Прежде чем приступить к проектированию и моделированию взаимодействия компонентов, давайте определимся с основными элементами и их функциональностью:

1. БКУ (бортовой комплекс управления): отвечает за управление всеми компонентами космического аппарата.

2. БИУС-Л (блок измерения угловых скоростей): предназначен для определения ориентации и скорости полета космической станции.

3. Двигатель: отвечает за перемещение аппарата и переход на эллиптическую орбиту.

4. Другие модули: передают данные и взаимодействуют с остальными компонентами.

Теперь перейдем к проектированию взаимодействия компонентов.

1. Катастрофическая версия:

Для создания катастрофической ситуации в результате неправильного расположения команды рапорта значений скорости аппарата в пуле данных БИУС-Л, мы можем предположить следующую последовательность действий:

- БКУ отправляет команду "Включить акселерометры" БИУС-Л.

- БИУС-Л получает команду, но из-за перегрузки входящего потока данных, команда рапорта значений скорости аппарата оказывается неправильно расположенной в пуле данных БИУС-Л.

- БКУ ожидает получения данных о скорости от БИУС-Л, чтобы фиксировать вывод аппарата на эллиптическую орбиту.

- Однако, из-за неправильного расположения команды рапорта значений скорости аппарата, БКУ не получает нужные данные и не может правильно фиксировать вывод аппарата на эллиптическую орбиту.

- В результате, двигатель продолжает работать, а аппарат не переходит на эллиптическую орбиту. В конечном итоге, столкновение с лунной поверхностью становится неизбежным, что приводит к катастрофе.

**#define NUM\_COMMANDS 5**

**mtype = {CMD\_TURN\_ON\_ACCELEROMETERS, CMD\_REPORT\_VELOCITY, CMD\_START\_ENGINE, CMD\_STOP\_ENGINE, CMD\_REBOOT\_BIUS\_L};**

**chan commandChannel = [NUM\_COMMANDS] of { mtype, int };**

**chan dataChannel = [NUM\_COMMANDS] of { mtype, int };**

**int BIUS\_L\_dataQueue[NUM\_COMMANDS];**

**int BKU\_commandQueue[NUM\_COMMANDS];**

**int engineRunning = 0;**

**active proctype BKU() {**

**int index = 0;**

**// Send commands to BIUS-L**

**commandChannel!CMD\_TURN\_ON\_ACCELEROMETERS, index;**

**commandChannel!CMD\_REPORT\_VELOCITY, index;**

**// Wait for data from BIUS-L**

**dataChannel?BIUS\_L\_dataQueue[index];**

**// Process data to control engine**

**if**

**:: BIUS\_L\_dataQueue[index] == 1 -> // Simulate misplaced velocity reporting command**

**commandChannel!CMD\_START\_ENGINE, index; // Incorrectly assuming we have received velocity data**

**:: else -> skip;**

**fi;**

**// Control engine and further commands**

**commandChannel!CMD\_STOP\_ENGINE, index;**

**commandChannel!CMD\_REBOOT\_BIUS\_L, index;**

**}**

**active proctype BIUS\_L() {**

**int index = 0;**

**// Simulate data processing**

**commandChannel?BIUS\_L\_dataQueue[index];**

**if**

**:: BIUS\_L\_dataQueue[index] == 0 -> // Simulate misplaced velocity reporting command**

**dataChannel!0, index; // Send incorrect data to BKU**

**:: else -> skip;**

**fi;**

**// Receive other commands**

**commandChannel?BIUS\_L\_dataQueue[index];**

**dataChannel!BIUS\_L\_dataQueue[index];**

**}**

**active proctype Engine() {**

**int index;**

**// Control engine based on BKU's commands**

**commandChannel?engineRunning, index;**

**if**

**:: engineRunning == 1 -> skip;**

**:: else -> skip;**

**fi;**

**}**

**init {**

**atomic {**

**// Start the processes**

**run BKU();**

**run BIUS\_L();**

**run Engine();**

**}**

**}**

2. Измененная версия для избежания катастрофы:

Чтобы избежать катастрофы, мы можем внести следующие изменения в проект:

- БКУ будет отправлять команду "Включить акселерометры" БИУС-Л только после того, как получит подтверждение от остальных модулей о готовности к работе.

- БИУС-Л будет проверять полученные команды и правильно располагать их в пуле данных перед отправкой.

- БКУ будет ожидать получения данных о скорости от БИУС-Л и только после этого фиксировать вывод аппарата на эллиптическую орбиту.

- При необходимости перезагрузки БИУС-Л, БКУ будет отправлять команду "Перезагрузить", но только после того, как убедится, что остальные модули не передают важные данные.

**#define NUM\_COMMANDS 5**

**mtype = {CMD\_TURN\_ON\_ACCELEROMETERS, CMD\_REPORT\_VELOCITY, CMD\_START\_ENGINE, CMD\_STOP\_ENGINE, CMD\_REBOOT\_BIUS\_L, CMD\_READY\_TO\_WORK};**

**chan commandChannel = [NUM\_COMMANDS] of { mtype, int };**

**chan dataChannel = [NUM\_COMMANDS] of { mtype, int };**

**int BIUS\_L\_dataQueue[NUM\_COMMANDS];**

**int BKU\_commandQueue[NUM\_COMMANDS];**

**int engineRunning = 0;**

**active proctype BKU() {**

**int index = 0;**

**// Send commands to other modules to confirm readiness**

**commandChannel!CMD\_READY\_TO\_WORK, index;**

**commandChannel!CMD\_READY\_TO\_WORK, index;**

**// Wait for confirmation from other modules**

**commandChannel?CMD\_READY\_TO\_WORK, index;**

**commandChannel?CMD\_READY\_TO\_WORK, index;**

**// Send commands to BIUS-L**

**commandChannel!CMD\_TURN\_ON\_ACCELEROMETERS, index;**

**// Wait for confirmation from BIUS-L**

**commandChannel?CMD\_READY\_TO\_WORK, index;**

**// Wait for data from BIUS-L**

**dataChannel?BIUS\_L\_dataQueue[index];**

**// Process data to control engine**

**if**

**:: BIUS\_L\_dataQueue[index] == 1 -> // Simulate correct velocity reporting command**

**commandChannel!CMD\_START\_ENGINE, index; // Correctly assuming we have received velocity data**

**:: else -> skip;**

**fi;**

**// Control engine and further commands**

**commandChannel!CMD\_STOP\_ENGINE, index;**

**commandChannel!CMD\_REBOOT\_BIUS\_L, index;**

**}**

**active proctype BIUS\_L() {**

**int index = 0;**

**// Confirm readiness to work**

**commandChannel?CMD\_READY\_TO\_WORK, index;**

**commandChannel?CMD\_READY\_TO\_WORK, index;**

**// Process and correctly place commands**

**commandChannel?BIUS\_L\_dataQueue[index];**

**if**

**:: BIUS\_L\_dataQueue[index] == 0 -> // Simulate correctly placed velocity reporting command**

**dataChannel!1, index; // Send correct data to BKU**

**:: else -> skip;**

**fi;**

**// Confirm readiness to work**

**commandChannel!CMD\_READY\_TO\_WORK, index;**

**// Receive other commands**

**commandChannel?BIUS\_L\_dataQueue[index];**

**dataChannel!BIUS\_L\_dataQueue[index];**

**}**

**active proctype Engine() {**

**int index;**

**// Control engine based on BKU's commands**

**commandChannel?engineRunning, index;**

**if**

**:: engineRunning == 1 -> skip;**

**:: else -> skip;**

**fi;**

**}**

**init {**

**atomic {**

**// Start the processes**

**run BKU();**

**run BIUS\_L();**

**run Engine();**

**}**

**}**

Таким образом, внесенные изменения в проект позволяют избежать катастрофы, так как правильное взаимодействие компонентов и правильная обработка команд гарантируют корректное выполнение миссии и переход на эллиптическую орбиту без столкновения с лунной поверхностью.