suffer serious instability if subjected to continuous reconfigurations.

Sometimes, the decisions to be taken require human interaction to consider the broader context of overall road conditions. On the other hand, the fact of re-configuring after a decision considering the global context of the road can avoid annoying situations such as continuously changing displayed messages and traffic light patterns which may destabilize the behavior of drivers, leading to additional incidents.

5.3.1 Scenario 1: Architecture Reconfiguration Mnimizing the Consequence of Injury

Road Observations

In this scenario, an accident has just occurred in a neighborhood where a fleet of emergency vehicles is already deployed. The vehicle dispatcher receives real-time information and acknowledges the occurrence of this accident. Relying on suitable processing functions, the dispatcher selects the best vehicle to reach the incident location and communicates with the driver of that ambulance. Sometime later, the dispatcher becomes aware of congestion similar to that caused by an accident in the neighborhood of the dispatched emergency vehicle.

Hypothesis Regarding Scenario 1

The congestion observed has been caused by an unpredicted event near a work area on the route of the emergency vehicle. The same event has had an impact on another part of the road to be travelled by the same emergency vehicle. In both cases, the chosen emergency vehicle remains the best option compared to the other emergency vehicles currently deployed.

Operations of the Management System

When information about the observed congestion is sent to the management network, which has been operating in monitoring state, it reconfigures its processing components to process this new information. This may lead to an emergency vehicle reassignment under existing real-time dispatching models [23]. The output of this new processing takes the form of communication to the emergency fleet located in the vicinity of the incident, selecting the best vehicle and assigning it to the accident.

Scenario Analysis Review

If we consider the actual context related to knowledge from the hypothesis, we will see that it was not necessary to trigger the ambulance reassignment since the observed situations were related to temporary situations considered to be too short-lived to justify a reassignment of emergency vehicles. Moreover, such reassignment can actually cause delays and make the intervention of the emergency vehicles less effective.

5.3.2 Scenario 2: Avoidance of Congestion Causing Crashes

According to studies reported in [24], if the road traffic model and its time prediction become close to the *pre-crash buffer zone* introduced in [13], then it is imperative to proceed with a rerouting of vehicles to avoid or at least minimize collision risks.

Road Observations

We consider a junction leading to the entrance of a long road (highway access), with a dynamic display for motorists signaling a traffic diversion. The traffic there has the kind of heavy flow that may cause crashes [13], [24]. Several vehicles at the front of the traffic queue are too close to the junction to change their route, so the notice for diversion is too short.

Scenario Hypothesis
In this scenario, the hypothesis is that a transient condition strongly affects road traffic but only for a short duration (as when people leave a movie theatre). We assume that the traffic model will have the same behavior as traffic that could lead to a crash.

Operations of Network Management System
The mode of operation of the network management system based on reactive reconfigurations is as follows: information flows from physical components to the system and from the system to the physical components. The video cameras and sensors provide information that is then used to estimate and predict congestion classified as presenting a potential crash risk. The system transmits information enabling display of notice about traffic diversions to motorists in real time. This mode of operation repeats itself two or three times and the system eventually gets a message indicating occurrence of an incident.

Scenario Analysis
If the same scenario is repeated, it will be possible to notify the same motorists with control advisories to change route in order to avoid congestion. However, if these motorists follow this advice and still meet congestion, there is a potential that they will then disregard display signs and take an alternative road which may be truly congested and dangerous. Thus, this reconfiguration leads to negative side effect.

6. Simulation Experiments for Multi-criteria Decision Making Relating to Road Safety

This section presents a brief overview of our framework instanciation for ITS and its use for testing the previous scenarios.

6.1 Overview of our ITS Framework Instanciation for ITS

The architecture of our ITS framework can be perceived as a structure of three layers. The top layer represents the Supervisor Agent, which balances the load of data processing relating to the transportation network. The middle layer represents CAs, which actually process local information from the transportation network and produce control decisions to manage network performance, specifically road safety. The third layer consists of physical or embedded hardware agents, called Environment Agents, with two-way communication capacities between the road network and the ITS management framework. The communication capabilities are enabled by a telecommunication layer (not illustrated) which is beyond the scope of this paper. Environment Agents such as detection loops, video cameras and vehicles receive or sense real-time network information and forward it to the data processing layer. The same agents also relay traffic management decisions from the CAs to transportation network devices such as traffic lights, variable message signs and variable speed signs. Fig. 5 illustrates the interactions between these layers

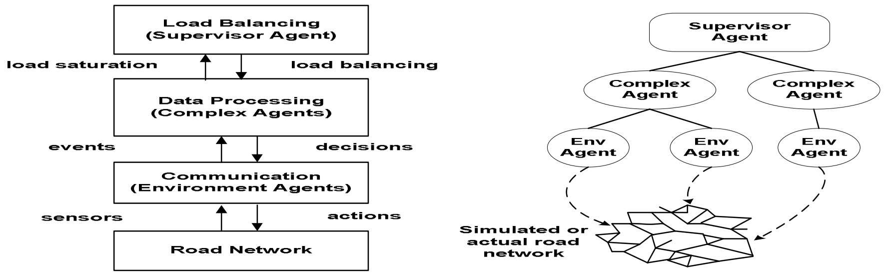

Fig. 5: Framework architecture agents communications

6.2 Dynamic Architecture of our Framework

Network management adaptation and dynamic reconfiguration is based on a decision process to avoid the negative side effects discussed in Section 5. This decision process is triggered before any reconfigurations are implemented. It takes place in two phases and involves two different decisions (see figure 6, time index ***t3*** and ***t4***). In the first phase, the CA decides how to process the transportation network information using run-time contextual information. In the second phase, it decides whether to act, if so, which network management action to take to adjust physical components. In other words, the first decision determines how the second decision will be produced.

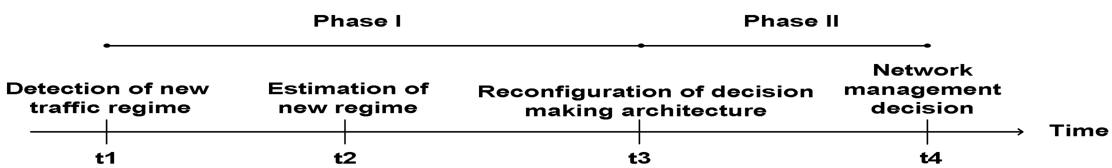

Fig. 6: Decision making sequence.

The processing architecture of CAs is designed as a pipeline of processing classes where each class is instantiated by at most one component. For example a framework for road safety management may consider the following pipeline of classes:

- Assessment of traffic regime (normal, medium, disturbed, etc);
- Filtering of sensor data;
- Data processing for vehicle routing.

The vehicle routing algorithm may be implemented by several alternate routing components in the form of heuristics and metaheuristics [25]. Each algorithm has different performance parameters according to the environment behavior phenomena.

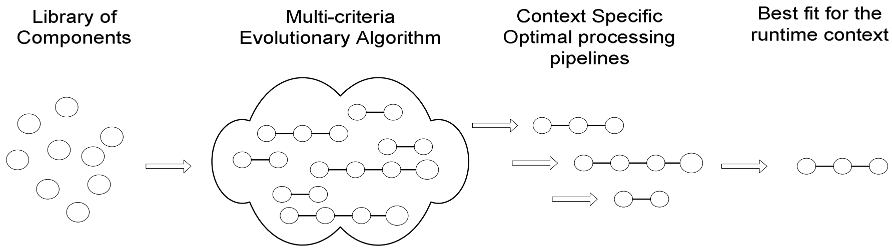

Fig. 7: Reconfiguration of the CA processing pipeline.

Each combination of components results in a unique processing architecture of the CA with distinct processing performance characteristics such as execution time and the accuracy of results. Furthermore, the performance of a component may vary greatly, depending on the actual behavior of its environment. Trying to define the best combination for a future network context at the time of design is unpractical because all possible contexts are rarely known then. An optimal combination decision is better achieved at run-time, when the most appropriate components can be accessed by the reconfiguration process from the library where they are stored. For examples of components which may be found in a library of components, we can see Table 1.

Monitoring of the traffic regime takes place on critical parts of the road network. It is performed primarily by the Environment Agents, which gather data from the network environment and relay this information to a CA for network management purposes.

If evaluation of incoming traffic information detects significant changes in the network, the CA triggers an online decision process (see Fig. 1). This process combines component performance information and data on real-time network conditions such as traffic flows to compute an optimized processing pipeline combination in the current context. Once a decision to reconfigure the pipeline is taken, the CA loads the components from the library and assembles the processing pipeline. It then proceeds to compute the best network management decision. Once calculated, the decision is forwarded to the Environment Agents for potential reconfigurations control on physical components.

6.3 Simulated Transportation Network Environment

For the experimental phase of our study, we selected a simulation tool called SUMO (Simulation of Urban MObility) [26], which we were able to connect with our ITS framework. Fig. 8 presents the overall system architecture we use in our experiments. SUMO is an open-source, highly portable, microscopic road traffic simulation package based on a car model. An important feature of the SUMO package is its client-server interface which allows an external system (such as our framework) to dynamically control a SUMO simulation run using a traffic control interface (TraCI) implemented over the TCP/IP protocol. SUMO simulations may also run in stand-alone mode.

In server mode, SUMO first initiates a simulation run by loading two files for network definition and traffic specification. It then awaits simulation commands from the client, who may dynamically control the simulation by modifying the behavior of physical network components such as vehicles, lanes and traffic lights. The state of the simulation is monitored by querying current status values for these network elements (position and speed of vehicles, occupancy rate of lanes, state of traffic lights, etc.). Control actions and queries about physical component states of the transportation network are handled by the Environment Agents in our ITS framework.

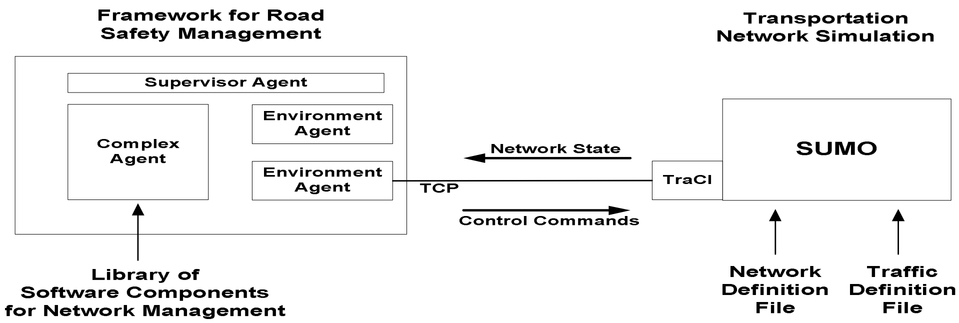

Fig.8: Framework connection to SUMO

6.3.1 Results of Experiment with Scenario 1

In Scenario 1 (presented in Section 5), we compare the performance of a reactive decision (Fig. 9) to that of a dynamic decision based on tolerance and run-time contextual information (Fig. 10). Fig. 9 shows that the reactive reassignment of the emergency vehicle is the best action to take at time index ***t4*** but the ambulance has an estimated arrival time ***t6'*** that will miss the response time target. In Fig. 8, the tolerance-based approach has more complete run-time information. The decision making process has some capacity to adapt to unexpected changes in the context. While taking longer to reach a decision, it may provide a more accurate time estimate for the arrival of emergency vehicle 1. This approach determines that the new arrival time at ***t6*** may satisfy the response time constraint without reassignment of another ambulance.

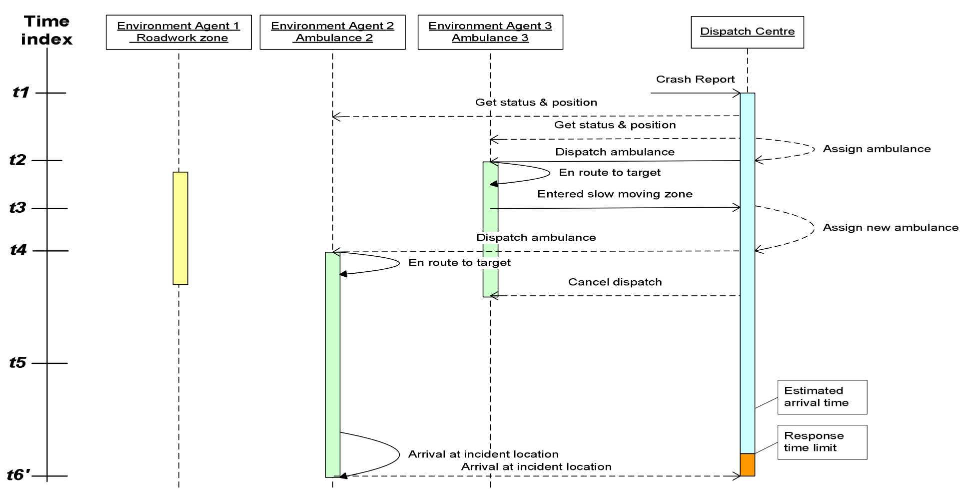

Figure 9: Reactive dispatching leading to an unsatisfied response time constraint.

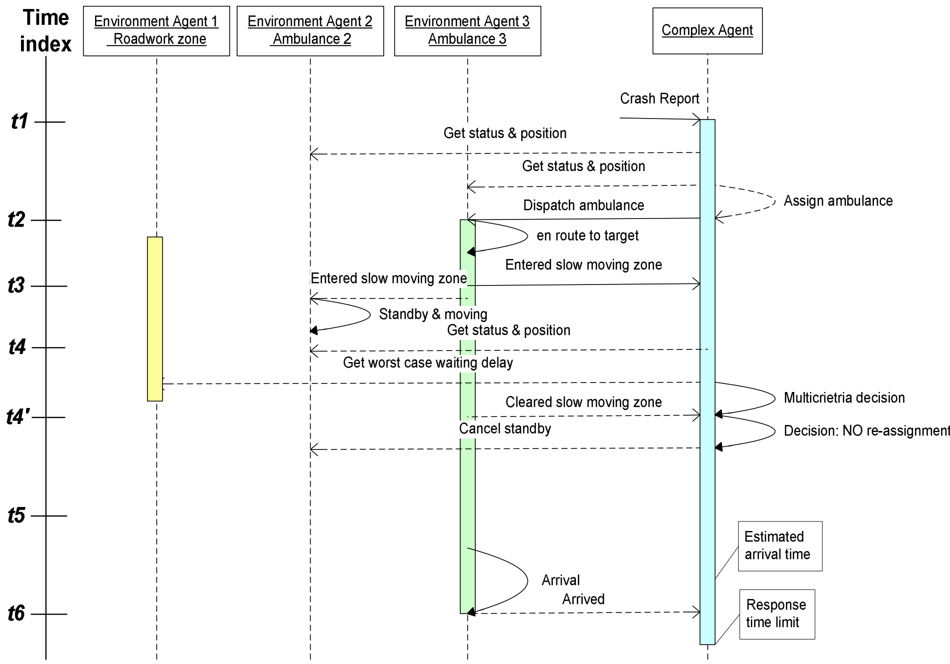

Fig. 10: Dispatching with waiting time reaction

6.3.2 Results of Experiments with Scenario 2

Fig. 11 shows a simple road network servicing a traffic flow which is mostly eastbound. This network is equipped with network management devices such as loop detectors and variable message signs (VMSs). A loop detector positioned on lane M2 is monitored by Environment Agent #1. Lanes L2 and L4 are equipped with VMS equipment which informs the motorists on these lanes about the state of the traffic downstream and the predicted duration of any detected congestion.

At the simulation time captured in Fig. 12, the traffic regime has recently changed from a ‘light free flow’ regime to a ‘congested flow’ regime due to the closing of a shopping mall near lane M2. This context change has been detected by the CA which estimates that this is a temporary change which should only last for a short time. Nevertheless, this transient regime could affect the safety of upstream vehicles planning to travel M2. No reactive decision is taken, as is always the case with our framework. Instead, the two-phase decision process constrained by a tolerance period is initiated. In this simulation run, the CA determines that rerouting is not necessary. It therefore decides to mitigate the short-term risk by simply informing motorists bound for segment M2 to beware of heavy traffic. This management action is relayed to all Environment Agents linked to VMSs positioned upstream of the affected routes (i.e., VMSs on L2 and L4).

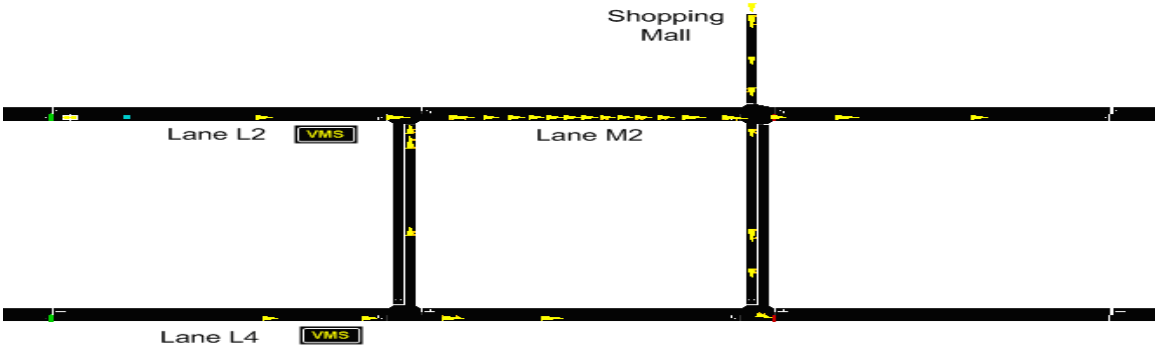

Fig. 11: Congestion detected on lane M2.

6.3.3 Dynamic Operation of the Reconfigurable Framework

Fig. 12 describes the sequence of events involved in the decision making process. At time index ***t1,*** the CA detects a change in the traffic regime and judges that the transportation network is moving towards a 'congested regime' which is unsafe. At ***t2,*** further assessment determines the value of additional regime parameters such as the intensity and probable duration of the perturbation. The CA triggers an MCDM process to select an optimized configuration for its own internal processing architecture. Multiple architectures are evaluated and the best configuration is selected at time index ***t3***. At time index ***t4,*** a network management decision is finally produced and relayed to Environment Agents #2 and #3 (not shown here) so that alternate route messages will be displayed on the VMSs in lanes L2 and L4.

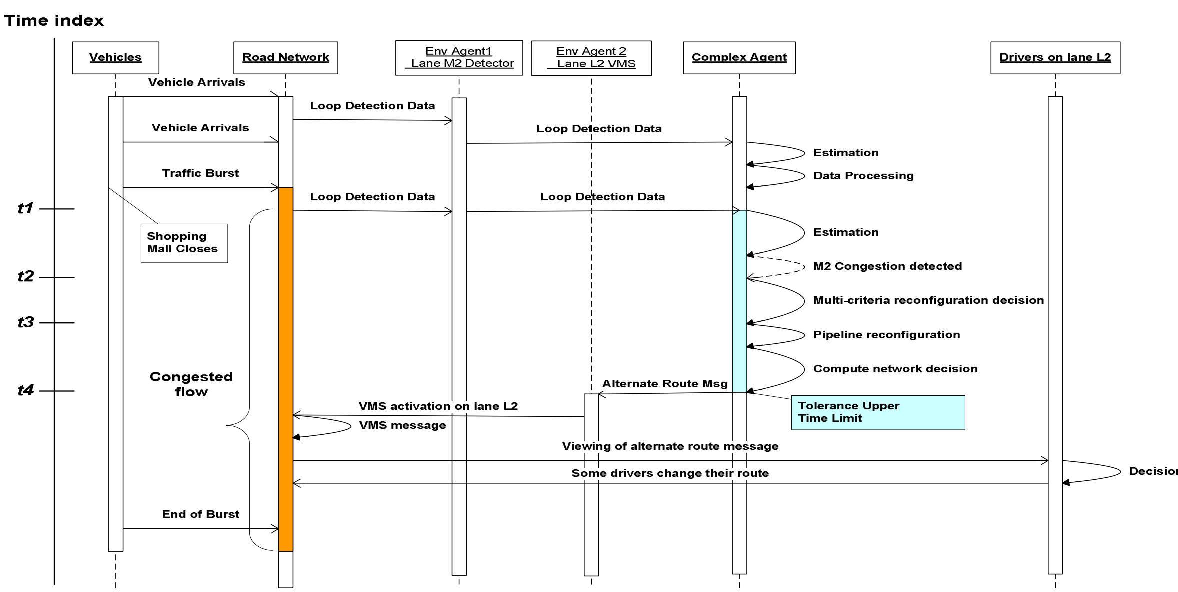

Fig. 12: Sequence of the network management decision process.

6.4 Benefits of a Run-time Decision with Tolerance

Reactive network management decisions are pre-calculated based on out-of-context data and are likely to be suboptimal. In this example, rerouting the vehicles on upstream lanes is not necessary and may in fact introduce instability in some other part of the network. Fig. 13 shows that under intensity burst of between 50 and 70 vehicles, a reactive rerouting approach actually exposes more vehicles to a congestion situation and therefore may increase the overall risk of a collision. Furthermore, motorists repeatedly exposed to poor traffic directives will eventually dismiss the VMS messages and may gradually adopt an unpredictable behavior.

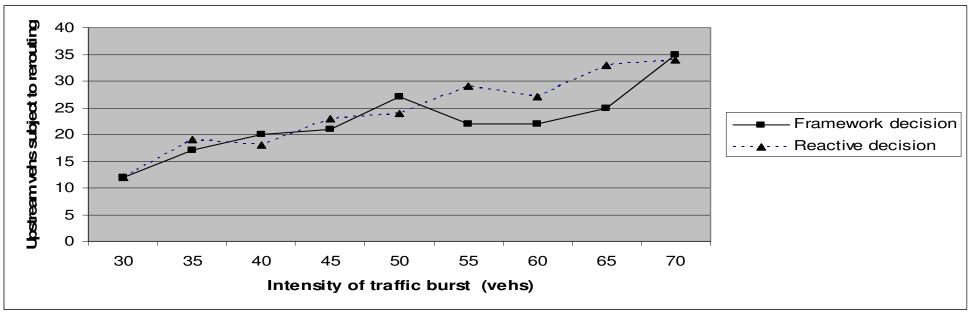

Fig. 13: Number of upstream vehicles heading towards congestion.

7. Conclusion and Future Directions

This paper deals with complex applications such as ITS that interact with unpredictable environments and are capable of adaptation and dynamic reconfiguration of their architecture based on large libraries of components. Such applications must meet high performance requirements even when operating in extreme situations.

Existing approaches to dynamic reconfiguration have given little consideration to the risk of destabilization due to frequent reconfigurations triggered by ongoing changes in the systems environment. Our approach incorporates evolutionary techniques to solve the NP-hard problem by combining suitable processing components and finding the best architecture configuration. The main contribution of this paper is the integration of an MCDM technique into the reconfiguration process using the Pareto Evolutionary Algorithm Adapting the Penalty (PEAP).

An important goal for ITS applications is to reduce the potential destabilization that may occur when frequent environment changes are followed by a flurry of reactive adaptations. Destabilization of the traffic regime may increase congestion or prolong the travel time of vehicles, including emergency vehicles, with a negative impact on road safety. Our simulation experiments on road safety scenarios have shown the potential benefits of MCDM both as a way of avoiding incidents related to congestion and as a means of minimizing any delay of emergency vehicles.

As a future direction, we intend to extend our approach to dynamic reconfiguration based on MCDM by including a module for supervised learning to enable autonomous systems and to reinforce expert decisions when expert knowledge is required.

ACKNOWLEDGEMENTS

This project is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Authors

Vincent Talbot is an information technology consultant in system engineering and distributed systems. He is currently studying as a graduate student at the Department of Computing and Engineering at Université du Québec en Outaouais

.

Ilham Benyahia is professor at the Department of Computing and Engineering at Université du Québec en Outaouais. She held a PhD in Computer Science in 1993 from University Pierre et Marie Curie (UPMC) in Paris. Before her position, she was researcher at IREQ (Institut de Recherche Hyro-Québec). She was involved in telecommunication, energy networks management and software engineering projects. Her current research interests include evolutionary computing and autonomous networks applied to Intelligent Transportation Systems and telecommunication technologies.

7. References

[1] Q. Wu,Jianhua Yang and Y. Li, "Semantic and Context-based Information Management in SCUDWare Middleware for Ubiquitous Computing", 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), 275-281, (2007).

[2] K. Byounghoon, K. Youngmann, D. Lee and S. Tak, "A Reconfigurable NoC Platform Incorporating Real-Time Task Management Technique for H/W-S/W Codesign of Network Protocols", pp 238-243, International Symposium on Ubiquitous Multimedia Computing (2008).

[3] J. Laitakari and D. Pakkala "Dynamic Context Monitoring Service for Adaptive and Context-Aware Applications", pp. 11-19, Eighth International Workshop on Applications and Services in Wireless Networks (aswn 2008).

[4] J. Hillman, and I. Warren, "An Open Framework for Dynamic Reconfiguration". ICSE 2004: 594-603.(2004).

[5] G. S. Blair, T. Coupaye, J-B. Stefani, "Component-based architecture: the Fractal initiative". Annales des Télécommunications 64(1-2): 1-4 (2009).

[6] E.Bruneton, T. Coupaye, M. Leclercq, V. Quéma, "The FRACTAL component model and its support in Java". Softw., Pract. Exper. 36(11-12): 1257-1284, (2006).

[7] H. Gomaa, "Designing Concurrent, Distributed, and Real-Time Applications with UML". ICSE (2001).

[8] J-Sim: http://sites.google.com/site/jsimofficial/

[9] D. Sykes, W. Heaven, J. Magee, J. Kramer, "From goals to components: a combined approach to self-management", Proceedings of the 2008 international workshop on Software engineering for adaptive and self-managing systems, (2008).

[10] M. Dellarocas, and K. H. Shrobe, "An Architecture for Construction Self-Evolving Software Systems". In Proc. of the Third Int. Soft. Arch. Workshop, (1998).

[11] L. Godard, "Distributed and Hierarchical Management Modeling for Reconfiguration and Decision Taking of Cognitive Radio Equipments". IETR - Institut d'Electronique et de Télécommunications de Rennes, (2008).

[12] N. Colson, A-A. Kountouris, A. Wautier, L. Husson, "Autonomous Decision Making Process for the Dynamic Reconfiguration of Cognitive Radios". ICCCN 2008: 497-502. (2008).

[13] T.F. Golob and W.W Recker, "A Method for Relating Type of Crash to Traffic Flow Characteristics on Urban Freeways", *Transportation Research Part A: Policy and Practice,* (2003).

[14] G. Ozerol and E. Karasakal, "Interactive outranking approaches for multicriteria decision-making problems with imprecise information", The Journal of the Operational Research Society, vol. 59, n° 9, pp. 1253-1268, (2008) .

[15] R. Slowiński , S. Greco and B. Matarazo**, "**Axiomatization of utility, outranking and decision rule preference models for multiple-criteria classification problems under partial inconsistency with the dominance principle". Control and Cybernetics 31 (4), 1005-1035 (2002).

[16] Ye 3 Chen, D. M. Kilgour, and W. Keith . Hipel, "Using a Benchmark in Case-Based Multiple- A: Criteria Ranking", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS— PART SYSTEMS AND HUMANS, Volume 39 , Issue 2, Pages 358-368 (2009).

[17] I. Benyahia & J-Y. Potvin, "Genetic Programming for Vehicle Dispatching." IEEE Transaction on Systems, Man and Cybernetics 28. 3, (1998):

[18] C. Sun; X. Qi; Ou Li, "Pareto-MEC for Multi-objective Optimization", Systems, Man and Cybernetics, 2003. IEEE International Conference on, (2003).

[19] S. Tak, P. Prathombutr, E. K. Park, "An efficient technique for a series of virtua topology reconfigurations in WDM optical networks". Computer Communications 30(6): 1301-1314,Section 4 (2007).

[20] I. Benyahia & V. Talbot, “Optimizing the Architecture of Adaptive Complex Applications Using Genetic Programming”. Proceedings of the Fourteenth International Conference on Distributed Multimedia Systems, Boston, Massachusetts, USA. 27-31, (2008).

[21] I. Benyahia, M.. Hilali. “An Adaptive Framework for Distributed Complex Applications Development”. IEEE TOOLS USA 2000 International Conference on Technology of Object-Oriented Languages and Systems, 339-349, (2000).

[22] http://www.tc.gc.ca/innovation/its/eng/architecture/documents/regional_its_architecture_guidance/pdf/regional_its_architecture_guidance.pdf

[23] A. Haghani, Q. Tian, H. Hu, “A Simulation Model for Real-Time Emergency Vehicle Dispatching and Routing”, Transportation Research Board (TRB) 2003, Annual Meeting CD-ROM, (2003).

[24] M. Pham, “Toward a Risk-Sensitive Active Traffic Management System”, YEAR 2010, The Young Arena of research, safety and security, (2010).

[25] L. Bianchi et al, “Hybrid Metaheuristics for Vehicle Routing Problem with Stochastic Demands”,Technical report no IDSIA-06-05 March 9, (2005).