Abstract

The distributed control applications of today are usually assumed to be run on a hardware consisting of heterogeneous processing elements. A standard like IEC 61499 attempts to provide an unanimous software infrastructure for these applications. An important research interest thereof is to find possible deployment of the components of the distributed application on these heterogeneous nodes. Since most of such applications are hard real-time ones the complexity of the problem is further increased by strict temporal restrictions. This paper addresses this issue as algorithms are presented for finding feasible and optimal solutions of such deployment problems. Moreover, the problems are also considered in the context of two different – while conforming to the standard – execution semantics realized in certain implementations of IEC 61499.

1. Introduction

IEC 61499 introduced itself in the Industrial Process Measurement and Control System (IPCMS) realm to help in maintaining unanimity in designing distributed applications along with additional functionality of reconfiguration. A major challenge is to deal with the possible variation of hardware and middleware platforms where an IEC 61499 compliant distributed control application can be deployed as one has to decide on where to deploy which component such that firstly the feasibility of the application is ensured and secondly certain objectives are satisfied optimally. The presented work focuses on this problem and its related facts.

The objective of the paper is to present an analytic approach for finding feasible deployment options for an IEC 61499 compliant distributed application.

The rest of the paper is organized as following. Section 2 discusses briefly related works by other researchers and highlights the principal features of the presented work. Then in the following section a concrete formulation of the problem is presented. Section 4 discusses assumptions and techniques regarding finding a pessimistic approximation of temporal feasibility analysis of the applications. Section 5 then discusses methodologies for finding feasible and optimal deployment of IEC 61499 compliant distributed control applications. In Section 6 a proof of applicability of the presented methodologies is presented. The concluding section summarizes the presentation and depicts a vision of the future works.

2. Related works

Prayati et al pioneered in presenting a methodology for development of IEC 61499 compliant distributed control applications along with an algorithm for deployment of developed artifacts on heterogeneous processing nodes [1]. The proposed design methodology was inspired by RT-UML [2]. During the early stages on one hand the functional requirements are elicited through Function Block Networks (FBNs) and on the other hand the characteristics of the devices are specified in terms of a generic device model contrived in [3]. These specifications are then fed into the deployment algorithm that decides on which device a particular FB of the network should be deployed maintaining compatibility with the industrial process connections, storage capacities and computational efficiencies of the devices and most importantly conforming to the real-time constraints. The deployment uses Branch and Bound (B&B) where each vertex represents allocation of a closely coupled set of tasks to a physical resource and the compatibility factors other than the real-time constraints are used in pruning the branches at each level. The objective function for the optimal allocation is feasibility ratio which refers to the probability of the set of tasks represented by a vertex to meet its deadline. It has been shown that a preprocessing of the search space through priority slicing of the FB tasks and the clustering of the most tightly coupled tasks shows positive effects through raising the feasibility ratio of the resulting schedules and success ratio of the B&B respectively.

In [4] Khalgui et al presented a heuristics based approach for deployment of IEC 61499 applications where exclusion, localization and resource constraints are considered. The deployment heuristic attributes static
priority level for each of the exclusion sets and then performs allocations in descending order of priority. After each deployment an integrated feasibility test of the concerned device and communication network is performed. These steps are performed repeatedly until feasibility is achieved. Unlike in [1] where optimization of an objective function is considered during deployment, here only a feasible deployment is looked for. Moreover, the algorithm requires formal specifications of the FB constructs and it is implicit how this specification can be achieved during a usual development lifecycle of a distributed control application.

In contrast to these works the presented work attempts to present a methodology for feasible as well as optimal deployment from the specifications prepared during the development phase.

3. Problem description

In general, deployment of IEC 61499 program constructs on heterogeneous processing nodes can be constrained by following three types of constraints [5],[6]:

1. Resource Constraints: a distributed real-time application can be restrained by any of the following:
   a. Memory Capacity
   b. Utilization Factor
   c. Network Usage

2. Allocation Constraints: The system architecture can impose the following constraints:
   a. Residence: restricting deployment of software components on a subset of available hardware.
   b. Co-residence: forcing that certain components are to be placed on the same processing node.
   c. Exclusion: this is the opposite of co-residence and inhibits co-existence of software components.

3. Time Constraints: this is the most important type of constraint and is usually stated in terms of deadlines in case of periodic tasks, or in terms of end-to-end response times for event-driven tasks.

In the context of the current work a hardware platform is taken where the processing nodes are modern network-enabled microcontrollers (i.e., TINI, SNAP etc.) connected to a dedicated Ethernet network. Therefore, as far as resource constraints are concerned both static and run-time memory constraints are to be dealt with. This means, beside the limitation of runtime memory usage, the size of codes and necessary library files to be deployed are also restrained by available space on the microcontroller memories. The Network, on the other side, does not impose any restriction on the number or size of usual messages exchanged between IEC 61499 components.

4. Feasibility of Temporal Constraints

The temporal constraints are the most critical ones and the most complex to deal with. The other constraints can be quite easily formulated as an integer programming problem but for a distributed system it is tough to formulate the temporal feasibility analysis problem to a familiar format. Therefore, a pessimistic worst case analysis is carried out. In the context of the current work two different execution semantics – sequential [7], [8] and NPMTR (Non-Preemptive Multi Threaded Resource) [9] – are considered along with some additional assumptions regarding them.

For execution semantics based upon sequential hypothesis [7], [8] the imposed assumptions are:

1. To spare space needed for saving the events along with the sampled data value it is assumed that every external event should have to be treated before the next instance of it arrives. Therefore, while analyzing the feasibility the deadline for finishing a sequence of real-time tasks triggered by an external event is equal to the minimum inter-arrival time of the triggering external event.

2. If there is more than one external event receiving SIFB in a resource, an event chain that is started by an external event cannot be preempted by an external event received by any other SIFB. This assumption has the implications that, firstly, the indefinitely long wait for certain tasks are avoided and secondly, from an analytic point of view this gives more tractability than other options.

3. For the sake of preemptively running the tasks triggered by an external event a separate queue has to be maintained where events generated as a result of a certain external triggers are saved.

4. While several resources are run on the same device that adopt this execution semantics it is assumed that concurrency is attained through alternately running the run-to-completion segments at each resource.

Furthermore, it needs to be mentioned that the event queues are assumed to be First-In-First-Out (FIFO). Considering the above mentioned assumptions feasibility analysis of an IEC 61499 compliant application is then divided into two parts. Firstly, the Worst Case Response Time (WCRT) for each of the tasks is to be calculated and Secondly, it needs to be ensured that the end-to-end deadline for a particular task chain is satisfied.

First, let us consider the case of single resource in single device. When there is only one SIFB for reception of external events in a resource, then the constituent tasks of the chain will have WCRT which is equal to the summation of Worst Case Execution Times (WCETs) of the preceding FB tasks plus the queuing delays plus the time allotted for the SIFB to perform reception. When there are multiple events receiving SIFBs in the
resource, the task sequences will compete for execution. Since there is no explicit prioritization other than precedence of occurrences, to be on the safe side the feasibility analysis calculates the WCRTs assuming that any task sequence can be blocked by all the others. Therefore, WCRTs are calculated starting from the one with smallest period implying strictest deadline.

The most generalized situation is that of having multiple resources in a device. Since in such a case the run-to-completion segments of each resource will be executed alternately, in addition to the blocking from other task sequences of the same resource, the execution of a task sequence may experience blockings due to execution of tasks of other resources. To consider the worst case of latency the WCRT is calculated adding to the aforementioned value the computational expenses of the most expensive sequences or set of most expensive sequences that can be executed in other resources.

After calculation of the WCRTs of task sequences the end-to-end response time can also be calculated adding cost of communication if necessary. Comparing the response time to the deadline one can be ensure whether the distributed application satisfies feasibility or not.

For NPMTR execution semantics [9] the following assumptions are made:
1. When a device is populated with more than one resource, SIFB and timer tasks are executed on the basis of timed multitasking.
2. The event processing threads within the resources form a chain of executions that goes through one or more critical regions. Since the execution of such chain of tasks could be considerably long the context switching among such threads will be done only after a critical region is executed.
3. Parallel branches are given predefined priorities defined upon the strictness of the demand on their completion.
4. Similar to the sequential execution semantics it is assumed that the chain of task executions instigated by an external or timer event should have to be finished before the following occurrence of it.

The methodology for analyzing feasibility of the NPMTR execution semantics is based upon the analysis technique of [10] and its later modification [11].

5. Finding Feasible Deployments

Finding the feasible deployment is a problem of searching in a huge search space that is built up of all possible combination of deployment of FBs. The resource and allocation constraints are an instrument for limiting the size of the search space. The temporal constraints are then used for the search in the remaining space.

In the current context two different approaches are taken dealing with the temporal constraint. The first one is inspired by the work presented in [5]. According to this solution approach the problem is decomposed into master and sub problem – the constraint satisfaction problem concerning the resource and allocation constraints forms the master problem. The feasibility constraint satisfaction problem is a sub problem. The master problem is solved initially using backtracking search along with use of arc consistency algorithm to infer from the explicit constraints the implicit ones [12]. Then the temporal constraint is checked on the set of feasible solutions resulting from the previous search. At an infeasible deployment from temporal perspective the possible reason is attempted to be searched through QuickXPlain algorithm [13]. Thus the conflicting set of tasks are found out and recorded as constraints. In this way the constraint processing system learns the temporally infeasible sets. This recursive enriching of knowledge about temporal infeasibility then helps in not exploring certain solutions which in the long run could have brought no feasible solution from temporal perspective.

In the other approach instead of using decomposition and learning branch and bound search algorithm is used. This branch and bound algorithm is applied only on the feasible sets that satisfy the resource and allocation constraints. Now the expansion of branches implies deployment of an FB on a particular resource on a physical device. The response times for the most asymmetric possible distribution and the most symmetric possible distributions that can be generated from each branch are calculated. The ratios of these response times respective to the deadlines are then summed to get a collective measure of upper and lower bounds at each expansion point.

The advantage of the latter method is that it is possible to take into consideration multiple objectives other than the one mentioned above. For example, when one wants to find the minimum number of certain processing nodes required or minimize utilization of certain set of devices it can be added as a part of the branching factor. A similar search of optimal set of solutions in the former case will require application of an optimization algorithm on the set of feasible solution.

6. Case study

To study the applicability of the proposed concepts and methodologies studies on two systems have been carried out. One of it is the well known FESTO MPS didactic model of a typical industrial application the other one is an example of a lift control system taken from [11]. The complete specifications of these systems as well as the detailed results of the analysis can be found in [14].

The most critical temporal constraints in the didactic plant were concerned with a rotary motor controller and a gantry crane. The response of the controller after
reception of sensor signal needed to be within 3 ms in the first case and within 10 ms in the second case.

For this system both the algorithms are tried for both sequential and NPMTR execution semantics. Moreover, the optimal deployments for both the semantics with respect to minimal usage of processing nodes and with respect to symmetric utilization of the physical resources and residual capacity is carried out.

Another application where the applicability of the algorithms is tested is a lift control application taken from [11]. The specialty of this case is that the number of required components can be simply multiplied inferring addition of the number of lifts in the system. It came out that as the number of components is doubled the explanation based algorithm converges quicker than the B&B algorithm. But it needs to be emphasized that the aim of the B&B algorithm is to deliver optimal deployments with respect to a definite objective function while the earlier algorithm gives only the feasible set of solutions. When the size of the set of feasible deployments is small enough it is quite easy to evaluate those but when the objective function is quite clear during the design phase and the number of feasible outcomes could possibly be large in size it is preferable to use the B&B algorithm. In case of the didactic plant the size of the set of feasible solutions was quite small and therefore it was quite easy to evaluate the feasible deployments resulting from the explanation-based algorithm.

7. Conclusion and Outlook

This work addresses the issue of finding a feasible as well as optimal deployment of IEC 61499 compliant distributed control applications. In [14] it is shown how the constraints that apply in such a problem can be specified during the different phases of software specification. The proposed work also addresses the context of differing execution semantics. There are still certain points which need to be resolved regarding the execution semantics. Therefore, certain assumptions are made to make the matters simple and intuitive.

During the temporal feasibility analysis a worst case scenario is assumed which might finally force the search for feasible deployments to come out empty handed. Considering this facet of the problem as well as keeping in mind that reconfigurability is an important feature of such application explanation based learning of the problem is used. The strength of the algorithm in combating the first problem is tested in the current work. To solve the latter one is one of the future works.

References