Proceedings of the Korean Nuclear Society Spring Meeting
Cheju, Korea, May 2001

Recent Enhancements of the INSIGHT Integrated In-core Fuel Management Tool

Akio Yamamoto

Nuclear Fuel Industries, Ltd.
950 Ohaza Noda Kumatori-cho Sennan-gun
Osaka Japan 590-0481

Abstract

Recent enhancements of the INSIGHT system are described in this paper. The INSIGHT system is an integrated in-core fuel management tool for pressurized water reactors (PWRs) runs on UNIX workstations. The INSIGHT system provides various capabilities which contribute to reduce fuel cycle cost and workload of in-core fuel management tasks, i.e. core follow calculations, interactive loading pattern design, automated multicycle analysis and interface between detailed core calculation codes. To minimize engineers' workload, most of input data for analysis modules are automatically generated by the INSIGHT system through specification of calculation conditions in the graphic user interface. Recent enhancements of the INSIGHT system are mainly focused to improve efficiency of loading pattern optimization and flexibility of multicycle analyses. To increase optimization efficiency, a parallel calculation capability, various optimization theories, extension of heuristic rules, screening by neural networks and so on were incorporated in the loading pattern optimization module. The multicycle analyses module was rewritten to increase flexibility such as cycle dependent specification of loading pattern search methods and so on. The INSIGHT system is currently used by Japanese utilities not only for regular in-core fuel management tasks but also for strategic fuel management studies to reduce fuel cycle cost.

1. Introduction

In-core fuel management is a complicated task which includes wide range of calculation activities; cross section preparation, core follow calculations, long term scoping analysis for successive several cycles, detailed loading pattern design for next cycle, reload safety evaluations, data base maintenance, reporting and so on. Due to beginning of de-regulation in the Japanese electricity market, the utilities are becoming more aggressive for the in-core fuel management calculation since it has significant impact on the fuel cycle cost. Therefore, a convenient tool for in-core fuel management calculations is highly desired.

The INSIGHT system is an integrated in-core fuel management tool for pressurized water reactor developed by Nuclear Fuel Industries, Ltd.[(1)]. The INSIGHT system is graphic user interface (GUI) driven interactive design tool operated on Unix workstations. Development of the INSIGHT system was started in 1994, and the initial version was released in 1995. The INSIGHT system includes various calculation capabilities, i.e. core follow calculations, interactive loading pattern design, loading pattern optimization, automated multicycle analyses, interface to other fuel management codes and integrated database, and can support most of complicated in-core fuel management tasks through GUI operation.

Up to present, various modifications were made in the INSIGHT system incorporating users' feedback and advances in computation technology. Especially the loading pattern optimization module, called GALLOP, and the automated multicycle analysis module, called MCA, are completely rewritten during this period. In the following sections, recent enhancements of the INSIGHT system will be described.

2. Overview of the INSIGHT system

The INSIGHT system is an integrated in-core fuel management calculation tool for PWR. Overview of the

INSIGHT system is shown in Fig.1 and typical GUI are shown in Fig.2.

The INSIGHT system has a modular structure; it consists of several independent programs and they are controlled by the task manager, which is responsible for user interaction. Since INSIGHT is a modular system, updating of its component program is easy. For example, the loading pattern optimization module and the automated multicycle analysis module were recently rewritten but coupling with INSIGHT required trivial work.

To enhance merits of the modular system, INSIGHT is designed by the object-oriented approach and mainly written in the C++ language. The INSIGHT system is composed of approximately 400,000 lines of source code.

The major objectives of the INSIGHT system are:

· To gain an insight for strategic fuel managements to reduce fuel cycle cost through loading pattern optimization capability.
· To reduce workload of in-core fuel engineers for trivial but inevitable jobs in calculations, e.g. data transfer among calculation codes and collection of various data for reporting.
· To evaluate many fuel management scenarios to effectively reduce the fuel cycle cost.
· Improve accuracy of fuel management by performing multi-dimensional core calculations instead of the traditional one-point reactor model.

To achieve these aims, the INSIGHT system was developed based on the following design concepts:

· Automation
· Optimization
· Database
· GUI

In general, input data preparation for in-core fuel management calculations are complicated tasks since it requires various data, e.g. loading patterns for previous cycles, operating conditions, fresh fuel/BP storage, date of fuel transportation, plant specific data, limitations of safety parameters and so on. Therefore, input data for these calculation codes generally reaches hundreds of lines and requires extensive error checking. The INSIGHT system provides automated input data generation capabilities using the database and minimum user input from GUI.

The optimization capability of loading patterns is a key feature of the INISIGHT system since performances of a core, i.e. safety and economics, depend on its fuel and burnable absorber location in a core. Therefore, the INSIGHT system has extensive optimization capability that will be described in the next section. For loading pattern optimizations, user can specify most of calculation conditions from the GUI; fuel assemblies/BP, pairing, fixed fuel/BP position, optimization method, heuristic rules, constraints and objectives for optimization and so on. Users can also monitor execution status and predicted wall clock time required for a whole optimization run.

The INSIGHT system has integrated database to assist automated input data generation. There are six databases in the INSIGHT system:

· Plant operating plan database
· Safety parameter limit database
· Calculation condition database
· Core geometry database
· Fuel specification database
· Loading pattern library

These databases are composed of text data look like ".ini" file of the MS-Windows except the loading pattern library. Therefore maintenance of these databases is easy.

Core calculation code is treated as a module in the INSIGHT system. Currently, the SHARP code, which is NFI's in-house core simulator, and the SIMULATE3, which is common LWR core simulator developed by Studsvik, can be used. These core simulators can be selected via an environmental variable of UNIX.

To reduce engineers' workload, the INSIGHT system has extensive printout for various loading pattern and core characteristics. Since quality of these printouts is important to reduce workload, continuing improvements are going on based on users' feedback. Since all calculations performed by the INSIGHT system is managed in it, engineers can retrieve any parameters of any pattern at any time.

3. Loading pattern optimization module, GALLOP

In the in-core fuel management activities, the loading pattern design is one of the most important tasks since core performance, i.e. safety and economics, almost depends on it. However, the loading pattern design is also one of the most difficult task that required state-of-art engineers' experience.

The GALLOP code[(1),(2)] is the LP optimization module of the INSIGHT system and recently rewritten to enhance its capability. Key features of the GALLOP code are described below.

Various optimization methods can be used in the calculations: genetic algorithms [(3),(4)], distributed genetic algorithms [(5)], simulated annealing[(6)], temperature parallel simulated annealing[(7)], direct search[(4)], enumerated binary exchange[(4)], rotational search[(4)], knowledge-based search and reactivity rank matching search. By utilizing these optimization methods, user can control computation time and quality of final loading pattern that have trade-off relationship. For example, the temperature parallel simulated annealing is suitable for detailed loading pattern design of next cycle to find out more economical solution, though it requires approximately 10,000 candidate evaluations. On the contrary to this, the knowledge-based search or the reactivity rank matching search can be used for preliminary analyses of successive several cycles since only a few to dozens candidate evaluations are necessary to obtain the final results using these search methods. Such flexibility is important in actual in-core fuel management, since requirement for accuracy and limitation on available time is different in each situation.

Extensive heuristics including more than 30 heuristic rules can be taken into account to improve optimization efficiency. The heuristic rules are greatly expanded from previous version (9 heuristics) based on the users' experience.

Rich objective function including about 50 core characteristics can be used. The objective function is also greatly expanded from previous version (7 core characteristics). Objective functions are treated by the weighted sum of penalties or by a multi-objective fashion.

Parallel execution of core simulators with dynamic load balancing is available to reduce turn around time[(8)-(10)]. Since most workstations are connected via network, inter-process communication can be easily established using capability of the Unix operation system. The GALLOP code utilizes the socket mechanism to communicate loading patterns and calculation results, and the internet service daemon (inetd) is used to invoke calculation servers in the remote workstations. Calculation servers can be specified through the input data. In the actual situation, calculation load and capability of each workstation are not the same. Therefore, dynamic load balancing procedure is implemented in the GALLOP code. The program has good scalability even on a heterogeneous workstation cluster; the parallel efficiency was larger than 90% using a cluster composed of seven heterogeneous workstations.

Recently, loading pattern screening capability based on predicted core characteristics by neural networks were added to the GALLOP code[(11)]. The neural network implemented in the GALLOP code can predict the radial peaking factor through cycle burnup, the cycle length and the maximum burnup. These predicted core characteristics are compared with required limitations and candidates that violate the limitations are rejected. Since prediction by the neural network throughout the cycle burnup is very fast (less than 0.001 second), the candidate screening by the neural network results is effective to improve optimization efficiency.

Examples of a loading pattern optimized with the GALLOP code and that manually optimized by an engineer are shown in Fig.3.

4. Multicycle analysis module, MCA

Multicycle analyses are complicated tasks since various data including plant operation, fuel/BP storage, neutronics design information and so on must be treated properly to obtain adequate results. In the INSIGHT system, the MCA module is responsible for automated multicycle analyses[(1)]. The MCA module was recently completely rewritten to enhance its flexibility. Major features of the MCA module are described below.

Number of feed assemblies and inventory of burnable poisons are automatically adjusted based on required cycle length and limitation on the moderator temperature coefficient.

The MCA module automatically invokes the GALLOP module to make loading pattern in given fuel and burnable poison inventory. All features of the GALLOP module can be used through the MCA code including various optimization methods available. Therefore, precise evaluation of a fuel management scenario with detailed loading pattern optimization (but time consuming) or rough evaluation with quick loading pattern search can be controlled through input data of the MCA module. Both types of evaluations are important to find out strategic fuel management scenarios. Since the parallel calculation capability of the GALLOP module can be also used through the MCA module, it greatly contributes to reduce turn around time of detailed evaluation of successive multicycle analyses.

Calculation conditions in each cycle, e.g. the safety limitations, the loading pattern search methods, available fuel/BP, discharge fuel/BP, can be independently specified in the MCA module.

Fuel and burnable poison inventory search is performed based on rules provided in input data of the MCA module. Since the search rules of fuel/BP inventory are not hardwired in the MCA module, user can change search strategy according to their own needs. Note that the rule is given in a kind of priority table.

Limitation of feed fuel and burnable poison can be treated explicitly including the date of their transportation to the nuclear power station. The discharge fuel and burnable poison can be also treated explicitly. They are important in the case of emergency core re-design due to fuel trouble.

The input data for the MCA module can be automatically generated through GUI. Hence minimum user input is required to perform typical calculation by the MCA module. Detail calculation conditions can be specified by slightly modifying the input data.

All the calculation results by the MCA module can be saved in the pattern library of the INSIGHT. Therefore, other modules in the INSIGHT system, i.e. the interactive loading pattern design, the loading pattern optimization and reporting modules can retrieve all the calculation results of MCA.

5. Interactive loading pattern design modules, PATMAKER

The INSIGHT has interactive loading pattern design tool, PATMAKER(1). By using the PATMAKER module, manual modification of a loading pattern can be easy performed. Major features of the current PATMAKER module are:

· Core-to-core and/or pool-to-core shuffling of fuel assemblies/burnable poisons can be performed by point and click through mouse operations.
· Core calculations can be performed through menu selection, which includes the depletion, the moderator temperature coefficient and the shutdown margin calculations.
· Any loading patterns developed in the INSIGHT system can be retrieved in the PATMAKER module through the pattern library. For example, loading pattern search results by the GALLOP module can be manually modified by the PATMAKER module.
· Core characteristics can be displayed through user configurable color map.

The PATMAKER module is currently rewritten to extend its capability and maintainability. Current version of the PATMAKER module is written in C language, but the next version is being written in C++ language. Examples of additional features in the next version are:

· Auto-loading of fuel assemblies according to pre-defined reactivity rank map.
· Auto-shuffling of fuel assemblies through enumerated binary exchange.
· Tracking of fuel assembly/BP movement during shuffling operation.
· Enhanced error checking.

The next version will be released at 3rd quarter of 2001.

6. Summary

In this paper, major modules of the INSIGHT system and their features including recent enhancements are briefly described. Currently, the INSIGHT system is used for not only for regular in-core fuel management calculations but also for wide range of in-core fuel management studies such as:

· Study of multiple feed enrichments splitting.
· Evaluation of fuel cycle cost improvement by relaxation of the radial peaking factor limitation.
· Evaluation of optimum cycle length.
· Evaluation of optimum Gadolinia design.

Due to automation capability of the INSIGHT system, various scoping analyses can be carried out with minimum engineers' workload. Furthermore, since calculation results can be summarized in well-organized printouts, evaluation and interpretation of the calculation results can be easily performed.

Japanese utilities are also using the INSIGHT system and their experiences are feedback to improve usability of the INSIGHT system.

References

1. A. YAMAMOTO, et al., "INSIGHT:An Integrated Scoping Analysis Tool for In-Core Fuel Management of PWR," J. Nucl. Sci. Technol., 34, 847 (1997).
2. A. YAMAMOTO, "Loading Pattern Optimizations Using Genetic Algorithms," Proc Int. Conf. Physics of Reactors (PHYSOR96), Mito, Japan, Vol.3, I-48 (Sep. 1996).
3. P. W. POON, G. T. PARKS, "Application of Genetic Algorithms to In-Core Fuel Management Optimization," Proc. Topical Meeting Mathematical Methods and Supercomputing in Nuclear Applications, Karlsruhe, Germany, 2, 777 (1993).
4. A. YAMAMOTO, "A Quantitative Comparison of Loading Pattern Optimization Methods for In-Core Fuel Management of PWR," J. Nucl. Sci. Technol., 34, 339 (1997).

5. H. HASHIMOTO and A. YAMAMOTO, "Application of the Distributed Genetic Algorithm for Loading Pattern Optimization Problems," Proc. Supercomputing in Nuclear Applications 2000 (SNA200), Tokyo, Japan, September, 2000, RE-2-G2 (CD-ROM), (2000).
6. D. J. KROPACKZEK, P. J. TURINSKY, "In-Core Fuel Management Optimization for Pressurized Water Reactors Using Simulated Annealing," Nucl. Technol., 95, 9 (1991).
7. A. YAMAMOTO, H. HASHIMOTO, "Application of Temperature Parallel Simulated Annealing to Loading Pattern Optimizations of Pressurized Water Reactors," Nucl. Sci. Eng. 136, 247 (2000).
8. J. K. AXMANN, "Parallel Adaptive Evolutionary Algorithms for Pressurized Water Reactor Reload Pattern Optimizations," Nucl. Technol., 119, 276 (1997).
9. J. H. LEE, C. H. LEE AND C. H. KIM, "Parallel Computing Fuel Assembly Loading Pattern Optimization by the Simulated Annealing Method," Proc. Trans. Am. Nucl. Soc., 80, 228 (1999).
10. A. YAMAMOTO, H. HASHIMOTO, "Loading Pattern Optimizations in a Distributed Parallel Environment," Proc. Trans. Am. Nucl. Soc., 80, 223 (1999).
11. A.YAMAMOTO, "Application of Neural Network for Loading Pattern Screening of In-Core Optimization Calculations," submitted to Mathematics and Computation 2001 (MC2001), (2001).

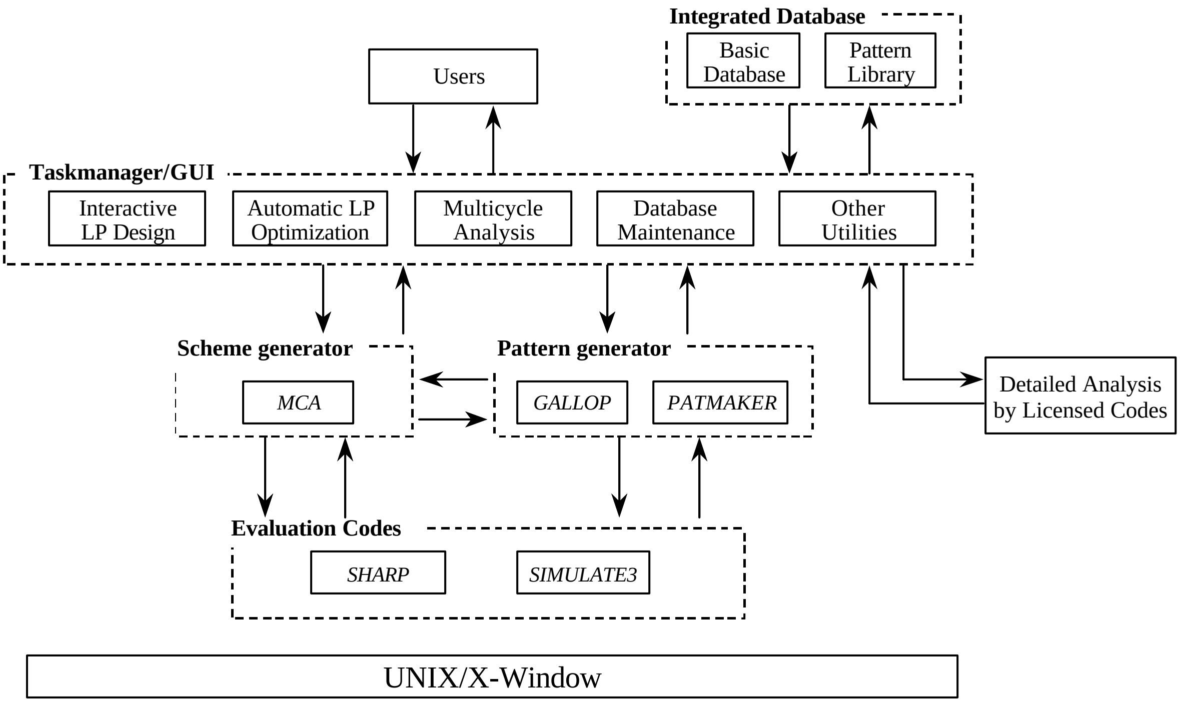

Fig.1 Overview of the INSIGHT system

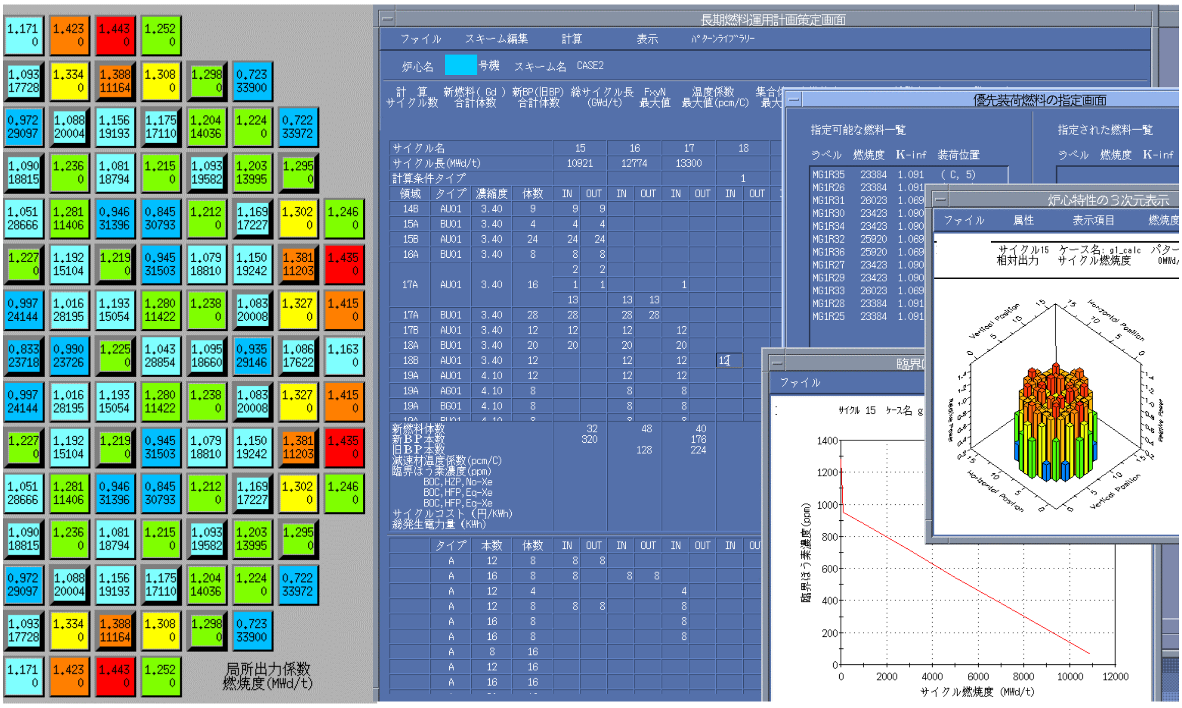

Fig.2 Typical graphics of the INSIGHT system

	H	G	F	E	D	C	B	A
8	4.10 U 23700	4.10 G 0	4.10 G 18900	4.10 G 20400	4.10 G 20300	4.10 G 0	4.10 U 11400	4.10 G 34100
9	4.10 G 0	4.10 U 15000	4.10 U 31700	4.10 U 29500	4.10 G 0	4.10 G 18800	4.10 U 14800	4.10 U 0
10	4.10 G 18900	4.10 U 31700	4.10 G 18300	4.10 G 0	4.10 G 18600	4.10 U 14000	4.10 G 0	4.10 U 0
11	4.10 G 20400	4.10 U 29500	4.10 G 0	4.10 U 11300	4.10 U 29600	4.10 G 0	4.10 U 13600	3.50 U 31800
12	4.10 G 20300	4.10 G 0	4.10 G 18500	4.10 U 29500	4.10 G 0	4.10 G 18000	4.10 U 0	
13	4.10 G 0	4.10 G 18800	4.10 U 13900	4.10 G 0	4.10 G 17800	4.10 U 0	4.10 G 32900	
14	4.10 U 11300	4.10 U 14700	4.10 G 0	4.10 U 13500	4.10 U 0	4.10 G 32900		
15	4.10 G 34200	4.10 U 0	4.10 U 0	3.50 U 31900	--Enrichment(wt%) --Fuel Type(U/G=without Gd/Gd) --Burnup(MWd/t, BOC)			

Radial Peaking factor		1.443	(≤1.450)
Cycle length	(MWd/t)	17429	
Maximum burnup	(MWd/t)	46989	(≤48000)
Mod. temp. coeff.	(pcm/C)	-0.5	(≤-0.5)
Radial power tilt	(%)	0.22	

(a) LP Generated by the GALLOP code(PWR)

	H	G	F	E	D	C	B	A
8	4.10 U 23700	4.10 U 14700	4.10 G 0	4.10 U 29500	4.10 G 0	4.10 U 29700	4.10 U 15000	4.10 G 34100
9	4.10 U 14800	4.10 U 14800	4.10 U 29500	4.10 G 0	4.10 G 20400	4.10 G 18300	4.10 G 0	4.10 U 0
10	4.10 G 0	4.10 U 29500	4.10 U 11300	3.50 U 31900	4.10 G 0	4.10 G 17800	4.10 U 13600	4.10 U 0
11	4.10 U 29500	4.10 G 0	3.50 U 31800	4.10 U 11400	4.10 G 18800	4.10 U 14000	4.10 G 0	4.10 U 31700
12	4.10 G 0	4.10 G 20300	4.10 G 0	4.10 G 18800	4.10 G 0	4.10 G 18600	4.10 U 0	
13	4.10 U 29500	4.10 G 18900	4.10 G 18000	4.10 U 13900	4.10 G 18500	4.10 U 0	4.10 G 32900	
14	4.10 U 15000	4.10 G 0	4.10 U 13500	4.10 G 0	4.10 U 0	4.10 G 32900		
15	4.10 G 34200	4.10 U 0	4.10 U 0	4.10 U 31700	--Enrichment(wt%) --Fuel Type(U/G=without Gd/Gd) --Burnup(MWd/t, BOC)			

Radial Peaking factor		1.445	(≤1.450)
Cycle length	(MWd/t)	17163	
Maximum burnup	(MWd/t)	47104	(≤48000)
Mod. temp. coeff.	(pcm/C)	-0.5	(≤-0.5)
Radial power tilt	(%)	0.14	

(b) LP Generated by an engineer(PWR)

Fig.3 Comparison of loading patterns generated by an engineer and the loading pattern optimization code, GALLOP.