A Fuzzy Expert System Architecture Implementing Onboard Planning and Scheduling for Autonomous Small Satellite

Yang Jiajun Xu Guodong Chao Xibin Ma Xingrui
Research Institute of Satellite Engineering and Technology,
Harbin Institute of Technology, Harbin, China 150001
xbcao@hope.hit.edu.cn

Abstract. A crucial element in achieving small satellite fully autonomy is onboard automated planning and scheduling. Though lots of successful applications about it in spacecraft operations have recently been reported, and all of them are important for the development of this field, the inherent fuzziness and uncertainty of planning and scheduling was ignored made the achieving of onboard automated planning and scheduling system become idealization and no reality.

In response to above-mentioned requirement, this paper presents an architecture which is developed using rules-based Fuzzy Expert System(FES). Fuzzy expert systems not only maintains the value of based rules and the merit of using fuzzy logic control to describe uncertainty systems, and utilizes the predominance of using expert systems to denote and control knowledge. In order to adapt the requirement of onboard operation, the resource restrain is considered in the architecture, such as processing speed of CPU, the capacity of storage and the real-time requirement.

As the application of this architecture, this paper also describes an onboard automated planning and scheduling system in the SMMS, which is a small satellite of Research Institute of Satellite Engineering and Technology of Harbin Institute of Technology and will be launched in 2000.

Introduction

Not only the improvement of technology makes the inherent characteristic of small satellite, that is “faster, better, and cheaper”, to be developed forward endless, the idea of establishing “virtual presence” in space in the next century also requires the small satellite developing towards “smarter”. So, several new technologies need to be demonstrated, and one of the most crucial is on board autonomy.

On Board Autonomy

For a long time, the satellite operations which include a large number of functions, such as planning mission, sequence the execution commands, tracking the spacecraft’s internal hardware state, ensuring correct functioning, recovering in cases of failure, and subsequently working around faulty subsystems, or reconfiguring system, were carried out through humans intervention on the ground. This traditional approach is necessary and useful to the traditional spacecraft, but will not be viable anymore in the future due to (1) up-link and down-link communication time delay which makes driving a deep space mission impossible; (2) a desire to limit the operations team and cost; (3) ensuring high reliability through handling failures real-time.

On board autonomy integrates three separate technologies: an on board planner/scheduler, a robust multi-threaded executive including internal commands and telemetering, telecontrol commands, and a fault diagnosis and recovery system.

In the new model of operations, the scientists or operators will communicate high level science goals directly to the spacecraft. The spacecraft will then perform its own science planning and scheduling, translate those schedules into commands sequence, verify that they will not damage the spacecraft, and ultimately execute them without routine human intervention. In the case of error recovery, the spacecraft will have to understand the impact of the error on its previously planned sequence and then reschedule in light of the new information and potentially degraded capabilities.

The goal of the planner/scheduler is to generate a set of synchronized low level commands that

once executed will achieve missions goals.

Planning and Scheduling Questions

Planning and scheduling is not a new subject. Many planning and scheduling methods have been proposed and analyzed since at least the 1950s. Although related and often tightly coupled, Strictly Speaking, planning and scheduling are distinctly different activities. Planning is the construction of the project/process model and definition of constraints/objectives. Scheduling refers to the assignment of resources to activities (or activities to resources) at specific points in, or duration of, time. The definition of the problem is thus primarily a planning issue, whereas the execution of the plan is a scheduling issue. Yet planning and scheduling are coupled; the performance of the scheduling algorithm depends on the problem formulation, and the problem formulation may benefit from information obtained during scheduling.

Because of uncertainties, and being based on incomplete data, planning and scheduling problems are dynamic. No schedule is static until the project is completed, and most plans change almost as soon as they are announced. Depending on the duration of the project, the same may also be true for the objectives. The dynamics may be due to poor estimates, incomplete data, or unanticipated disturbances. As a result, finding an optimal schedule is often confounded not only by meeting existing constraints but also adapting to additional constraints and changes to the problem structure.

Practically speaking, finding an optimal schedule is often less important than coping with uncertainties during planning and unpredictable disturbances during schedule execution. In some cases, plans are based upon well known processes in which resource behaviors and task requirements are all well known and can be accurately predicted. In many other cases, however, predictions are less accurate due to lack of data or predictive models. In these cases the schedule may be subject to major changes as the plan upon which it is based changes.

Although methods exist for finding optimal solutions to some specific scheduling problem formulations, many methods do not work when the structure of the constraints or objectives change.

Though lots of successful applications about planning and scheduling in spacecraft operations have recently been reported, and all of them are important for the development of this field, the inherent fuzziness and uncertainty of planning and scheduling was ignored made the achieving of onboard automated planning and scheduling system become idealization and no reality.

Rules-based Fuzzy Expert Systems(FES) not only maintain the value of based rules and the merit of using fuzzy logic control to describe uncertainty systems, and utilize the predominance of using expert systems to denote and control knowledge.

Fuzzy Expert Systems

Expert systems based conventional logic are not efficient in handling inaccurate and inexact information. Fuzzy logic based expert system is a powerful tool providing failure analysis of a complex and nonlinear dynamic system, like a final control element.

To date, fuzzy expert systems are the most common use of fuzzy logic. They are used in several wide-ranging fields, including:

- Linear and Nonlinear Control
- Pattern Recognition
- Financial Systems
- Operation Research
- Data Analysis

As previously mentioned, This paper adopts a rules-based FES architecture used to on-board planning and scheduling. The architecture also considered all kinds of limits of on board operations, such as processing capability of CPU, memory size, and the desire of real-time. In response to above consideration, this paper presents an architecture which is developed

using rules-based FES. In order to adapt the requirement of on board operation, the resource restrain is also considered in the architecture, such as processing speed of CPU, the capacity of storage and the real-time requirement. As the application of this architecture. This paper also describes an onboard automated planning and scheduling system in the SMMS, which is a small satellite of Research Institute of Satellite Engineering and Technology at Harbin Institute of Technology, and will be launched in 2000.

Domain and Requirements

As we known, The characteristics of small satellites make on board automated planning and scheduling function different form other project applications. Small Satellite domain places a number of requirements on the software architecture that differentiates it from domains considered by other researchers or other projects. There are some major properties of the domain that drove the architecture design as following:

1. Human couldn't intervene a on board small satellite real-time, but the high reliability must be ensured.

Though small satellite is cheaper than large spacecraft, it is also expensive and often unique, so a high reliability must be requirement by user. However, the harsh environment of space or the inability to test in all flight conditions and still cause unexpected hardware or software failures, so that small satellite must have autonomous operations function that can rapidly react to contingencies by retrying failed actions, reconfiguring subsystems or ensuring the small satellite to prevent further, potentially irretrievable, damage.

2. The resources of small satellite is severely limited, and must be used optimized in order to achieve the missions.

Small satellite uses various resources, including obvious ones like fuel and electrical power, and less obvious ones like the number of times a battery can be reliably discharged and recharged. Some of these resources are renewable but most of them are not. Hence, autonomous operations requires significant emphasis on the careful utilization of non-renewable resources and on planning for the replacement of renewable resources before they run dangerously low.

3. Small satellite operation is a complicated concurrent activity.

Small satellite has a number of different subsystems, all of which operate concurrently. Hence, reasoning about the small satellite needs to reflect its concurrent nature. In particular, the planning and scheduling needs to be able to schedule concurrent activities in different parts of the small satellite, including constraints between concurrent threads active to handle concurrent commands to different parts of the small satellite.

Module and Method

Formally, Resource Constrained Project Scheduling(RCPS) is characterized by the following:

Given: A set of tasks T, a set of resources R, a capacity function $C:\rightarrow \mathrm{N}$, a duration function $D:T\rightarrow \mathrm{N}$, a utilization function $U:T\times R\rightarrow \mathrm{N}$, a partial order P on T, and a deadline d.

Find: An assignment of start times $S:T\rightarrow \mathrm{N}$, satisfying the following:

1. Precedence constraints: if t_1 precedes t_2 in the partial order P, then $S(t_1)+D(t_1)\le S(t_2)$.
2. Resource constraints: For any time x, let $running(x)=\{t \mid S(t)\le x<S(t)+D(t)\}$, then for all time x, and all $r\in R$, $\sum_{t\in running9x)} U(t,r)\le C(r)$.
3. Deadline: For all tasks $t:S(t)\ge 0$ and $S(t)+D(t)<d$.

Several concepts

Activities

As a data structure, activities performs specific detail functions which spacecraft must execute. An activity represents an action or step in the database. It maybe use one or more constraints. Moreover, An activity maybe include one or more subactivities.

Subactivities are activities that can be scheduled any time within the parent activity subject to resource constraints within the subactivity. Subactivities are similar to the constraint-defined activities without the exact temporal relationship between the parent and subactivities.

An activity may have multiple execution modes. Any activity may be executed in more than one manner depending upon which constrains are used to complete it. Interruption modes may depend on the resources that are applied to the activity.

Constrains

Different from typical Resource Constrained Project Scheduling(RCPS) questions, onboard autonomous planning and scheduling includes other types of constraints: temporal constraints, precedence constraints, availability constraints, besides resource constraints. Constraints turn a relatively smooth solution space with many optimal solutions to a very non-uniform space with few feasible solutions.

A valid plan must satisfy many constraints, including ordering constraints(e.g., the catalyst-bed heaters must warm up for ninety minutes before using the reaction control thrusters), synchronization constraints(e.g., the antenna must be pointed at the Earth during up-link), safety constraints(e.g., do not point the radiators within twenty degrees of the sun), and resource constrains(e.g., the CCD camera requires 50 watts of power). These are all expressed as temporal constraints.

As the most important constraints of small satellite on board planning and scheduling system, resource constraints have four types: automic, concurrency, depletable, and non-depletable. Automic resources are physical devices that can only be used(reserved) by one activity at a time, star tracker, reaction wheel, CPU are automic resources. Concurrency resources are similar to atomic except they must be made available to the activity before they are reserved, a telecommunications down-link pass is a kind of concurrency resources. Non-depletable resources are resources that can used more than one activity can use a different quantity of the resource, solar array power is the typical non-depletable resources. Depletable resources are similar to non-depletable except that their capacity is diminished after use, in some cases their capacity can be replenished(battery energy, memory capacity) and in other cases it cannot(fuel).

In many solution methods which have been proposed and implemented, heuristic methods were devised to find good solutions, or to find simply feasible solutions for the really difficult problems mostly. Most research now consists of designing better heuristics for specific instances of scheduling problems. However, heuristic solutions are typically limited to a specific set of constraints or problem formulation, and devising new heuristics is difficult at best.

Because of the severely limited on-board processing capabilities, heuristic methods which were used frequently in other projects are no more suitable for on-board spacecraft planning and scheduling. In order to suffice the fuzzyness or uncertaincy of spacecraft planning and scheduling under condition of severely limited on-board resources, we adopted Onboard Planning and Scheduling Fuzzy Expert System Architecture(OPSFESA) which fuzzy logic is useful in handling uncertain systems, and expert system theory is advantage of handling rule based knowledge system.

Constrains are denoted in the form of fuzzy rules as “if… then…”. Figure 1 is the example of CCD Camera Working, in which, attitude control precision, attitude control stability,

universal time, latitude position and longitude position status is required, and power resource must be sufficed, all these premises have one fuzzy degree, under the premise and fuzzy degree, the activities include CCD camera preparation, data recorder opening, CCD camera working and compressing image data. Figure 2 is the example of CCD camera preparation activity, the example of resources and status is in Figure 3 and 4

CONSTRAINS_IF:
Status_AttitudePrecision<0.3
Status_AttitudeStability<0.001
Status_StartUniversalTime=[6000,1200]
Status_Latitude=[-35,14]
Status_longitude=[25,70]
Resource_Power>120
Premise_FuzzyDegree=0.85
ACTIVITIES_THEN:
Activity_CCDCamera_prepare
Activity_Recorder_Open
Activity_CCDCamera_Work
Activity_DataProcess_compress
RULE_END

Figure 1 An Example of Constrains which is used to photograph

ACTIVITY Activity_CCDCamera_prepare
Related_Device=[CCDCamera, heater]
Constraint= {
Status_CCDCameraTemperature>20,
Resource_Power>10,
Duration=90 }
SubActivities= {NONE}
Commands=14
//the procedure which achieve the activity
Running_Flag
//one of Idle, Running, Finished, Non-valid
ACTIVITY_END

Figure 2 An Activity Example

RESOURCE Resource_Power
NO=155
Value=120
Reliability=0.97
RESOURCE_END

Figure 3 An Example of resource

STATUS Status_Precision
NO=22
Value=0.57
Change_Order=[]
// if Statue value changes by order
Reliability=0.97
STATUS_END

Figure 4 An Example of status

Architecture Describe

OPSFESA is achieved through the cooperation of the following components(See as Fig. 5):

Figure 5 On board Planning and Scheduling Fuzzy Expert system architecture

Goals Profile contains the list of all high level goals that have to be achieved over the entire duration of the mission, which is put into by communication from ground or initiate before being lunched;

Engine management center which attempers and coordinates every parts, has functions, activating fuzzy inference mode, retrieving mission goals and commands sequence, driving subsystems and assemblies, and so on.

Fuzzy rules Base a storage of all constrains and activities, also include maintenance of them.

Fuzzy inference mode is the kernel of OPSFESA, in there high level goals is translated acceptable low level commands sequence which be used to control subsystems and assemblies directly.

Commands sequence The detailed sequence of low level commands to achieve high level goals, however, is not pre-stored but is generated on board by the planner.

Fuzzy planning and scheduling mode is composed of fuzzy rules base and fuzzy inference mode(See as Fig. 6)

Figure 6 Fuzzy planning and scheduling mode

Implementation

Continuous operation is achieved by repetition of the following cycle:

1. Retrieve high level mission goals from the goals profile send to fuzzy inference mode;
2. Ask the engine to activate fuzzy planning and scheduling mode, in there, mission goals are transmitted low level commands

sequence;

3. Send the new generated schedule which is a set of commands sequence to the database, in there, commands are retrieved to control subsystems or assemblies.

If a new schedule is generated, the engine will continue executing its current schedule and start executing the new schedule when the clock reaches the beginning of the new scheduling horizon, and making sure that the commands succeed and either retries failed commands or generates an alternate low level command sequence. Hard command execution failures may require the modification of the schedule in which case the executive will coordinate the actions needed to keep the spacecraft in a "safe state" and request the generation o a new schedule from the planning and scheduling architecture.

4. Repeat the cycle from step 1 when one of the following conditions apply:
   (a) A new goals profile needs for generating a new schedule;
   (b) The engine has requested a new schedule as a result of a hard failure.

OPSFESA is the only component that is activated as a "batch process" and dies after a new schedule has been generated. This ensures the high reliability and rapidity required by the domain.

The results of OPSFESA is generation of spacecraft low level commands sequence from high level goals specifications. So, it will encoding of complex spacecraft operability constraints, flight rules, spacecraft hardware models, science experiment goals and operations procedures to allow for automated generation of low level commands sequence.

Because on-board resources are severely limited, the architecture needs to generate courses of action that achieve high quality execution and cover extended periods of time. OPSFESA always has to trade off the level of goal satisfaction with respect to the long term "mission success" and within the resource limitations.

We choose to plan at infrequent intervals because of the limited on-board processing capabilities of the spacecraft. The planner must share the CPU with other critical computation tasks such as the execution engine, the real-time control loops and the fault detection, isolation and recovery system. While the planner generates a plan, the spacecraft must continue to operate. More importantly, the plan often contains critical tasks whose execution cannot be interrupted in order to install newly generated plans.

There will be a copy of the on-board planner built into the ground system. This copy will be used to generate experience and rules of thumb as to what sets of goals are easily achievable and what sets are difficult to achieve for the on-board system based on these rules of thumb. The operators will define the goals for each mission phase and since the Remote Agent is closing the loop around these goals, the best prediction of spacecraft behavior is that the goals will be achieved on schedule.

OPSFESA must be able to respond to unexpected events during plan execution without having to plan the response. Although it is sometimes necessary to re-plan, this should not be the only option. Many situations require responses that cannot be made quickly enough if the has to plan them.

The executive must be able to react to events in such a way that the rest of the plan is stall valid. To support this, the plan must be flexible enough to tolerate both the unexpected events and the executive's responses without breaking. By choosing an appropriate level of abstraction for the activities, and second, by generating plans in which the activities have flexible start and end time.

Changing the start or end time of an activity

may also affect other activities in the plan. Although flexibility in the activity start and end times typically occurs because the times are under-constrained, flexibility can also occur because the duration of an activity is not determined until execution time. The plan must be able to represent this kind of uncertainty. There two ways of doing this. One is to use the existing capability for flexible and end times; a second approach is to fix the end time of the activity to the latest end time, and change the semantics of the activity.

Conclusion

As an example, OPSFESA is used to the on board mission planning and scheduling of Stereo Mapping Micro Satellite(SMMS) which is researched by Research Institute of Satellite Engineering and Technology at Harbin Institute of Technology, and will be lunched in 2000.

The most important distinction between the OPSFESA and other similar systems is that our architecture pays more attention to the characteristic of small satellite and its constrains and limitations, that makes the realized system to be smarter and terser.

Besides planning and scheduling technology, the key technologies of spacecraft autonomy also include a robust multi-threaded executive including internal commands and telemetering, telecontrol commands, and a model-based fault diagnosis and recovery system. Many researchers have important achievement in spacecraft autonomy, such as NMRA[3-7]) and DCAPS[1,2]. All the key technologies of spacecraft autonomy are study subjects of Research Institute of Satellite Engineering and Technology of Harbin Institute of Technology, and will be applied into the small satellites series they are developing.

Acknowledgements

The authors would like to thank the staff of Research Institute of Satellite Engineering and Technology of Harbin Institute of Technology.

References

1. Sam Siewert, Linden H. Mcclure, A system architecture to advance small satellite mission operations autonomy, 9th Annual AIAA/Utah State Univ. Conference on small satellites, 1995.
2. Sam Siewert, Elaine Hansen, A distributed operations automation tested to evaluate system support for autonomy and operator interaction protocols, 4th .International Symposium on Space Mission Operations and Ground Data Systems, ESA, Forum der Technik, Munich, Germany, September,1996.
3. Muscettola, B. Smith, C. Fry, S. Chien, K. Rajan, G. Rabideau, and D. Yan. On-Board Planning for New Millennium Deep Space One Autonomy, Proceedings of the IEEE Aerospace Conference, Snowmass CO, 1997.
4. Smith, K. Rajan, N. Muscettola, Knowledge Acquisition for the Onboard Planner of an Autonomous Spacecraft, Lecture Notes in Artificial Intelligence (Proceedings EKAW-97), Springer-Verlag.
5. Pell, Gat, Keesing, Muscettola, and Smith, Plan Execution for Autonomous Spacecraft, Working Notes of the AAAI Fall Symposium on Plan Execution, Cambridge, MA, 1996. Copyright 1996, AAAI.
6. Barney Pell, Erann Gat, Ron Keesing, Nicola Muscettola, and Ben Smith, Robust Periodic Planning and Execution for Autonomous Spacecraft, Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), Nagoya, Japan, August 1997.
7. Pell, D. Bernard, S. Chien, E. Gat, N Muscettola, P. Nayak, M. Wagner, B. Williams, An Autonomous Spacecraft Agent Prototype, Proceedings of the First

Annual Workshop on Intelligent Agents, Marina Del Rey, CA, 1997.

8. Jouni P., Harri C., Fuzzy Logic Expert System and Failure Analysis of Dynamic System, http://www.ics.org/memOnly/Conference/papers/
9. Barry, John M., and Sary, Charisse, Expert system for on-board satellite scheduling and control, Fourth Conference on Artificial Intelligence for Space Applications 1989, p 193-203.
10. Barry, John M. and Gathmann, Thomas P., The application of fuzzy logic control to the IntelliSTAR (TM) architecture, Annual Rocky Mountain Guidance and Control Conference, 16th, 1993, p. 423-430

Author Biographies

Yang Jiajun is currently pursuing a Ph.D. of spacecraft design at Harbin Institute of Technology in the Astronautics engineering and mechanics Department. His research interest is on board spacecraft autonomy.

Xu Guodong, member of SMMS project from 1996, received B.E of Wireless Engineering at Harbin Institute of Technology.

Cao Xibin Ph.D., The superintendent of Research Institute of Satellite Engineering and Technology at Harbin Institute of Technology

Ma Xingrui Ph.D., The manager of Chinese Institude of Space Technology, is the tutor of Yang Jiajun.