Advances in Engineering Software 34 (2003) 139–162

www.elsevier.com/locate/advengsoft

# Modeling station duty officer operations assistant at Johnson Space Center

Madjid Tavana[a,*], James N. Ortiz[b,1], Susan E. Torney[c,2]

[a]*Department of Management, La Salle University, Philadelphia, PA 19141-1199, USA*
[b]*Systems Management Office, Lyndon B. Johnson Space Center, National Aeronautics and Space Administration, Houston, TX 77058-3696, USA*
[c]*Advanced Projects and Analysis Office, Lyndon B. Johnson Space Center, National Aeronautics and Space Administration, Houston, TX 77058-3696, USA*

Accepted 18 November 2002

**Abstract**

The mission operations directorate (MOD) at the Johnson Space Center (JSC) is responsible for the planning and operation of human space flight missions. MOD is being challenged with sustaining and developing new operations capabilities to support increasingly demanding requirements and to improve its processes to accomplish these missions at higher levels of safety, mission success, and effectiveness. Automation is being considered as an enabling technology to meet the aforementioned challenges. The synergistic combination of flight controllers and intelligent software providing the function of 'operations assistants' (OA) is being pursued as the key implementation of this technology in the Mission Control Center (MCC).

The flight control team (FCT) assesses the condition and operability of the major systems such as electrical power, thermal control, life support, communications, altitude control, and data handling at MCC. OA assist the flight controllers with their tasks of monitoring the status and health of the flight systems. They also help maintain the flight controller's awareness of the operations being performed during the mission and help assure that operational objectives are being met. The station duty officer (SDO) performs the lead operations role for the International Space Station (ISS) during quiescent times when FCT and the flight director are off-duty. SDOs assess the condition and operability of the major ISS systems. This assessment involves monitoring and controlling several periodic processes on systems such as the electrical power, thermal control, life support, communications, altitude control, and data handling systems. The SDO is also responsible for coordinating operations with the Russian FCT. The OA will help the SDO maintain an awareness of all the processes performed on board and will assist with the responses to anomalous conditions. The OA for this position will support the concept of reduced control center staffing during quiescent times. The purpose of this paper is twofold: (1) to present a unique two-stage specification methodology that combines data flow diagrams and petri nets and (2) to apply the proposed methodology in a complex space station system.
Published by Elsevier Science Ltd.

*Keywords:* NASA; Automation technology; Operations assistants; Systems development; Structures analysis; Process modeling; Data flow diagrams; Petri nets

## 1. Introduction

Automation technology has been applied to unmanned spacecraft, robots, and ground-based systems at NASA-Johnson Space Center (JSC). However, it remains to be successfully utilized in human mission operations. Operations Assistants (OA), an automation technology at NASA, are a set of distributed customized systems designed to assist the flight control team (FCT) and flight director (FD) in the Mission Control Center (MCC). This technology, which is required for exploration programs, was developed for the International Space Station (ISS) program through a series of incremental prototypes. Once these prototypes are mature, they will be transferred to other programs, including the Space Shuttle and Mars exploration. OA maintains flight controller awareness of planned activities and operational objectives (mission cognizance) and monitors, tracks, and checks operational processes, events, and systems. In addition, OA provides just-in-time support tailored to events, alarms, and other problems with intelligent software designed to monitor and retrieve relevant data and perform analyses. Finally, OA proposes procedures and actions for troubleshooting and recovery where feasible by assessing

[*] Corresponding author. Tel.: +1-215-951-1129.
*E-mail addresses:* tavana@lasalle.edu (M. Tavana), james.n.ortiz@nasa.gov (J.N. Ortiz), susan.e.torney@nasa.gov (S.E. Torney).
[1] Tel.: +1-281-483-0520.
[2] Tel.: +1-281-483-2866.

0965-9978/03/$ - see front matter Published by Elsevier Science Ltd.
PII: S0965-9978(02)00127-8

the conditions, availability, and operability of resources and systems.

Automation, while it is an expensive investment, has great potential for any organization especially for NASA. For example, according to mission operations studies performed by NASA (internal memo, April 1996), the Shuttle budget, equally allocated across all the flights, is about $400 million per flight. Further allocation of this cost across an average of 10 days results in a cost for each flight day of approximately $40 million. By allocating the $40 million per day cost across the 8 h payload, the hourly single shift rate is approximately $5 million. There are several reasons why NASA is actively, yet cautiously, seeking process automation. Process automation is generally considered: (a) in areas where operators are not capable of performing particular tasks because of their inability to sense and react to the environment, (b) in order to ensure consistent, repeatable processes or performance, (c) to perform tasks that are extremely tedious or dangerous to human operators, (d) to free the operator to perform more appropriate tasks such as decision making under uncertainty, and (e) to reduce costs by replacing humans with machines. At the same time, there are tradeoffs to be considered. For every $1 million cost of pursuing automation in a particular area, the program can support 1 person for nearly 10 years at current contract loaded rates.

This paper addresses the emerging field of modeling and specification used to model a large and complex automation project at NASA. Flight rules and procedures, log events and organization, and system goals and constraints are among some of the knowledge-based data sources used in this study. Large and complex information systems are difficult and expensive to build and maintain. The need for formal system specifications has led to the introduction of several modeling tools. We focus on structured analysis to translate the user's vague and ambiguous requirements into precise and formal specifications required for system implementation [6,8].

Structured analysis supports systems analysts in defining user requirements and developing creative solutions. Many structured analysis tools have been developed over the past 25 years to aid systems analysts. A data flow diagram (DFD), representing information flow, system boundaries, and environmental interactions, is a popular structured analysis technique used during the analysis phase of development to communicate with non-technical users [1]. DFD, a popular technique introduced in the late 1970s, is used by most systems analysts in today's software development projects [11,12]. The popularity of DFD stems from its use of intuitively defined concepts and notations. However, DFDs lack a formal basis to rigorously investigate semantic properties of the application. France [7] has addressed this problem with semantically extended DFD, capable of producing formal specifications. Tao and Kung [19] address this problem by proposing precedence relation, an abstraction of the functional semantics that specifies the 'is-used-to-produce' relationships among the data flows. Also, DFDs do not represent control logic in the systems being modeled. Ward [22] proposes transformation scheme, a notation and formation rule for building a comprehensive system model. Several authors have tried to address the control logic problem inherent in DFDs by integrating DFDs with petri nets (PNs) [4,17,20]. However, these studies do not provide a direct translation of DFDs into PNs for further verification and enhancement of the process model.

Structured analysis emphasizes the description of 'what' or the functional aspects of the system. However, a second dimension of modeling complexity, control flow, emphasizing the description of 'how', is often as important as functional analysis [23]. Some extensions to structured analysis methodology such as the structured analysis and design technique [18] and operational-requirements approach [24] have considered the explicit representation of control. However, most of these techniques are inadequate for modeling concurrent and asynchronous systems [13].

PNs were initially defined by Carl Adam Petri [16] and later refined and named after him by Holt [9]. Peterson [15] elegantly discusses the dynamic behavior of PNs, while Murata's tutorial review paper provides a thorough review of PNs' history and applications [14]. PNs and their modifications provide a rich and versatile approach to modeling. They have been proven to be useful for the modeling and analysis of several classes of systems including communication systems [2], knowledge-based systems [10], and process control systems [3,5]. Wang et al. [21] have used PNs to design a coordination system for intelligent mobile robots.

One of the strengths of PNs is their broad based applicability to a wide range of systems. Ordinary place-transition PNs are used in this study. Place-transition nets are classical models with black tokens that model the control flow in business systems quite comfortably and provide efficient ways of 'qualitative' verification. In addition to their modeling power, ordinary PNs are described as both a graphical and mathematical tool.

As a graphical tool, PNs provide a visual medium for a modeler to describe a complex system. As a mathematical tool, PN models can be represented by linear algebraic equations, creating the possibility for the formal analysis of the model [25]. Mathematical properties of PN can be classified into (a) structural properties that depend on the net structure and (b) behavioral properties that depend on the initial and subsequent markings. Mathematically, analyses of PNs can be based on enumerating all possible markings to form reachability trees and/or through methods and theories in discrete mathematics like matrix equations. We use both graphical and mathematical properties of PN in modeling the SDO OA. As a graphical tool, PNs were used to enhance communications and produce accurate and complete specifications while the mathematical properties of PNs were used to detect deadlock, overflow, and irreversible situations. Performance evaluation is also possible through mathematical analysis of the model using stochastic timed PNs. However, due to the complexities of the SDO operations, we did not use stochastic timed PNs.

We present a two-stage specification methodology that combines two independent modeling tools in conjunction: DFD and PN. The proposed methodology, data flow petri nets (DFPN), is intended to enhance structured analysis by providing process and control specifications that relate the descriptions of 'what' and 'how' more closely to the actual system implementation. DFPN is especially useful for systems that may possess concurrent, distributed, asynchronous, parallel, or event-driven qualities. We illustrate DFPN in the context of a SDO OA project. SDOs are particularly good candidates for applying this methodology since they perform several concurrent tasks during their daily shifts at MCC. In Section 2, we illustrate our methodology in the context of a simple supermarket check-cashing problem. A detailed description of the SDO DFPN system developed in this study is presented in Section 3. Section 4 shows the implementation of SDO OA with remote agent (RA) followed by the conclusion and future research directions presented in Section 5.

## 2. Data flow petri nets example

In order to understand the specification techniques utilized here, we begin with an illustrative check-cashing example in a supermarket:

> When the customer has a valid check-cashing card, check will be accepted for the amount of purchase plus $25.00. If the customer does not have a valid check-cashing card and the purchase is less than $20.00, two forms of identification must be shown in order to pay by check in the amount of purchase. Otherwise, the store manager must be called to authorize the acceptance of the check for the amount of purchase.

First, we develop the DFD shown in Fig. 1 describing the check-cashing problem presented above. DFDs are graphs with four different types of components called processes, data stores, external entities, and data flows. A process is represented with a circle, a data store with an open-ended

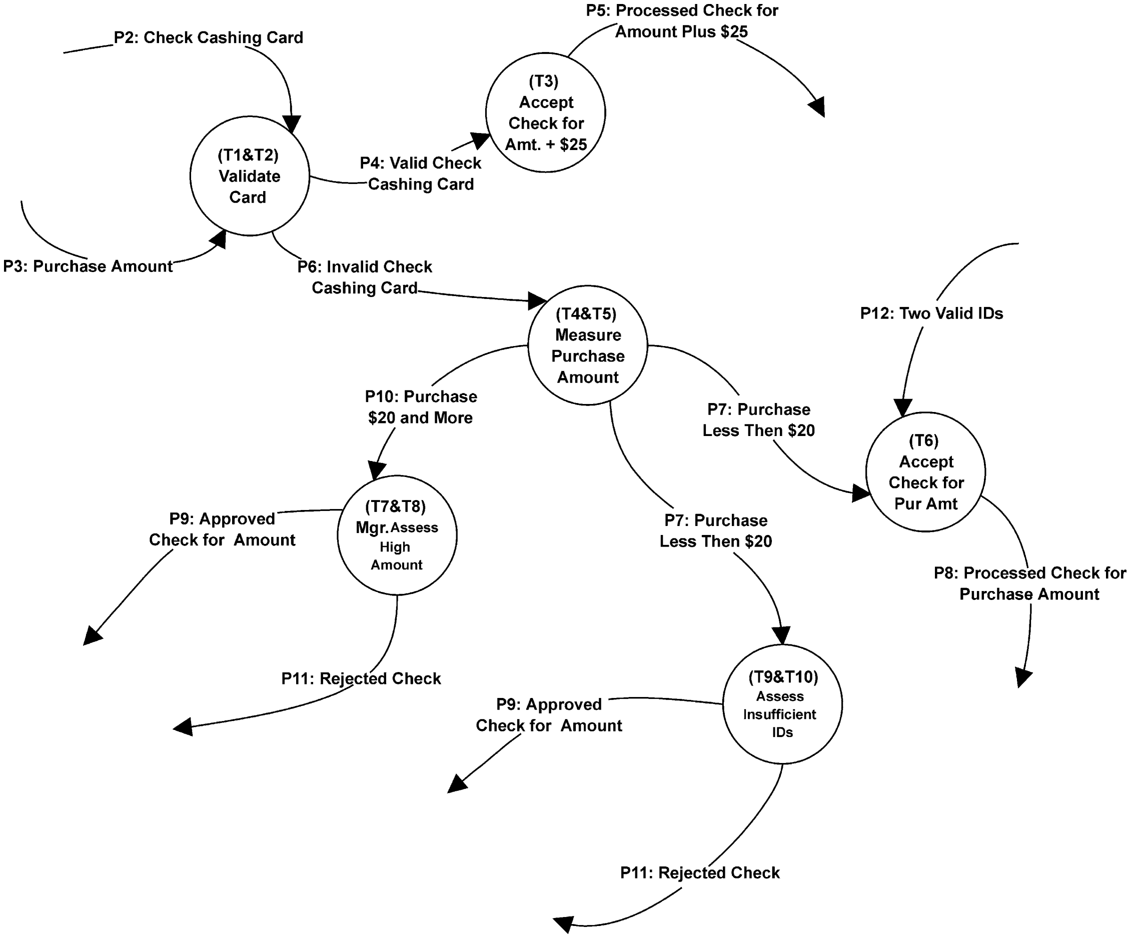

P2: Check Cashing Card
P5: Processed Check for Amount Plus $25
(T3) Accept Check for Amt. + $25
(T1&T2) Validate Card
P4: Valid Check Cashing Card
P3: Purchase Amount
P6: Invalid Check Cashing Card
(T4&T5) Measure Purchase Amount
P12: Two Valid IDs
P10: Purchase $20 and More
P7: Purchase Less Then $20
(T6) Accept Check for Pur Amt
P9: Approved Check for Amount
(T7&T8) Mgr. Assess High Amount
P7: Purchase Less Then $20
P8: Processed Check for Purchase Amount
P11: Rejected Check
P9: Approved Check for Amount
(T9&T10) Assess Insufficient IDs
P11: Rejected Check

Fig. 1. Check cashing data flow diagram.

rectangle, an external entity with a square, and data flows are directed arcs connecting processes to either processes, data stores, or external entities. External entities are normally shown in the highest level DFD, called context diagrams. Directed arcs into a process are input data flows while directed arcs out of processes are output data flows. In order to validate a check-cashing card, check-cashing data and purchase amount data flows are the input to the card validation process. This validation results in a valid or invalid check-cashing card. If the customer has a valid check-cashing card, the check is accepted in the amount of purchase plus $25. Invalid check-cashing cards go through the purchase amount evaluation. If the purchase amount is $20 or more, a manager is called to approve/reject the check. For purchases less than $20, checks are accepted for the purchase amount if the customer has two valid forms of identification. A manager is called to approve checks without two valid IDs.

Once the DFD is developed, it is converted into a PN. We have developed an algorithm that automatically converts DFDs into PNs. A summary of the conversion rules are given below.

(a) All processes are converted into transitions.
(b) All data flows are converted into places.
(c) DFD processes with mutually exclusive outputs are split, resulting in one PN transition for each of the mutually exclusive outputs. This split requires a signal variable to control the resulting multiple transitions that were created. To illustrate, the 'validate card' process in our check-cashing example has mutually exclusive outputs: 'valid check cashing card', 'invalid check cashing card'. A signal variable 'valid_card' is needed to control the two transitions made from 'validate card' ('validate card: valid' and 'validate card: invalid').

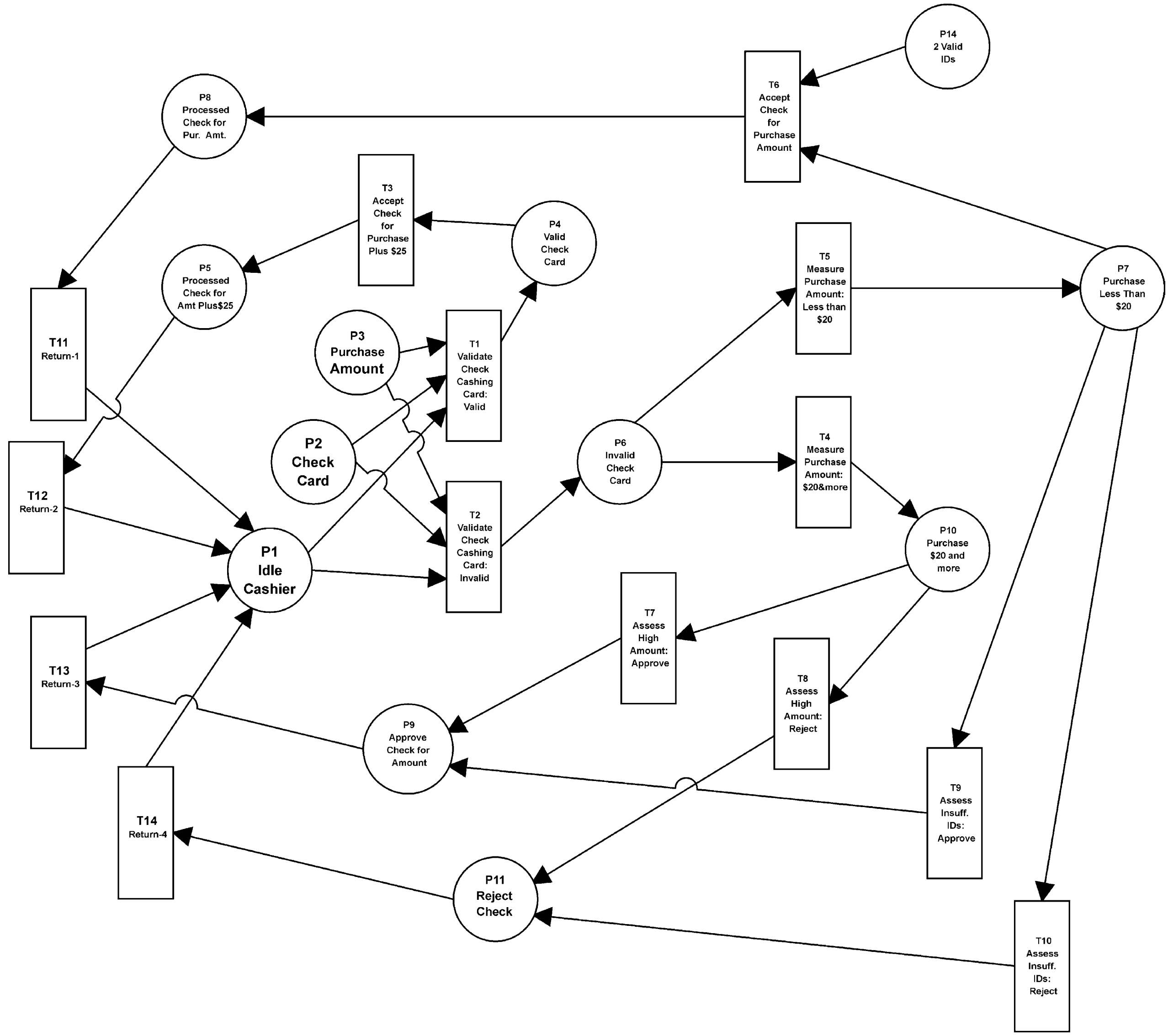

P14 2 Valid IDs
T6 Accept Check for Purchase Amount
P8 Processed Check for Pur. Amt.
T3 Accept Check for Purchase Plus $25
P4 Valid Check Card
P5 Processed Check for Amt Plus$25
T5 Measure Purchase Amount: Less than $20
P7 Purchase Less Than $20
T11 Return-1
P3 Purchase Amount
T1 Validate Check Cashing Card: Valid
P2 Check Card
P6 Invalid Check Card
T4 Measure Purchase Amount: $20&more
T12 Return-2
T2 Validate Check Cashing Card: Invalid
P1 Idle Cashier
P10 Purchase $20 and more
T7 Assess High Amount: Approve
T13 Return-3
T8 Assess High Amount: Reject
P9 Approve Check for Amount
T9 Assess Insuff. IDs: Approve
T14 Return-4
P11 Reject Check
T10 Assess Insuff. IDs: Reject

Fig. 2. Check cashing petri net.

(d) An ‘initial state’ place must be created for the PN. This provides a good beginning point for initiating the PN processing. It also gives a good return point so that the PN can be executed multiple times.

(e) For each data flow exiting the DFD, a ‘return’ transition is created so that a token can be sent to the ‘initial state’ place, thereby enabling the system to be re-initiated after it has completed a cycle. These return transitions can be numbered sequentially to generate a unique ID for each return (e.g. return-1, return-2, return-3, etc.).

(f) Data stores do not transform or produce data.

(g) Double-headed arrows are treated as two single-headed arrows and every arrow is translated into a distinct place in the PN.

A detailed example of a DFD to PN transition specification is presented in Appendix A for the check-cashing problem. In general, all processes are changed into transitions and all data flows are changed into places. Mutually exclusive data flows in DFDs cause a process to be broken down into two equivalent transitions in a PN. PNs are directed bipartite graphs with two different types of nodes called places and transitions. A place $p$ is presented with a circle and a transition $t$ is presented by a rectangle. The nodes are connected through directed arcs. Directed arcs from $p$ to $t$ create input places while directed arcs from $t$ to $p$ create output places. Each place contains zero or multiple tokens drawn as black dots. The execution of PN may affect the number of tokens in a place. As it is shown in Fig. 2, our check cashing PN includes 13 places and 16 transitions. In this example, there may be three tokens: one in place ‘idle cashier’ ($p_1$), one in place ‘check cashing card’ ($p_2$), and one in place ‘purchase amount’ ($p_3$). The token in $p_1$ indicates that the cashier is free, the token in $p_2$ indicates that the customer has a check cashing card, and the token in $p_3$ shows that the customer has finalized his/her selection and the purchase amount is known. A transition is called enabled when each of its input places has enough tokens. A transition can be fired only if it is enabled. When a transition is fired, tokens from input places are used to produce tokens in output places. With a token each in $p_1$, $p_2$, and $p_3$, transitions $t_1$ and $t_2$ (validating the check cashing card) are enabled. If the check-cashing card is valid, $t_1$ is fired. Firing $t_1$ means consuming three tokens, one from $p_1$, one from $p_2$, and one from $p_3$, and producing one

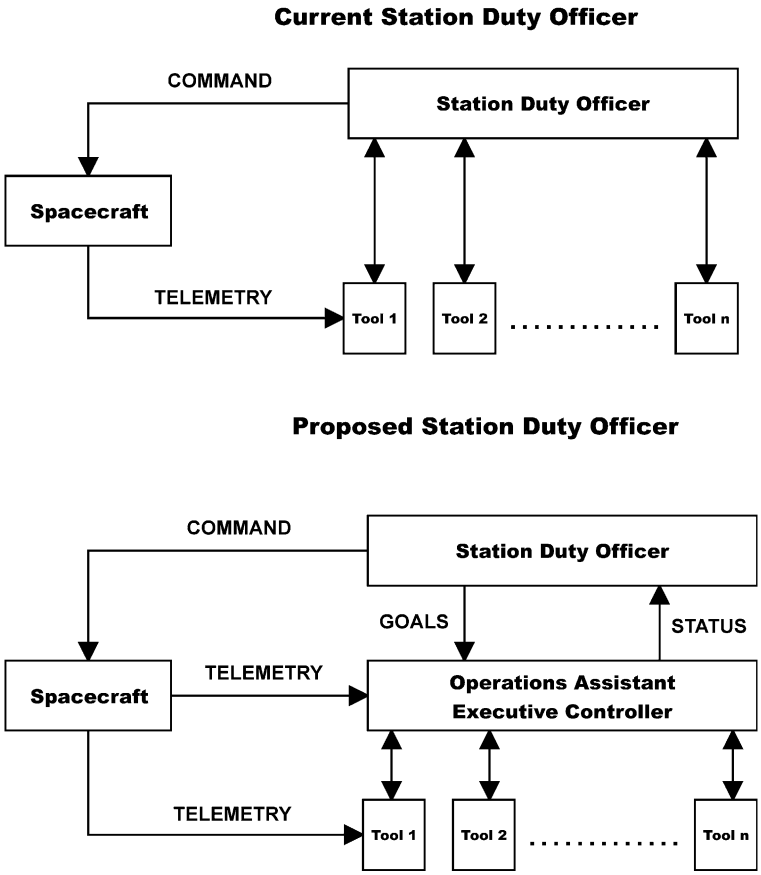


Fig. 3. Station duty officer operations assistant.

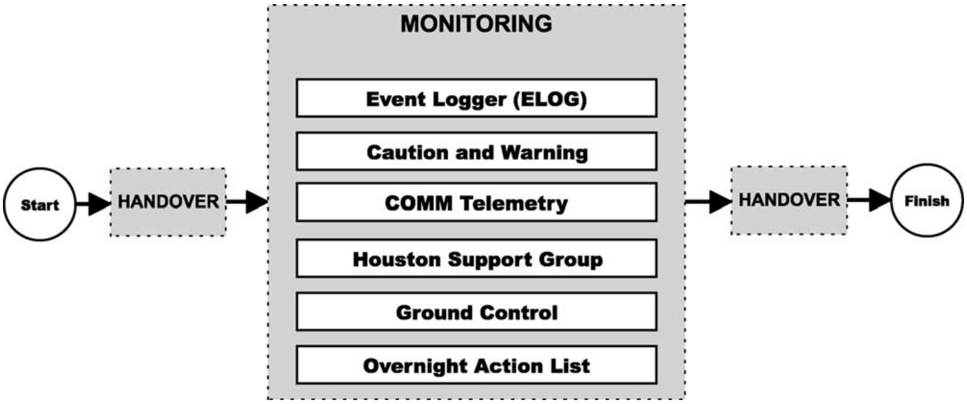


Fig. 4. Station duty officer duties.

token for $p_4$. Now transition $t_4$ (accept check for purchase amount plus $25) is fired producing a token for $p_5$. Next, transition $t_{14}$ is fired and a token is produced for $t_1$ meaning the cashier is now free to process the next customer. As long as there are tokens in $p_2$ and $p_3$, the two transitions $t_1$ and $t_2$ are enabled and this cycle is repeated depending on the specific circumstances. Note that the cashier modeled by this PN processes only one customer at a time.

## 3. Station duty officer DFPN

The station duty officer (SDO) performs the lead ISS operations role during quiescent periods when the FCT and FD are off-duty. The flight controller is responsible for alerting appropriate FCT discipline and the FD if an off-nominal condition develops. The SDO will maintain the radio-frequency command and telemetry link with the ISS via the early communications subsystem (ECS) to assess the condition and operability of major station systems such as electrical power, thermal control, life support, communications, attitude control, and data handling systems.[3]

OA assist SDOs on their tasks concerning monitoring the status and health of the ISS. The OA will help the SDO maintain an awareness of all the processes performed on board and will assist with the responses to anomalous conditions. The OA for this position will support the concept of reduced control center staffing during quiescent times (see Fig. 3).

SDOs, as it is shown in Fig. 4, are primarily responsible for two *sequential* activities, monitoring and handover. Monitoring activities involve *concurrent* observation and examination of several computer displays in MCC including the event logger (ELOG) display, SDO display, caution and warning (C&W) logger display, and emergency, warning, caution, and advisory (EWCA) status display. In addition, SDOs listen to various communication loops such as Houston support group (HSG) and ground control (GC). Handover activities described in detail later in this paper are the result of shift changes between FCTs and SDOs or SDOs and SDOs.

Two systems are described in this paper, ELOG and C&W processing. Both modules are among a series of incremental prototypes developed for SDO OA. Additional modules presented in Fig. 11 through 18 are included in Appendix B. ELOG processing shows how SDOs record critical events, messages, or ELOGs in the SDO log and respond to them by following the procedure specified in the SDO anomaly response instruction (ARI). The SDO log is a continuous report recording the time and event associated with mostly ELOG and C&W messages. Each FCT discipline has its own customized ELOG messages collected from the information down-linked from the ISS and distributed throughout MCC via the information sharing protocol (ISP). The ELOG display combines all messages identified by each FCT discipline as necessary for SDOs to monitor. ELOGs, which contain knowledge about critical events, are the primary monitoring tools used by SDOs. An ELOG message consists of the time, an identification code, and an abbreviated description. Once an ELOG message is initiated, the SDO must determine whether the message is routine. Non-routine or irregular messages are further verified by referring to the SDO display. The SDO display, complementing the ELOG display, shows additional discipline-specific data concerning various flight controllers such as CATO, ODIN, ADCO, ECLSS, PHALCON, THOR, and OPS PLAN. Anomalous messages are further assessed by referring to the ARI and overnight action list (OAL). The OAL initiated by the previous FCT contains special instructions for the SDO from each FCT discipline.

[3] SDOs interact with the following ISS FCTs: communications and tracking officer (CATO), on-board data and interface networks (ODIN), attitude dynamics control officer (ADCO), environmental control and life support system (ECLSS), power, heating, articulation, and lighting control (PHALCON), thermal operations and resources (THOR), operations planner (OPS PLAN), ground control (GC), assembly and checkout officer (ACO), extravehicular activity officer (EVA), operation support officer (OSO), trajectory operations officer (TOPO), capsule communicator (CAPCOM), surgeon (SURGEON)/biomedical engineer (BME), and space radiation analysis (RADIATION).

This may include any temporary modifications to normal procedures, what to expect during the upcoming SDO shifts, or any specific changes to ARI. Occasionally, the OAL may contain additional instructions concerning specific messages. SDOs respond to critical messages by logging each event (time and message) in the SDO log and following the procedure specified in the SDO ARI. ARI is a binder that contains all the critical messages an SDO can expect to see. The instruction identifies the FCT discipline affected by each anomaly and any corrective procedure (including non-critical commanding procedures). ARI direct SDOs to perform specific tasks such as checking the status of a parameter, making comparisons and/or judgments, and notifying the appropriate on-call FCT discipline representative and/or FD. ARIs are grouped together in order to eliminate the need for a written procedure for every possible message. This process is described in detail with DFD in Fig. 5 and PN in Fig. 6.

The C&W module describes monitoring of onboard C&W messages. SDOs use two displays to monitor C&Ws.

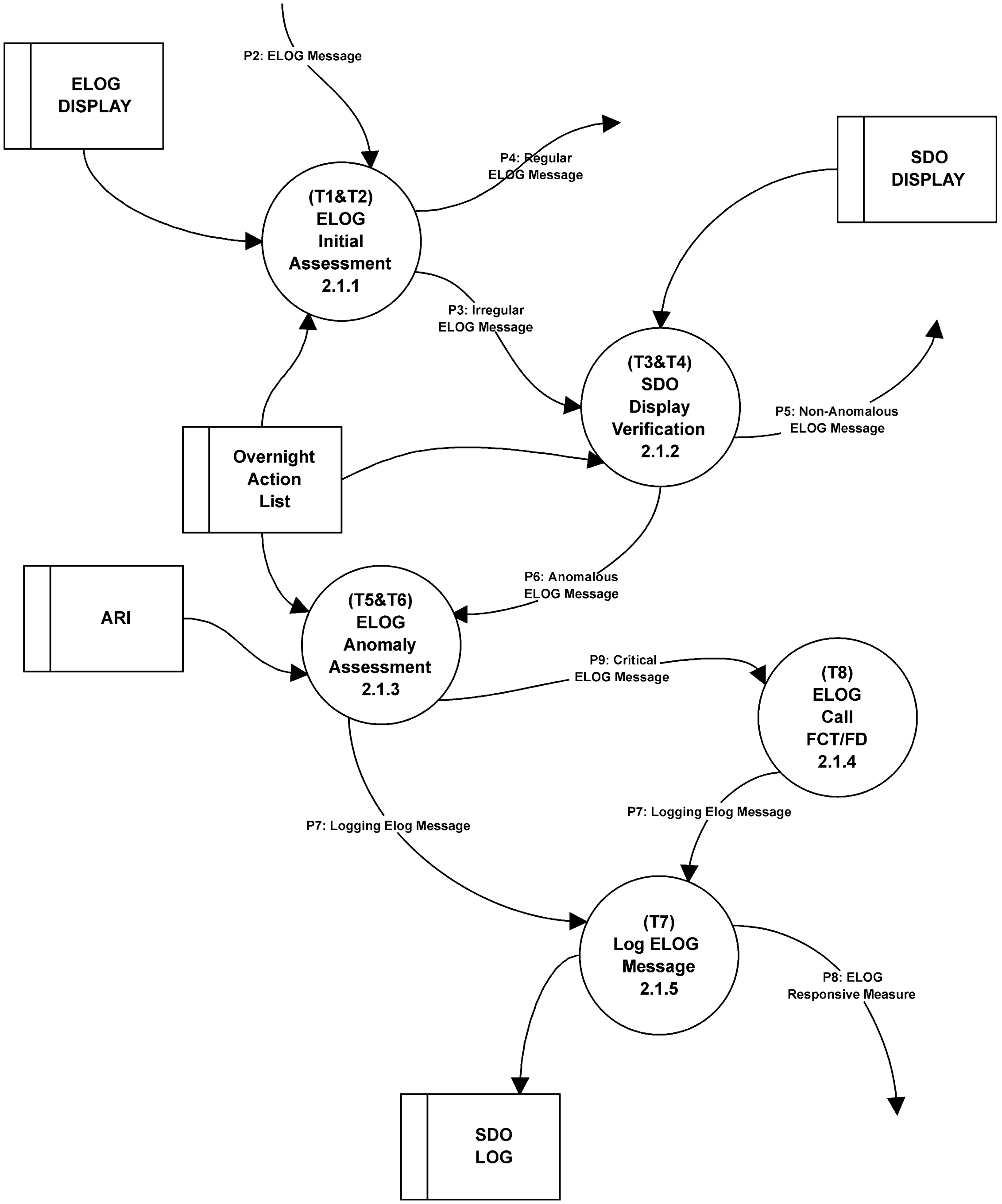

ELOG DISPLAY
P2: ELOG Message
(T1&T2) ELOG Initial Assessment 2.1.1
P4: Regular ELOG Message
SDO DISPLAY
P3: Irregular ELOG Message
(T3&T4) SDO Display Verification 2.1.2
P5: Non-Anomalous ELOG Message
Overnight Action List
ARI
P6: Anomalous ELOG Message
(T5&T6) ELOG Anomaly Assessment 2.1.3
P9: Critical ELOG Message
(T8) ELOG Call FCT/FD 2.1.4
P7: Logging Elog Message
P7: Logging Elog Message
(T7) Log ELOG Message 2.1.5
P8: ELOG Responsive Measure
SDO LOG

Fig. 5. ELOG processing data flow diagram.

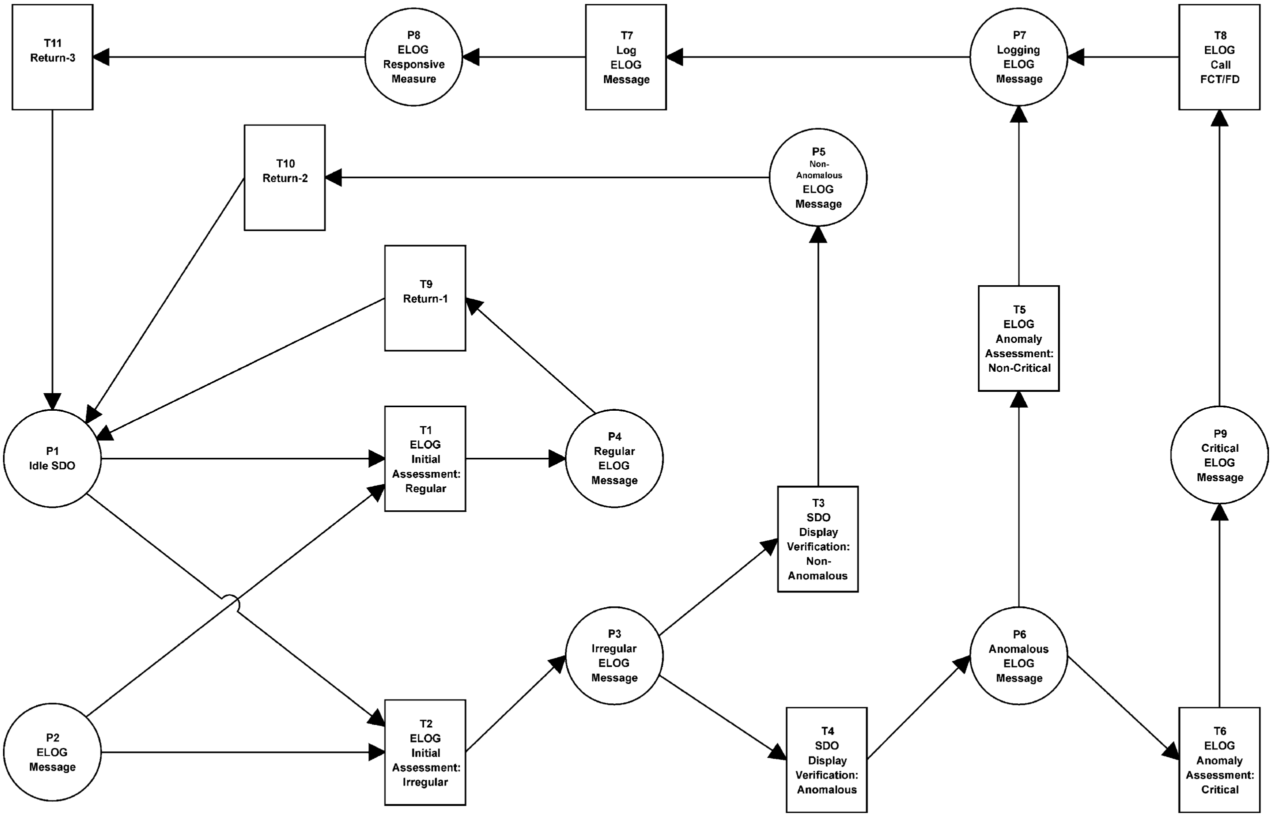

T11 Return-3
P8 ELOG Responsive Measure
T7 Log ELOG Message
P7 Logging ELOG Message
T8 ELOG Call FCT/FD
T10 Return-2
P5 Non-Anomalous ELOG Message
T9 Return-1
T5 ELOG Anomaly Assessment: Non-Critical
P1 Idle SDO
T1 ELOG Initial Assessment: Regular
P4 Regular ELOG Message
P9 Critical ELOG Message
T3 SDO Display Verification: Non-Anomalous
P3 Irregular ELOG Message
P6 Anomalous ELOG Message
P2 ELOG Message
T2 ELOG Initial Assessment: Irregular
T4 SDO Display Verification: Anomalous
T6 ELOG Anomaly Assessment: Critical

Fig. 6. ELOG processing petri net.

The EWCA status display shows the down-linked values of the onboard flight software counters for the number of EWCA messages generated by the onboard software. The C&W logger display shows ground-processed data generated from the down-linked EWCA messages. The C&W logger display replicates the text of the EWCA message that the onboard software displays to the crew on the ISS PC in response to an alarm. When the EWCA display shows a counter increase, the SDO refers to the C&W display for further verifications. If the counter increase is found to be anomalous, the SDO refers to the ARI and OAL in order to determine whether the message is critical or not. Critical messages are relayed to the FCT immediately. SDOs respond to critical C&W messages by logging each event (time and message) in the SDO log and following the procedure specified in the SDO ARI. Figs. 7 and 8 describe DFD and PN for C&W process.

SDOs also listen to HSG on the Houston consult loop to answer questions from the Russian FCT, identify Russian operations that may impact US operations, confirm completion of US-initiated procedures/commands using Russian ground station communications passes, and respond to requests from the Russian FD to speak to the US FD. Should the Russian FD wish to speak with the US FD, the SDO will notify the US FD and activate the FCT per instructions in the SDO ARI. In addition, SDOs listen to the GC and command loops for consulting on the impacts of near real-time tracking and data relay satellite system (TDRSS) service re-planning to US station operations, provide instructions to the GC regarding TDRSS ground network configuration to support the early communication subsystem (ECS) link with the ISS, and coordinate TDRSS coverage changes with the Russian team through the HSG. SDOs also process OALs that are initiated by the FCT and include the pre-positioned command packages (PPCPs). The OAL, described earlier, contains special instructions for the SDO from each FCT discipline. Process and control specifications for CT, HSG, GC, and OAL are presented in Appendix B.

Once all modules are completed, synthesis is performed to develop an integrated PN for SDO operations. This is accomplished by identifying overlapping places and transitions and eliminating them. Fig. 9 presents the overall PN for SDO operations. The data dictionary for this PN is presented in Fig. 10.

Handover activities, described in Fig. 11, result from shift changes between FCTs and SDOs or SDOs and SDOs. SDOs normally work in three shifts from Monday to Thursday and two shifts from Friday to Sunday. There is a 1 h shift overlap used for handover briefings. Handover activities are the result of these shift changes. There are three different types of handover activities, FCT to SDO, SDO to SDO, and SDO to FCT. During the FCT to SDO handover, the FCT initiates an SDO OAL described earlier. An SDO to SDO handover includes the SDO log along with a verbal briefing. SDOs usually

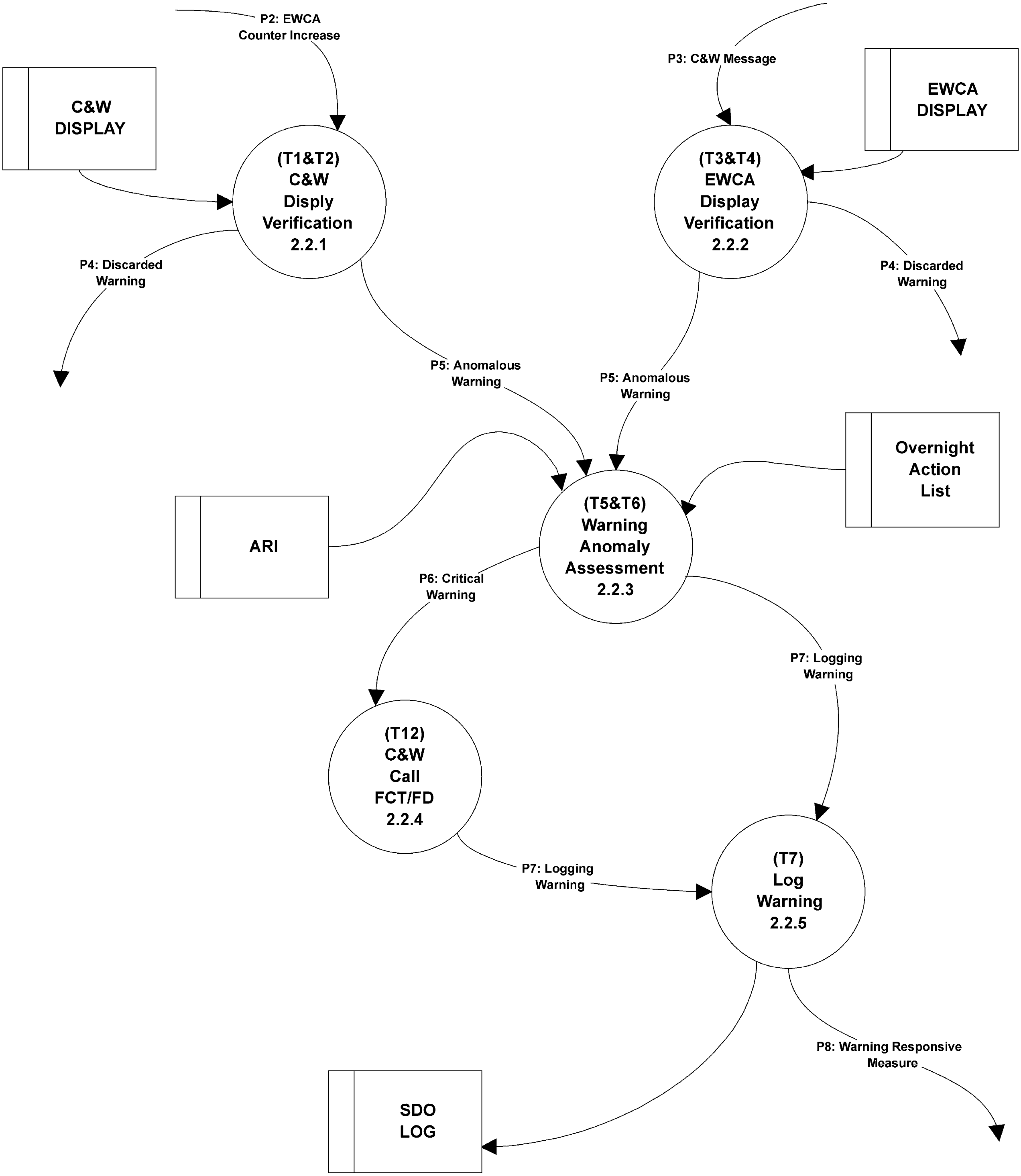


Fig. 7. C&W processing data flow diagram.

complement this report with a verbal briefing cautioning the incoming SDO to conditions relevant to any alerts. The SDO handover to FCT involves a shift handover report for the incoming FCT, which is a synopsis of SDO logs in prior shifts. This report serves as the basis for updating incoming FCT and includes a summary of activities since a FCT was last on duty. This report is usually accompanied by an informal and verbal briefing of certain FCT disciplines on conditions relevant to any alerts.

## 4. SDO OA proposed implementation with remote agent

The RA was developed by researchers at NASA’s Ames Research Center and Cal Tech’s Jet Propulsion Laboratory

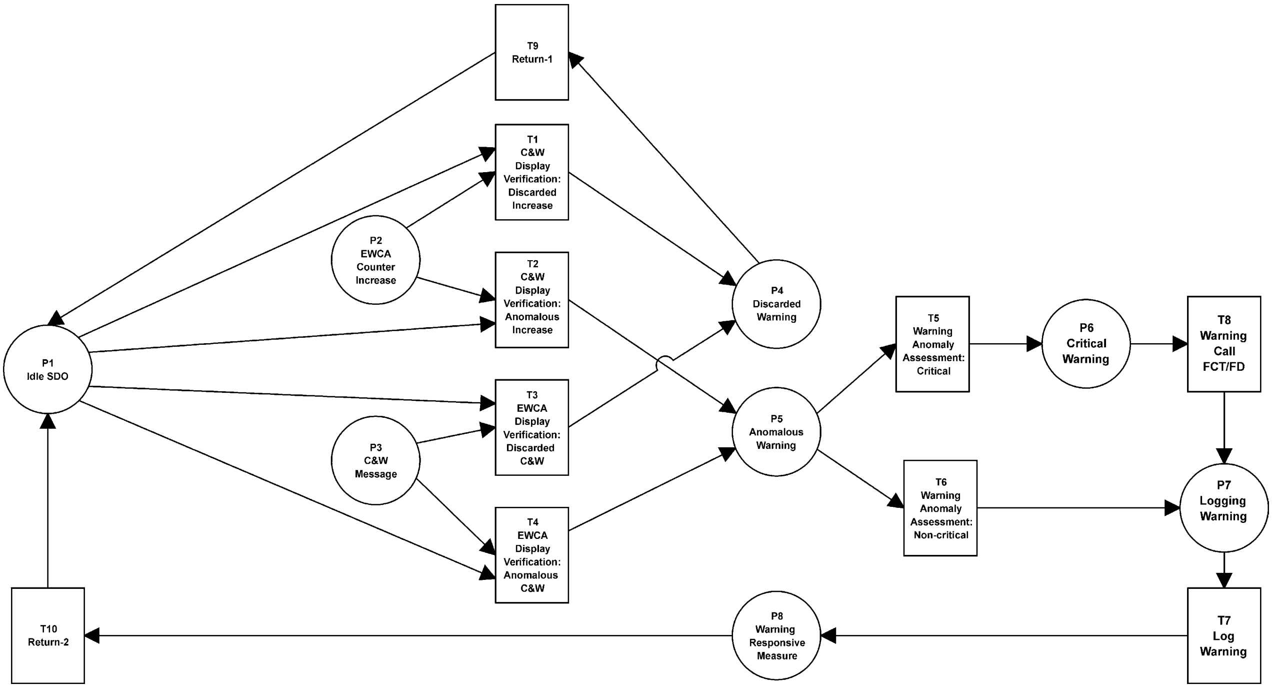


Fig. 8. C&W processing petri net.

in response to a high-priority request to meet the goals of NASA to have a 'virtual presence' in space. The RA software uses model-based reasoning algorithms along with constraint-based and goal-directed planning and execution algorithms. The software, as it is shown in Fig. 12, includes a planner/scheduler (PS), a smart executive (EXEC), and a mode identification and recovery (MIR). PS produces flexible plans and specifies the basic activities that must take place to accomplish the mission's goals, EXEC carries out the planned activities, and MIR monitors the health of the spacecraft and attempts to correct any problems that occur.

The suggested implementation of OA with RA allows SDOs to focus on more interesting and challenging tasks and not be saddled with routine operational functions, such as monitoring telemetry. SDOs become more skilled and knowledgeable about how the different spacecraft systems interact by allowing them to focus on the operation as a whole, leaving the details for the spacecraft software to monitor and control. Finally, MCC should be able to operate more spacecraft equipped with this technology. For more information about RA, readers should refer to http://rax.arc.nasa.gov/

The SDO OA designed in this study has not yet been implemented with RA. There are studies underway to assess several RA implementation strategies. Preliminary results of the study show that a likely implementation scenario could be through a series of programs developed with MIR and EXEC. MIR could be used to implement routine and programmable tasks such as log message (T37) or call FCT/FD–ELOG (T7). The following transitions could be programmed with RA: T7, T8, T9, T16, T17, T24, T25, T26, T31, T32, T37, T38, T39, T40, T47, and T48. Next, EXEC could be used to develop a set of 16 decision support systems (DSS) modules to help SDOs make various assessments, verifications, and investigations. The following is a listing of DSS modules for processing ELOGs: initial assessment (T1 and T2), display verification (T3 and T4), and anomaly assessment. C&W DSS modules include: C&W display verification (T10 and T11), EWCA display verification (T12 and T13), and anomaly assessment (T14 and T15). COMM telemetry modules include: comparison (T18 and T19), investigation (T20 and T21), and anomaly assessment (T22 and T23). HSG modules include: COMM assessment (T27 and T28) and anomaly assessment (T29 and T30). GC group modules include: COMM assessment (T33 and T34) and anomaly assessment (T35 and T36). Finally, OAL modules include: comparison (T41 and T42), investigation (T43 and T44), and anomaly assessment (T45 and T46).

## 5. Conclusion and future research directions

Mission operations at the JSC is responsible for the planning and conduct of human space flight missions. This includes planning the trajectories and the activities for the missions, training the crews and mission control personnel, and controlling flight activities and systems. Mission operations are challenged with sustaining and developing new operations capabilities to support increasingly demanding requirements. These include complex, new operational scenarios such as multi-vehicle control, long distance missions and extended duration operations. The length of

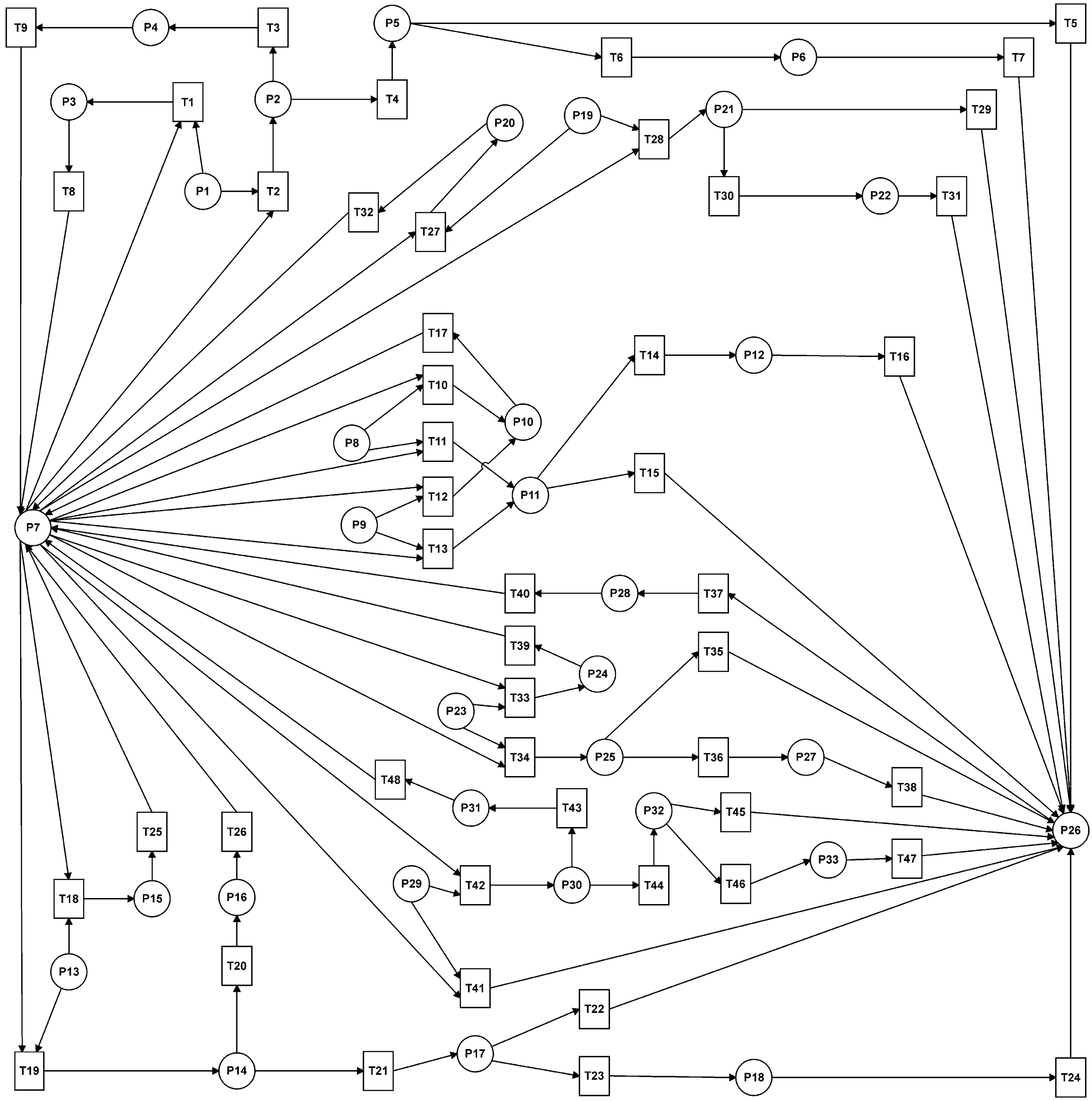
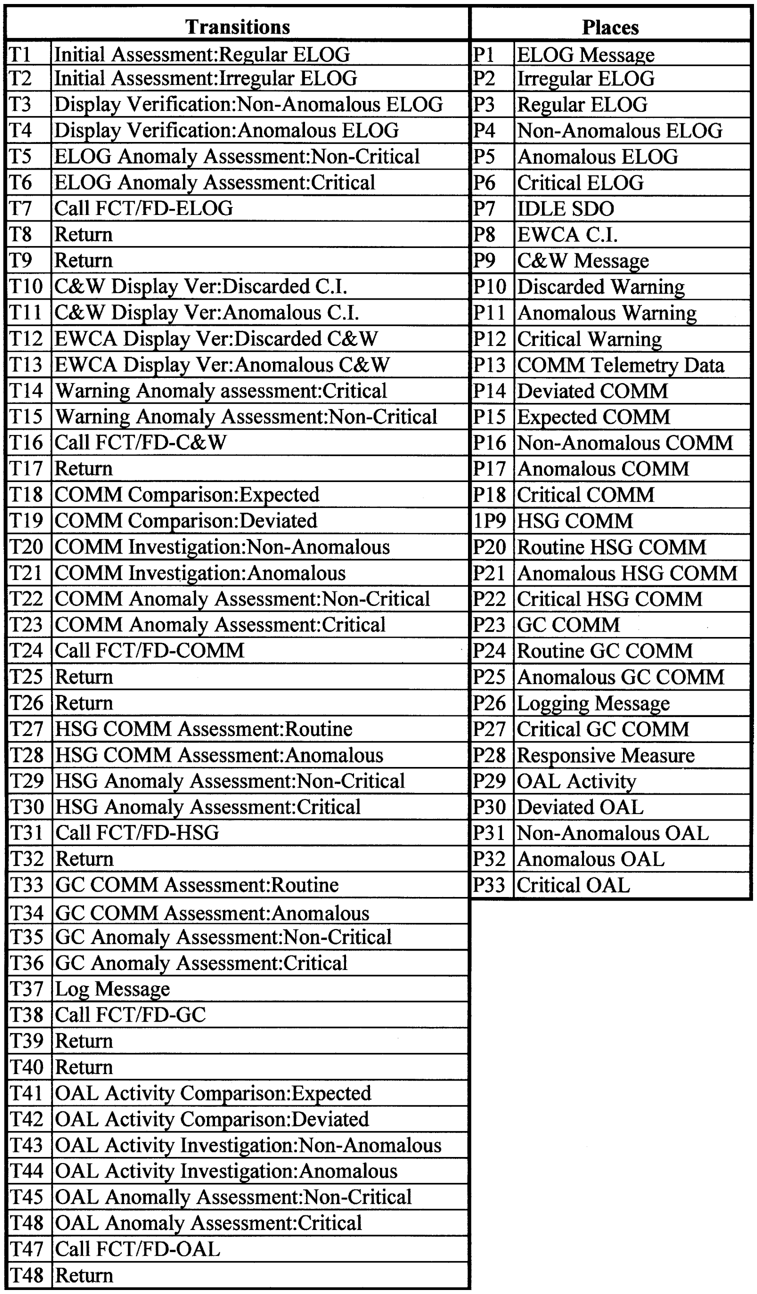


Fig. 9. SDO overall petri net.

missions will increase from two weeks in the shuttle, to three months on the space station, and then to three years for a future Mars exploration mission. Long distance operations will affect the capability of the FCT to react to events in real-time, due to the inherent delays in the information coming back to MCC from the spacecraft up to 20 min each way for a vehicle on the surface of Mars. In addition, mission operations are continuously being challenged to improve its processes to accomplish these missions at higher levels of safety, mission success, and effectiveness.

Automation is considered as an enabling technology to meet the aforementioned challenges. The synergistic combination of human flight controllers and intelligent software providing the function of OA is envisioned as the key implementation of this technology in the MCC. This technology is being developed in current programs to achieve increased operational efficiencies and improve safety. For future programs of exploration, this technology will be a requirement for achieving mission success. The long distances and extended mission lengths involved in exploration missions require that high levels of automation be functional on the vehicle. This effectively requires moving a significant amount of the functionality of the automated control center to onboard the exploration vehicles. OA can assist the flight controllers in their task to monitor the status and health of the spacecraft's systems.



| Transitions | | Places | |
|---|---|---|---|
| T1 | Initial Assessment:Regular ELOG | P1 | ELOG Message |
| T2 | Initial Assessment:Irregular ELOG | P2 | Irregular ELOG |
| T3 | Display Verification:Non-Anomalous ELOG | P3 | Regular ELOG |
| T4 | Display Verification:Anomalous ELOG | P4 | Non-Anomalous ELOG |
| T5 | ELOG Anomaly Assessment:Non-Critical | P5 | Anomalous ELOG |
| T6 | ELOG Anomaly Assessment:Critical | P6 | Critical ELOG |
| T7 | Call FCT/FD-ELOG | P7 | IDLE SDO |
| T8 | Return | P8 | EWCA C.I. |
| T9 | Return | P9 | C&W Message |
| T10 | C&W Display Ver:Discarded C.I. | P10 | Discarded Warning |
| T11 | C&W Display Ver:Anomalous C.I. | P11 | Anomalous Warning |
| T12 | EWCA Display Ver:Discarded C&W | P12 | Critical Warning |
| T13 | EWCA Display Ver:Anomalous C&W | P13 | COMM Telemetry Data |
| T14 | Warning Anomaly assessment:Critical | P14 | Deviated COMM |
| T15 | Warning Anomaly Assessment:Non-Critical | P15 | Expected COMM |
| T16 | Call FCT/FD-C&W | P16 | Non-Anomalous COMM |
| T17 | Return | P17 | Anomalous COMM |
| T18 | COMM Comparison:Expected | P18 | Critical COMM |
| T19 | COMM Comparison:Deviated | 1P9 | HSG COMM |
| T20 | COMM Investigation:Non-Anomalous | P20 | Routine HSG COMM |
| T21 | COMM Investigation:Anomalous | P21 | Anomalous HSG COMM |
| T22 | COMM Anomaly Assessment:Non-Critical | P22 | Critical HSG COMM |
| T23 | COMM Anomaly Assessment:Critical | P23 | GC COMM |
| T24 | Call FCT/FD-COMM | P24 | Routine GC COMM |
| T25 | Return | P25 | Anomalous GC COMM |
| T26 | Return | P26 | Logging Message |
| T27 | HSG COMM Assessment:Routine | P27 | Critical GC COMM |
| T28 | HSG COMM Assessment:Anomalous | P28 | Responsive Measure |
| T29 | HSG Anomaly Assessment:Non-Critical | P29 | OAL Activity |
| T30 | HSG Anomaly Assessment:Critical | P30 | Deviated OAL |
| T31 | Call FCT/FD-HSG | P31 | Non-Anomalous OAL |
| T32 | Return | P32 | Anomalous OAL |
| T33 | GC COMM Assessment:Routine | P33 | Critical OAL |
| T34 | GC COMM Assessment:Anomalous | | |
| T35 | GC Anomaly Assessment:Non-Critical | | |
| T36 | GC Anomaly Assessment:Critical | | |
| T37 | Log Message | | |
| T38 | Call FCT/FD-GC | | |
| T39 | Return | | |
| T40 | Return | | |
| T41 | OAL Activity Comparison:Expected | | |
| T42 | OAL Activity Comparison:Deviated | | |
| T43 | OAL Activity Investigation:Non-Anomalous | | |
| T44 | OAL Activity Investigation:Anomalous | | |
| T45 | OAL Anomally Assessment:Non-Critical | | |
| T48 | OAL Anomaly Assessment:Critical | | |
| T47 | Call FCT/FD-OAL | | |
| T48 | Return | | |

Fig. 10. SDO overall petri net data dictionary.

They also help maintain the flight controller's awareness of the operations being performed during the mission and help assure that operational objectives are met.

This study presents a unique two-stage specification methodology used to develop an OA for the SDO. The SDO DFPN model described in the study is a unique specification effort that has not been adopted in previous automation efforts at JSC. The proposed methodology has assisted systems analysts, systems designers, and software engineers at JSC to better understand the automation process and validate the data and control flow specifications needed in systems development (elimination of the black box syndrome). This methodology has provided users and development teams with a more complete and efficient set of systems specification blueprints.

There is a pressing need at JSC for system modeling techniques to support reliable, maintainable, and extensible design. This is especially critical when modeling systems with concurrent, distributed, parallel, event-driven, asynchronous, reactive, and non-deterministic qualities. DFDs, commonly used in systems analysis and design, do not represent control flow and are not well-suited for detailed design. We integrate PNs with DFDs to overcome this deficiency. This integration allows DFDs to be used for structured specifications in systems analysis and PNs for verification in systems design. PNs help analyze critical

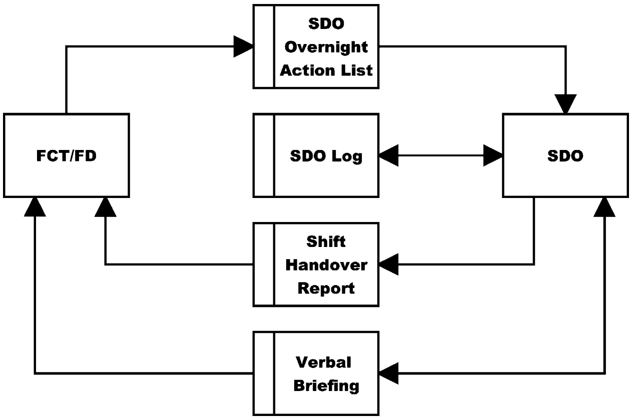


Fig. 11. Handover activities.

control flow properties and compare the performance of alternate system designs. The movement of tokens through the PN illustrates how PNs model system behavior. Marked PNs can be analyzed for properties that are important to specific designs, e.g. whether a given design can be implemented with limited capability hardware (flip–flops), whether a given sequence of transition firings is possible, or whether a design can become deadlocked. It is also possible to analyze performance characteristics and compare those characteristics for alternative system designs. Systems analysts at JSC use this methodology to analyze and design new systems more accurately and expeditiously, to reduce development efforts. It has enabled JSC analysts, without specialized PN training, to apply the power and precision of PNs for user verification in system specification.

Another area of concern at JSC is the difficulty of communicating the products of systems analysis and design for complex modeling problems in the field of space flight missions. Communication is required so that the systems analyst can gain a clear understanding of the system requirements from the expert users. This clear understanding is critical for creating a well-designed and accurate system. Effective communication among the members of the development team (systems analysts, systems designers, software developers, end users, and subject matter experts) is an essential component for good design. Using the proposed methodology at JSC would enhance communication between systems analysts and expert users, ensuring more accurate specifications. This methodology also improves the communication between systems analysts and software developers by producing control flow specifications for complex systems. Using this methodology, systems analysts, software designers, and subject matter experts at JSC can validate that the system design accurately reflects the intended system behavior. The specifications provided are used to communicate the system design and behavior to the software designers in an unambiguous fashion.

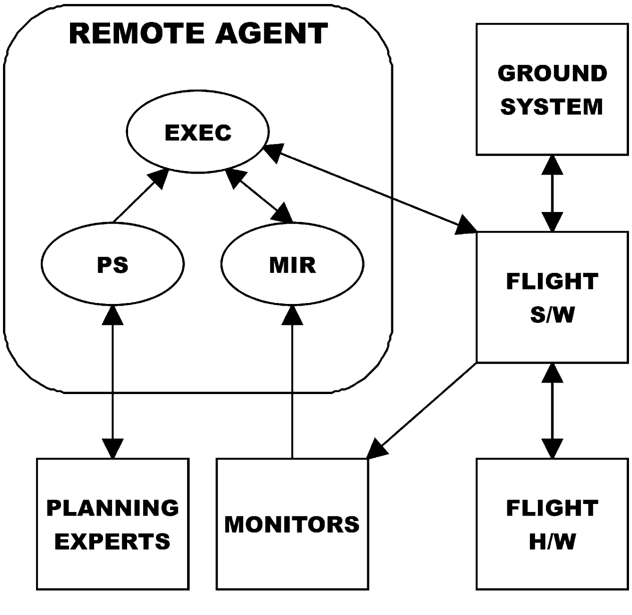


Fig. 12. Ames research center's remote agent.

Significant progress has been made in automation technology over the last few years. The rapid development of intelligent control (executive controllers, intelligent planners, etc.) and advanced computation techniques, such as intelligent-agent based systems, has produced several impressive implementations of autonomous control systems for robotics spacecraft, satellites, and ground-based life support systems. In spite of these successes, this technology remains to be successfully applied to human space operations. Lessons learned from previous efforts to develop and deploy automation systems for the control center point to several areas of open work and further study. In the area of architectures for automated systems, several architectures exist within NASA and the Department of Defense. Although these architectures have been successfully implemented in several specific applications, they should be evaluated more comprehensively. A significant difference of our application is that the to-be-developed OA can interact extensively with human operators and have to perform as members of the control team, making the OA a critical component of the operation. Another area of open work is to develop this technology in a form that is generically available for re-use by non-computer experts. The goal is to have a series of well-documented components and tools that allow non-computer experts to assemble these components into applications enabling them to solve a variety of complex operational automation problems. An important enhancement to current systems analysis capabilities is the development of tools to manage the knowledge bases required for operations (flight rules, procedures, etc.). Tools are also needed to accurately capture the flight controllers' operational knowledge while minimizing the burden on their time. This requires the development of dynamic modeling techniques that support design and validation of complex space flight systems.

## Acknowledgements

The authors are grateful to the editor, the anonymous reviewers, and Professor Anne Marie Smith for their insightful comments and suggestions. This research is supported by NASA grants NAG9-931 and NAG9-1196. Software Analysis Tool Integrating Data Flow Diagrams and Petri Nets, is a registered invention (MSC-23242-1-SB) of the primary author, Dr Tavana, and is supported by NASA contract number NAS9-00031.

## Appendix A. DFD to PN transition specification for the check-cashing problem

Table A1

Table A1

```
//Process 1.0: “Validate Card”
// inputs: “Check Cashing Card”, “Purchase Amount”
// outputs: “Valid Check Cashing Card”, “Invalid Check Cashing Card”
// description: (corresponds to PN T1 and T2)
//   Examine “Check Cashing Card” and its expiration date to verify
//   if it is valid or invalid. Output as invalid or valid, as appropriate.
T1. “Validate Card: Valid”
  signal: Valid Card                                      //signal is needed, outputs > 1
  inputs: P1: “Initial State”                             //inputs all from outside
    P2: “Check Cashing Card”                              //from P1.0
    P3: “Purchase Amount”                                 //from P1.0
  outputs: P4: “Valid Check Cashing Card”                 //from P1.0
T2. “Validate Card: Invalid”
  signal: Invalid Card                                    //signal is needed, outputs > 1
  inputs: P1: “Initial State”                             //inputs all from outside
    P2: “Check Cashing Card”                              //from P1.0
    P3: “Purchase Amount”                                 //from P1.0
  outputs: P6: “Invalid Check Cashing Card”               //from P1.0
//Process 2.0: “Accept Check for Purchase Plus $25”
// inputs: “Valid Check Cashing Card”
// outputs: “Processed Check for Purchase Plus $25”
// description: (corresponds to PN T3)
//   Process the check and complete the sales transaction, allowing the
//   check to be cashed for purchase amount plus $25.
T3. “Accept Check for Purchase Plus $25”
  inputs: P4: “Valid Check Cashing Card”
  outputs: P5: “Processed Check for Purchase Plus $25”
T14. “Return”                                             //destination for data flow which goes outside the diagram
  inputs: P5: “Processed Check for Purchase Plus $25”
  outputs: P1: “Initial State”                            //destination for all returns
//Process 3.0: “Measure Purchase Amount”
// inputs: “Invalid Check Cashing Card”
// outputs: “Purchase Less Than $20”, “Purchase $20 and More”
// description: (corresponds to PN T4 and T5)
//   If the purchase amount is less than $20.
//     output as “Purchase Less Than $20”;
//   otherwise
//     output as “Purchase $20 and More”
T4. “Measure Purchase Amount: Less Than $20”
  signal: amount less than $20                            //signal because DFD > 2 outputs
  inputs: P6: “Invalid Check Cashing Card”
  outputs: P7: “Purchase Less Than $20”
T5. “Measure Purchase Amount: $20 and More”
  signal: amount $20 and more                             //signal because DFD > 1 output
  inputs: P6: “Invalid Check Cashing Card”
  outputs: P13: “Purchase $20 and More”
//Process 4.0: “Assess IDs”
// inputs: “Purchase Less Than $20”
// outputs: “Two IDs”, “Less Than Two Valid IDs”
// description: (corresponds to PN T6 and T8)
//   If the number of valid IDs is less than 2
//     output as "Less Than Two Valid IDs"
//   otherwise
//     output as "Two IDs"
T6. "Assess IDs: Less Than Two Valid IDs"
  signal: Less than Two Valid Ids                                   //signal because DFD > 1 output
  inputs: P7: "Purchase Less Than $20"
  outputs: P10: "Less Than Two Valid IDs"
T8. "Assess IDs: Two IDs"
  signal: Two Ids                                                   //signal because DFD > 1 output
  inputs: P7: "Purchase Less Than $20"
  outputs: P8: "Two IDs"
//Process 5.0: "Accept Check for Purchase Amount"
// inputs: "Two IDs"
// outputs: "Processed Check for Purchase Amount"
// description: (corresponds to PN T7)
//   Process the check and complete the sales transaction, allowing the
//   check to be cashed for the exact purchase amount.
T7. "Accept Check for Purchase Amount"
  inputs: P8: "Two IDs"
  outputs: P9: "Processed Check for Purchase Amount"
T13. "Return"                                                       //destination for data flow which goes outside the diagram
  inputs: P9: "Processed Check for Purchase Amount"
  outputs: P1: "Initial State"                                      //destination for all returns
//Process 6.0: "Manager Accept High Amount"
// inputs: "Purchase $20 and More"
// outputs: "Approved Check in Amount Because of Amount",
//     "Rejected Check Because of Amount"
// description: (corresponds to PN T9 and T10)
//   If manager intuition is happy
//     output as "Approved Check in Amount Because of Amount"
//   otherwise
//     output as "Rejected Check Because of Amount"
T9. "Manager Accept High Amount: Approved Check Because of Amount"
  inputs: P13. "Purchase $20 and More"
  outputs: P11. "Approved Check in Amount Because of Amount",
T3. "Return"                                                        //destination for data flow which goes outside the diagram
  inputs: P11. "Approved Check in Amount Because of Amount",
  outputs: P1: "Initial State"                                      //destination for all returns
T10. "Manager Accept High Amount: Rejected Check Because of Amount"
  inputs: P13. "Purchase $20 and More"
  outputs: P12. "Rejected Check Because of Amount"
T4. "Return"                                                        //destination for data flow which goes outside the diagram
  inputs: P12. "Rejected Check Because of Amount"
  outputs: P1: "Initial State"                                      //destination for all returns
//Process 7.0: "Manager Accept Insufficient IDs"
// inputs: "Less Than Two Valid IDs"
// outputs: "Approved Check in Amount Because of IDs",
//     "Rejected Check Because of IDs"
// description: (corresponds to PN T11 and T12)
//   If manager intuition is happy
//     output as "Approved Check in Amount Because of IDs"
//   otherwise
// output as "Rejected Check Because of IDs"
T11. "Manager Accept Insufficient IDs: Approved Check in Amount
Because of IDs"
  inputs: P10. "Less Than Two Valid IDs"
  outputs: P14. "Approved Check in Amount Because of IDs"
T14. "Return"                                                       //destination for data flow which goes outside the diagram
  inputs: P14. "Approved Check in Amount Because of IDs"
  outputs: P1: "Initial State"                                      //destination for all returns
T12. "Manager Accept Insufficient IDs: Rejected Check Because of IDs"
  inputs: P10. "Less Than Two Valid IDs"
  outputs: P15. "Rejected Check Because of IDs"
T14. "Return"                                                       //destination for data flow which goes outside the diagram
  inputs: P15. "Rejected Check Because of IDs"
  outputs: P1: "Initial State"                                      //destination for all returns
```

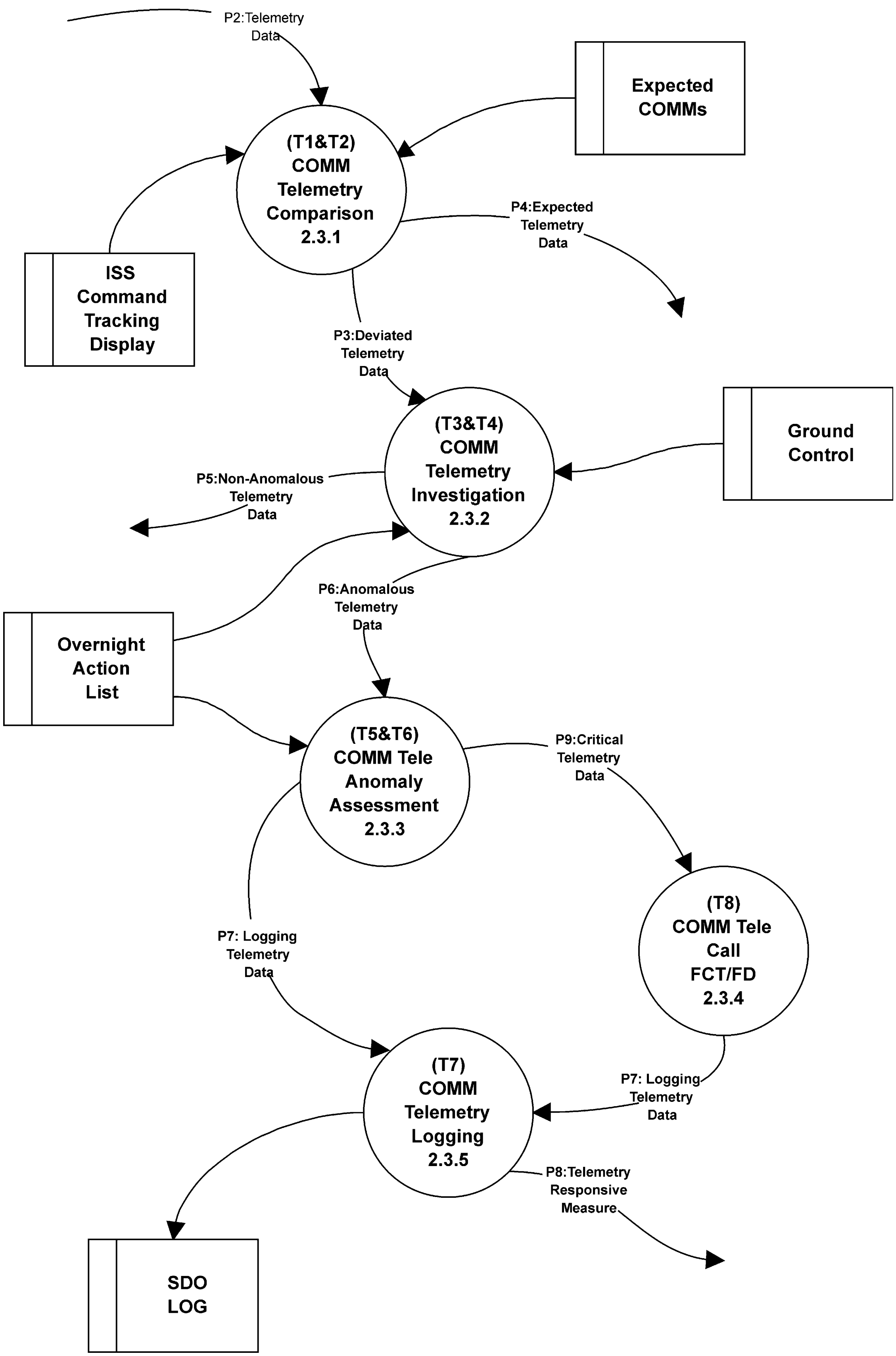

P2:Telemetry Data
Expected COMMs
(T1&T2) COMM Telemetry Comparison 2.3.1
ISS Command Tracking Display
P4:Expected Telemetry Data
P3:Deviated Telemetry Data
(T3&T4) COMM Telemetry Investigation 2.3.2
Ground Control
P5:Non-Anomalous Telemetry Data
P6:Anomalous Telemetry Data
Overnight Action List
(T5&T6) COMM Tele Anomaly Assessment 2.3.3
P9:Critical Telemetry Data
(T8) COMM Tele Call FCT/FD 2.3.4
P7: Logging Telemetry Data
(T7) COMM Telemetry Logging 2.3.5
P7: Logging Telemetry Data
P8:Telemetry Responsive Measure
SDO LOG

Fig. B1. COMM telemetry processing data flow diagram.

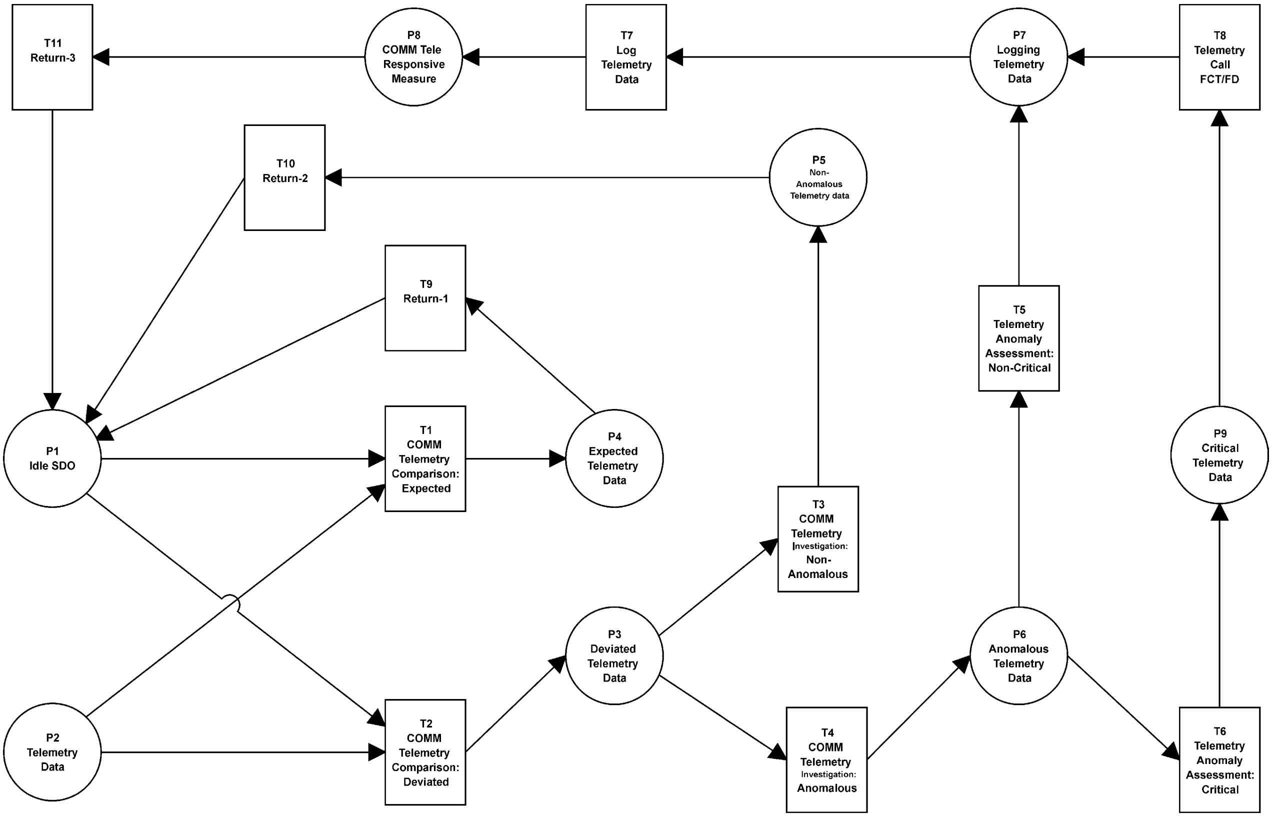


Fig. B2. COMM telemetry processing petri net.

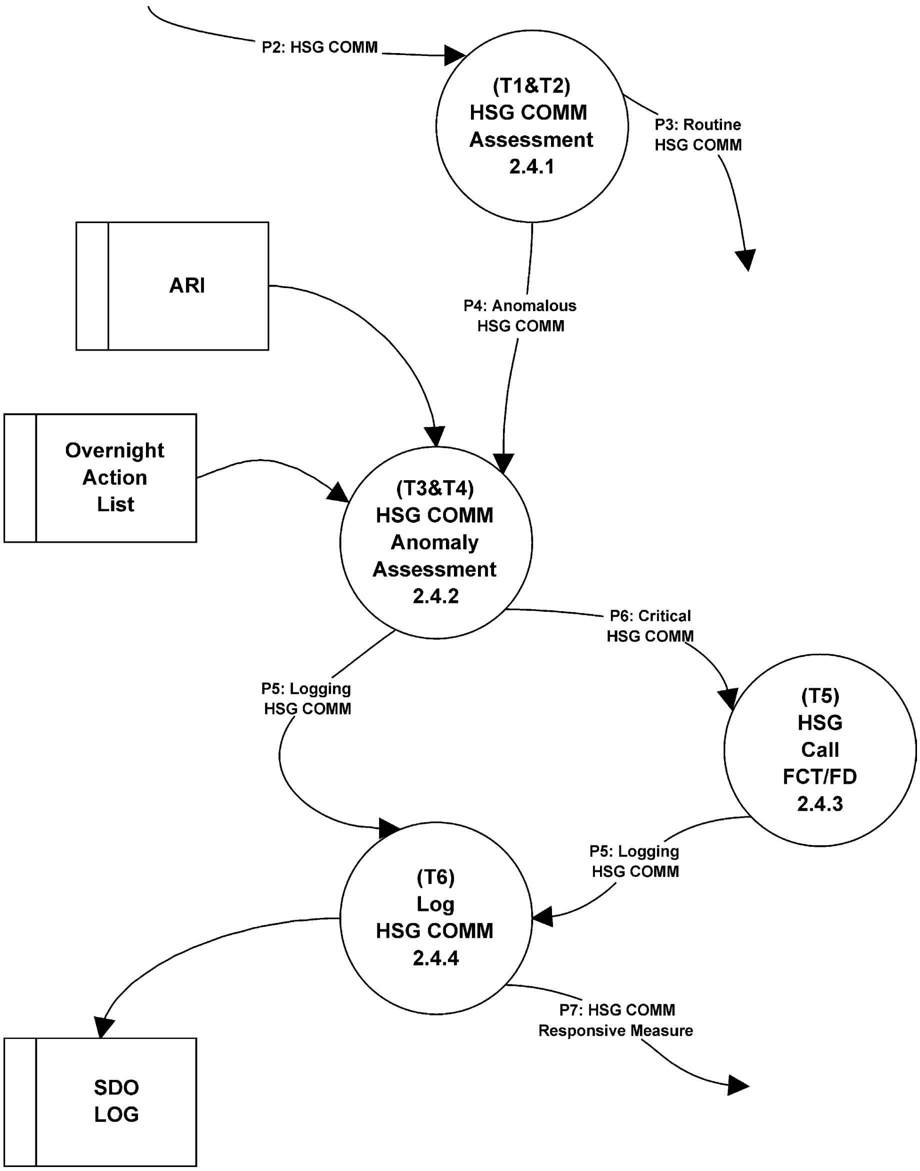


Fig. B3. Houston support group processing data flow diagram.

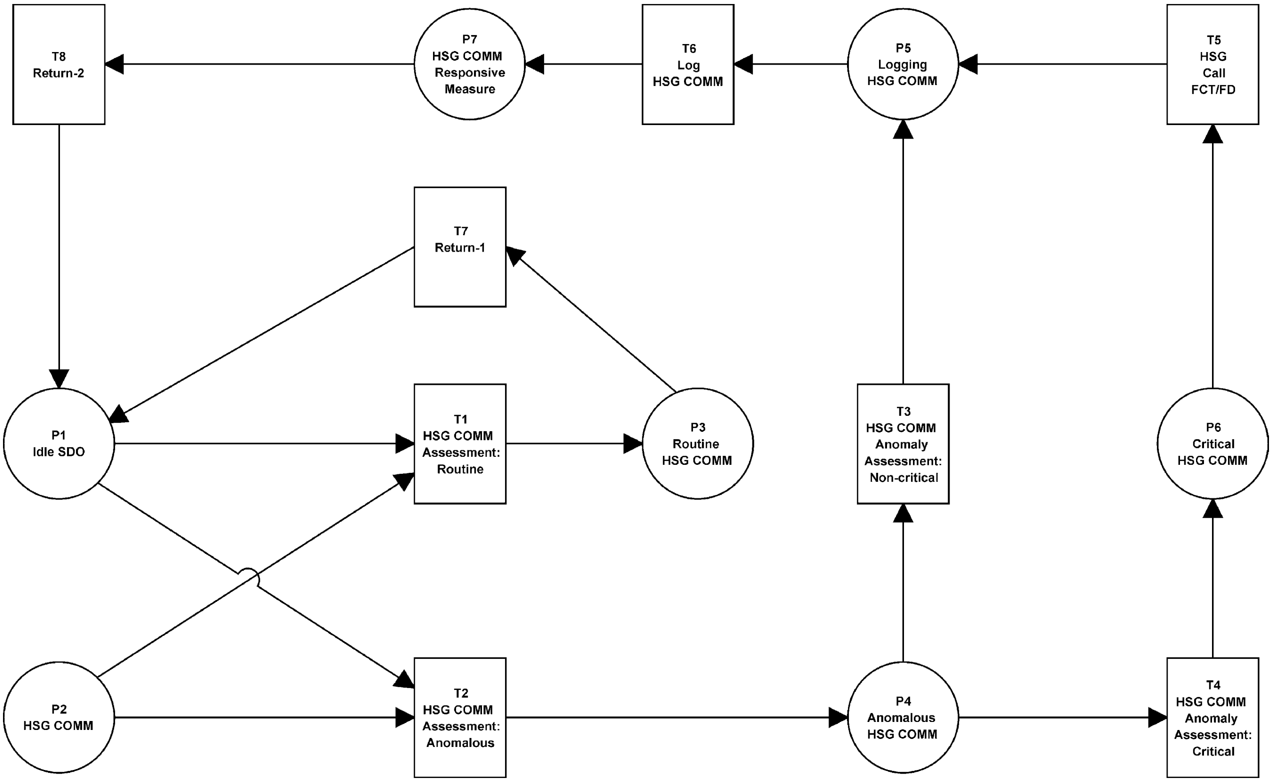


Fig. B4. Houston support group processing petri net.

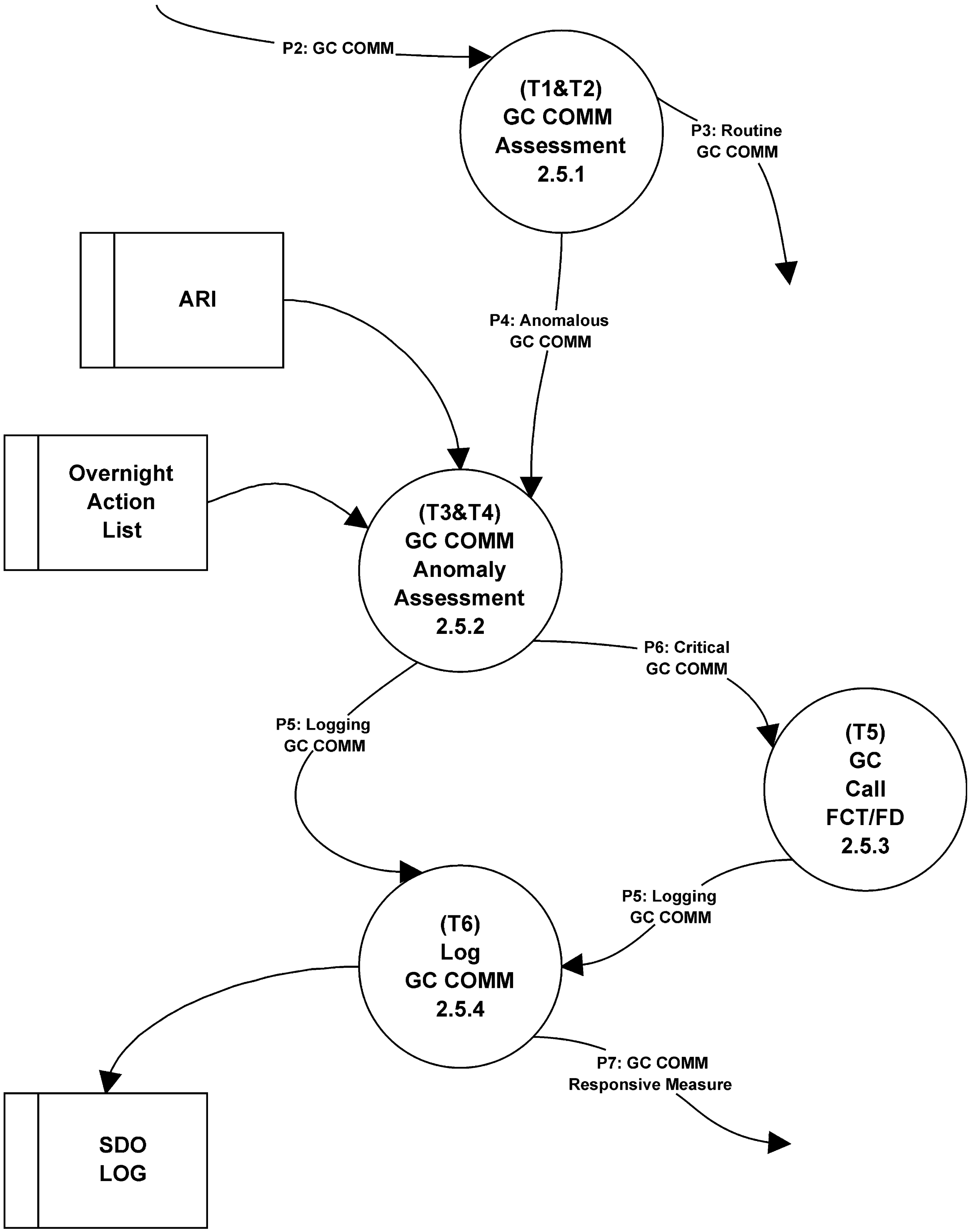


Fig. B5. Ground control communication processing data flow diagram.

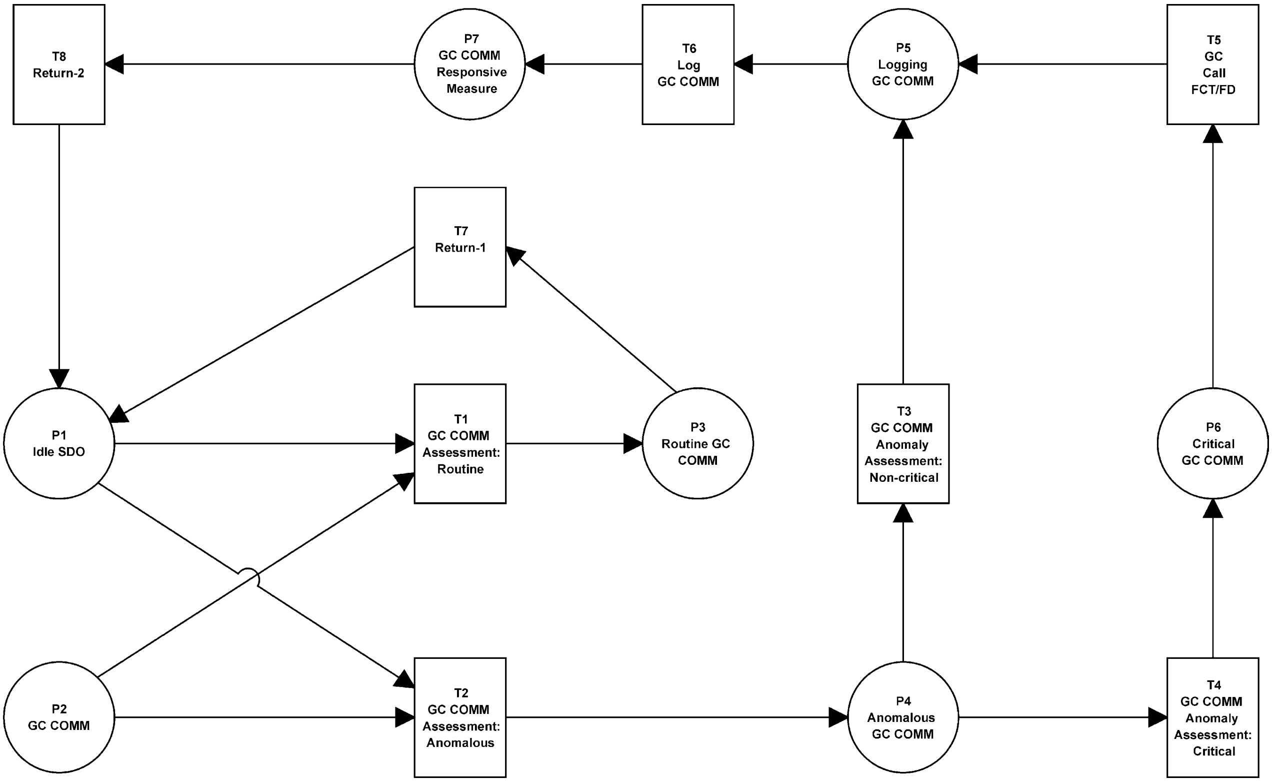


Fig. B6. Ground control communication processing petri net.

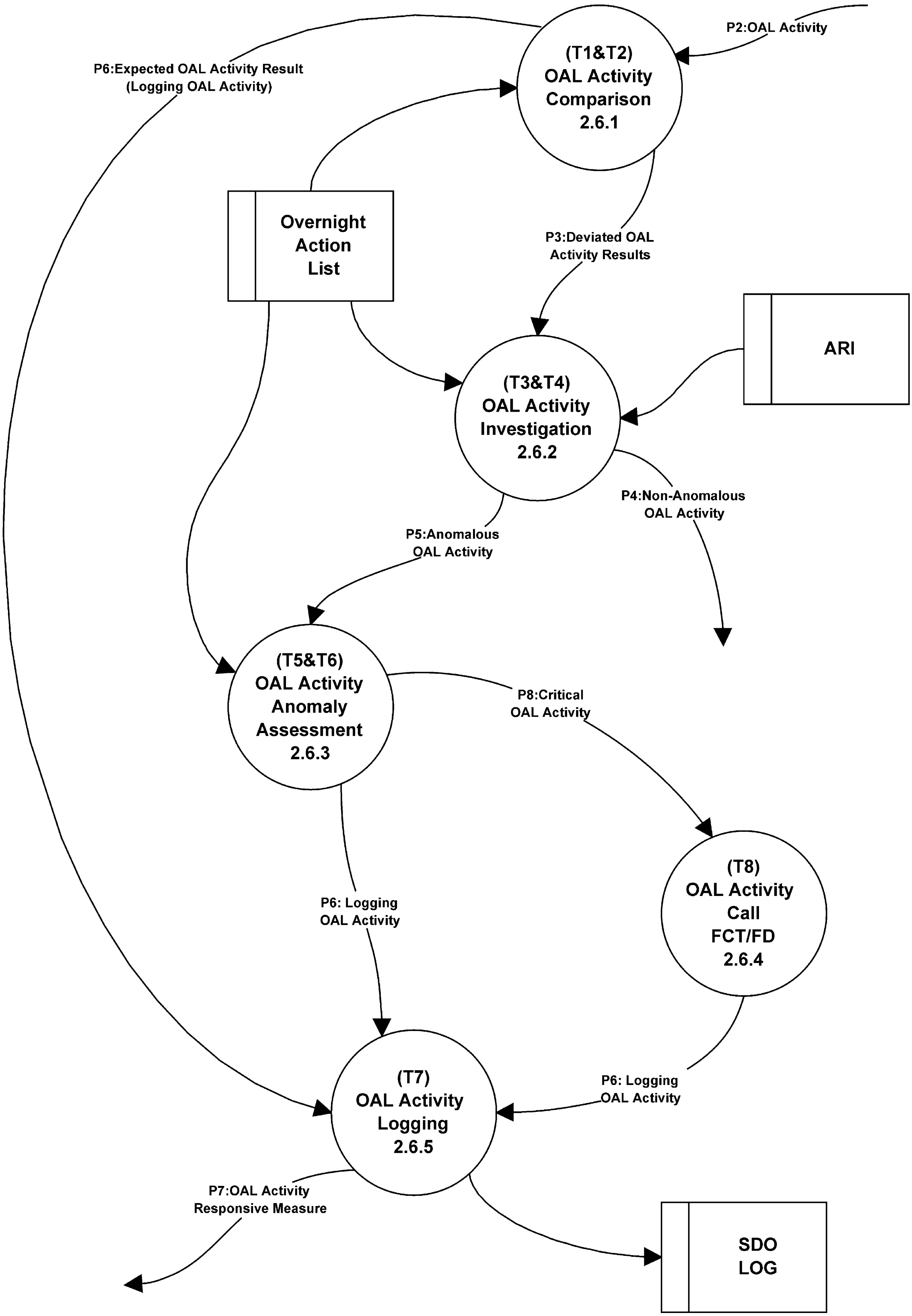


Fig. B7. OAL activity processing data flow diagram.

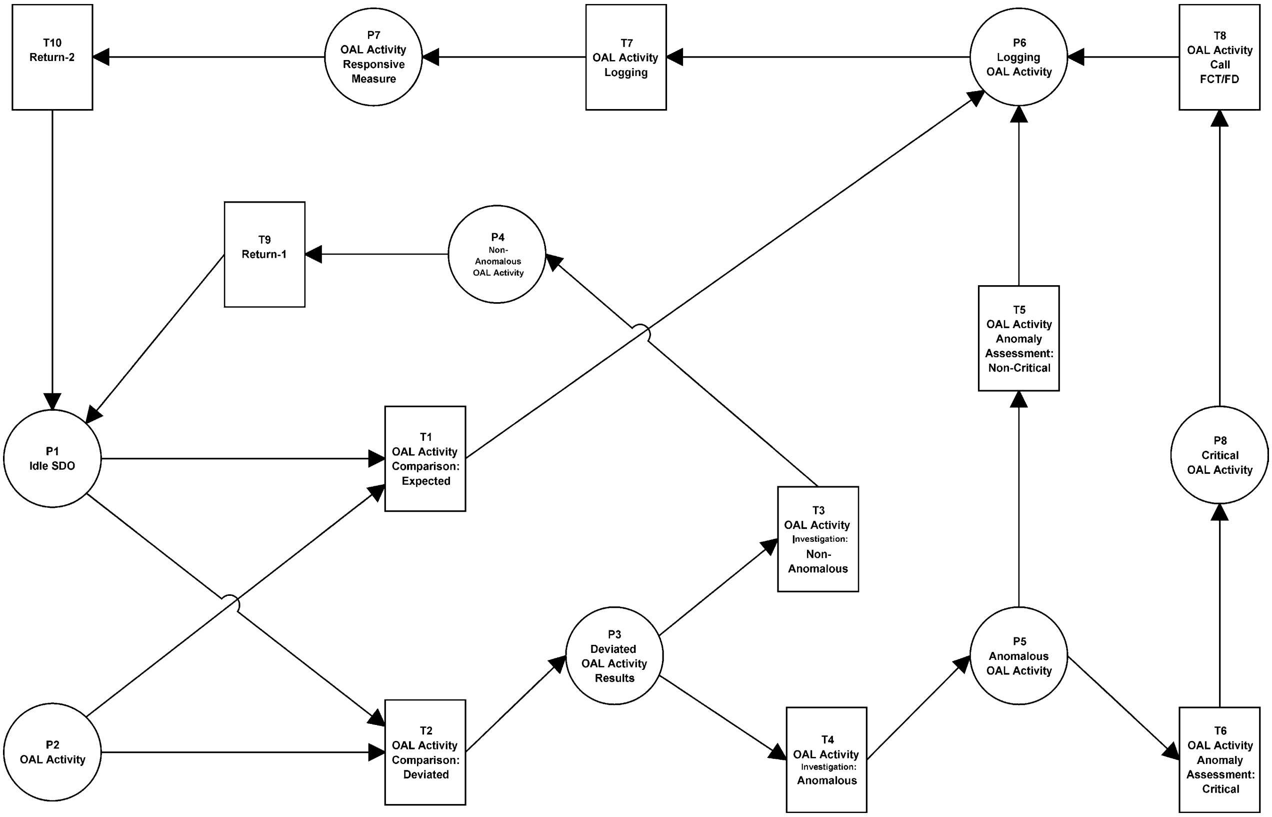


Fig. B8. OAL processing petri net.

## Appendix B

Figs. B1–B8.

## References

[1] Baylin EN. System diagramming methods: which works best? Inform Syst Mgmt 1987;4(3):29–40.

[2] Berthelot G, Terrat R. Petri net theory for correctness of protocols. IEEE Trans Commun 1982;30(12):2497–505.

[3] Bruno G, Marchetto G. Process-translatable petri nets for the rapid prototyping of process control systems. IEEE Trans Software Engng 1986;vSE-12:346–57.

[4] Bullers WI. A tripartite approach to information systems development. Decision Sci 1991;22(1):120–36.

[5] Camurri A, Franchi P, Gandolfor F, Zaccaria R. Petri net based process scheduling: a model of the control system of flexible manufacturing systems. J Intell Robotic Syst 1992;.

[6] DeMarco T. Structured analysis and system specification. Englewood Cliffs, NJ: Prentice-Hall; 1979.

[7] France RB. Semantically extended data flow diagrams: a formal specification tool. IEEE Trans Software Engng 1992;18(4): 329–47.

[8] Gene C, Sarson T. Structured systems analysis: tools and techniques. Englewood Cliffs, NJ: Prentice-Hall; 1980.

[9] Holt AW. Introduction of occurrence systems. In: JCKS EL, editor. Assoc Inform Tech. New York: Elsevier; 1971.

[10] Jantzen M. Structure representation of knowledge by petri nets as an aid for teaching and research. Lecture notes in computer science, New York: Springer; 1980.

[11] Kievit KA, Martin MP. Systems analysis tools: who's using them? J Syst Mgmt 1989;40(7):26–31.

[12] Millet I. Technical note: a proposal to simplify data flow diagrams. IBM Syst J 1999;38(1):118–21.

[13] Murata T. Modeling and analysis of concurrent systems. In: Vicks CR, editor. Handbook of software engineering. New York: Van Nostrand Reinhold; 1984. p. 39–63.

[14] Murata T. Petri nets: properties, analysis and applications. Proc IEEE 1989;77(4):541–580.

[15] Peterson JL. Petri net theory and the modeling of systems. Englewood Cliffs, NJ: Prentice-Hall; 1981.

[16] Petri C. Communicatiomn with automata. Supplement 1 to technical report RADC-TR-65-377, vol. 1, Rome Air Development Center, Griffith Air Force Base, Jan 1966. Translated by Greene Jr CF from 'Kommunikation mit Automaten', University of Bonn, Germany; 1962.

[17] Richter G, Maffeo B. Toward a rigorous interpretation of ESNL: extended systems modeling language. IEEE Trans Software Engng 1993;19(2):165–81.

[18] Ross DT. Structured analysis (SA): a language for communicating ideas. IEEE Trans Software Engng 1977;SE3(1):16–34.

[19] Tao Y, Kung C. Formal definition and verification of data flow diagrams. J Syst Software 1991;16(1):29–37.

[20] Van Hee KM, Somers LJAM, Voorhoeve M. A modeling environment for decision support systems. Decision Support Syst 1991;7(3): 241–52.

[21] Wang F, Kyriakopoulos KJ, Tsolkas T, Saridis GN. A petri-net coordination model for an intelligent mobile robot. IEEE Trans Syst Man Cybernetics 1991;21(4).

[22] Ward PT. The transformation schema: an extension of the data flow diagram to represent control and timing. IEEE Trans Software Engng 1986;SE12(2):198–211.

[23] Yourdon E. What ever happened to structured analysis? Datamation 1986;32(11):133–8.

[24] Zave P. An operational approach to requirements specification for embedded systems. IEEE Trans Software Engng 1982;8(3):250–69.

[25] Zurawski R, Zhou M-C. Petri nets and industrial applications: a tutorial. IEEE Trans Ind Electron 1994;41(6):567–83.