Global Variables Arguments + TransactionItem: QueueItem + TransactionData: DataTable + TransactionNumber: Int32 = 1 + RetryNumber: Int32 + Config: Dictionary<string, object> + in_wbType: String = "MainTask" + SystemError: Exception + in_wbLevel: Int32 = 0 + BusinessRuleException: BusinessRuleException + in_wbParent: String = string.Empty + wbMain_Type: String = in_wbType + wbMain_Level: Int32 = in_wbLevel + wbMain_Parent: String = in_wbParent + wbMain_Audit: Dictionarry<string, object> = New Dictionary(Of String, Object) from { {"Successful", 0}, {"Failed", 0} } + wbMain_StartTime: DateTime = now() SystemError BusinessRuleException Success <<State Machine - Initialization State>> <>State Machine - Get Transaction Data State>> <<Framework Layer>> <<Framework Layer>> wblnitialisation wbGetTransactionData Local variables: Local variables: + wbGetTransactionData_Type: String = in_wbType+", "+"Get Transaction Data State" + wblnit_Type: String = in_wbType+", "+"Initialization State" + wblnit_Level: Int32 = wbMain_Level+1 + wbGetTransactionData_Level: Int32 = wbMain_Level+1 + wblnit_Parent: String = wbMain_Type + wbGetTransactionData_Parent: String = wbMain_Type + wbGetTransactionData_Audit: Dictionarry<string, object> = New Dictionary(Of String, Object) + wblnit_Audit: Dictionarry<string, object> = New Dictionary(Of String, Object) from { {"Successful", 0}, {"Failed", 0} } from { {"Successful", 0}, {"Failed", 0} } + wblnit_StartTime: DateTime = now() + wbGetTransactionData_StartTime: DateTime = now() + ShouldStop: Boolean Try executing Components: Try executing Components: **System - reserved** 包 System - reserved + InitAllSettings (in_ConfigFile:"Data\Config.xlsx", in_ConfigSheets:{"Settings", "Constants"}, out_Config:Config) + OrchestratorStop () **System - reserved Business Process Layer Interface** << Business Process Layer>> << Business Process Layer>> + CloseAllApplications (in_wbType:in_wbType+", "+"Close All + GetTransactionData (in_TransactionNumber:TransactionNumber, in_Config:Config, Applications", in_wbParent:wbInit_Type, in_wbLevel:wbInit_Level+1, out_TransactionItem:TransactionItem, io_TransactionData:TransactionData) io_Audit:wblnit_Audit) 包 Log success **Business Process Layer Interface** << Business Process Layer>> + wbLogging(in_wbType:wbGetTransactionData_Type, in_wbLevel:wbGetTransactionData_Level, in_wbParent:wbGetTransactionData_Parent, + InitAllApplications (in_Config:Config, in_wbType: in_wbType+", in_LogLevel:1, in_LogMessage:"Work Block "+ wbGetTransactionData_Type.toString + " "+"InitAllApplications", in_wbParent:wblnit_Type, in_wbLevel:wblnit_Level+1, successful", in wbStart:wbGetTransactionData StartTime, io_Audit:wblnit_Audit) in_ChildAudit:wbGetTransactionData_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:true, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false)) Log success + wbLogging(in_wbType:wblnit_Type, in_wbLevel:wblnit_Level, Catch all exceptions, log them, treat or rethrow in_wbParent:wbInit_Parent, in_LogLevel:1, in_LogMessage:"Work Block "+ wblnit_Type.toString + " successful", in_wbStart:wblnit_StartTime, 包 in_ChildAudit:wblnit_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:true. **System.Exception: Log failure** in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false)) + wbLogging(in_wbType:wbGetTransactionData_Type, in_wbLevel:wbGetTransactionData_Level, in_wbParent:wbGetTransactionData_Parent, in_LogLevel:4, in_LogMessage:"Work Block "+ wbGetTransactionData_Type.toString + ' Catch all exceptions, log them, treat or rethrow failed with exception "+exception.Message+" at Source "+exception.Source, in_wbStart:wbGetTransactionData_StartTime, in ChildAudit:wbGetTransactionData Audit, io ParentAudit:wbMain Audit, in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError is Nothing AND System.Exception: Log failure BusinessRuleException is Nothing, true, false)) + wbLogging(in_wbType:wblnit_Type, in_wbLevel:wblnit_Level, in_wbParent:wbInit_Parent, in_LogLevel:4, in_LogMessage:"Work Block "+ wblnit_Type.toString + " failed with exception "+exception.Message+" at Source "+exception.Source, in_wbStart:wbInit_StartTime, in_ChildAudit:wbInit_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false)) Success <<State Machine - Process Transaction State>> << Framework Layer>> **wbProcess** Local variables: + wbProcessTransaction_Type: String + wbProcessTransaction_Level: Int32 + wbProcessTransaction_Parent: String + wbProcessTransaction_Audit: Dictionarry<string, object> + wbProcessTransaction_StartTime: DateTime Try executing Components: System - reserved <<State Machine - End Process State>> 起 <<Framework Layer>> **Business Process Layer Interface** wbMain <<Business Process Layer>> Local variables: + **Process** (in_TransactionItem:TransactionItem, in_Config:Config, in_TransactionNumber:TransactionNumber, in_RetryNumber:RetryNumber, Try executing Components: in_wbType:in_wbType+", "+"Process Transaction", in wbLevel:wbProcessTransaction Level+1, in wbParent:wbProcessTransaction Type, 起 io_Audit:wbProcessTransaction_Audit) System - reserved 起 Log success **Business Process Layer Interface** + **wbLogging**(in_wbType:wbProcessTransaction_Type, << Business Process Layer>> in_wbLevel:wbProcessTransaction_Level, in_wbParent:wbProcessTransaction_Parent, in_LogLevel:1, in_LogMessage:"Work Block "+ wbProcessTransaction_Type.toString + " successful", in_wbStart:wbProcessTransaction_StartTime, + CloseAllApplications(in_wbType:in_wbType+", "+"Close All Applications", in_wbParent:wbMain_Type, in_wbLevel:wbMain_Level+1, in_ChildAudit:wbProcessTransaction_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:true, in_wbStatusSuccessful:if(SystemError is Nothing AND io_Audit:wbMain_Audit) BusinessRuleException is Nothing, true, false)) Log success Catch all exceptions, log them, treat or rethrow + wbLogging(in_wbType:wbMain_Type, in_wbLevel:wbMain_Level, in_wbParent:wbMain_Parent, in_LogLevel:1, in_LogMessage:"Work Block "+ System. Exception: Log failure wbMain_Type.toString + " successful", in_wbStart:wbMain_StartTime, _ChildAudit:wbMain_Audit, io_ParentAudit:Nothing, in_SuppressLogging + wbLogging(in_wbType:wbGetTransactionData_Type, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, in_wbLevel:wbGetTransactionData_Level, in_wbParent:wbGetTransactionData_Parent, true, false)) in_LogLevel:3, in_LogMessage:"Transaction Number " + TransactionNumber.ToString + " " + "Work Block "+ wbProcessTransaction_Type.toString + " failed with exception Catch all exceptions, log them, treat or rethrow "+exception.Message+" at Source "+exception.Source, in_wbStart:wbGetTransactionData_StartTime, System. Exception: Log failure in_ChildAudit:wbGetTransactionData_Audit, io_ParentAudit:wbMain_Audit, in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError is Nothing AND + wbLogging(in_wbType:wbMain_Type, in_wbLevel:wbMain_Level, BusinessRuleException is Nothing, true, false)) in_wbParent:wbMain_Parent, in_LogLevel:4, in_LogMessage:"Work Block "+ wbMain_Type.toString + " failed with exception "+exception.Message+" at Source **BusinessRuleException: Log failure** "+exception.Source, in_wbStart:wbMain_StartTime, in_ChildAudit:wbMain_Audit, io_ParentAudit:Nothing, in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError + wbLogging(in_wbType:wbGetTransactionData_Type, is Nothing AND BusinessRuleException is Nothing, true, false)) in_wbLevel:wbGetTransactionData_Level, in_wbParent:wbGetTransactionData_Parent, in_LogLevel:3, in_LogMessage:"Transaction Number " + TransactionNumber.ToString + " " + "Work Block "+ wbProcessTransaction_Type.toString + " failed with exception "+exception.Message+" at Source "+exception.Source + ". BRE", in_wbStart:wbGetTransactionData_StartTime, in_ChildAudit:wbGetTransactionData_Audit, io_ParentAudit:wbMain_Audit, Legend in_SuppressLogging:false, in_wbStatusSuccessful:if(SystemError is Nothing AND BusinessRuleException is Nothing, true, false)) input argument output argument Finally io argument 包 NextTransaction + **NextTransaction**(in_Config:Config, in_SystemError:SystemError, in_BusinessRuleException:BusinessRuleException, io_RetryNumber:RetryNumber, io_TransactionNumber:TransactionNumber, in_TransactionItem:TransactionItem, in_wbType:in_wbType+", "+"Next Transaction", in_wbLevel:wbMain_Level+1, in_wbParent:wbMain_Type, io_Audit:wbMain_Audit) **Multi-Layer view**

Flat view (UML)

<< Framework Layer>> wbMain

