Flat view (UML) <<Framework Laver>> Global Variables Arguments + Config: Dictionary<string, object> + TransactionItem: QueueItem + TransactionData: DataTable + SystemReserved: New system.Collections.Generic.Dictionary(Of String, Object) From { ("isQueueltem", False), ("TransactionNumber", 1), ("RetryNumber", 0), {"InitRetryNumber", 0), ("ContinuousRetryNumber", 0), ("RobotFail", "") } + SystemError: Exception + wbMain_Type: String = "MainTask" + wbMain_StartTime: DateTime = now() + in_wbParentStart: DateTime = datetime.now() + in_wbParentPath: String = string.Empty + wbMain_Path: if(not(string.lsNullOrEmpty(in_wbParentPath)), in_wbParentPath+"|"+wbMain_Type, wbMain_Type) + wbMain_Audit: List<Dictionary<string, object>> = New List(of Dictionary(Of String, Object)) + in wbFinalExec: Boolean = True r: String + wblnit FinalExec: Boolean = True + io_Audit: List<Dictionary<string,object>> + wblnit_HandleError: String = "Failed" + wbGetSetTransactionData_FinalExec: Boolean = True + wbGetSetTransactionData_HandleError: String = "Failed" + wbProcessTransaction_FinalExec: Boolean = True + wbProcessTransaction_HandleError: String = "Failed" Error, Recover <<State Machine - Initialization State>> <<Framework Layer>> wblnitialisation >Errors, Abort Try executing Components: 囙 System - reserved System - reserved << Business Process Layer>> + InitAllSettings (in_ConfigFile:"Data\Config.xlsx", in_ConfigSheets:{"Settings", + CloseAllApplications(In_wbParentStart:wbInit_StartTime, in_wbParentPath:wbInit_Path, in_wbFinalExec:wbInit_FinalExec, in_wbHandleError:"Rethrow", io_Audit:wbInit_Audit, in_Config:Config) "Credentials", "Workblocks", "Tasks", "Constants"}, out_Config:Config) **Business Process Layer Interface** << Business Process Laver>> + InitAllApplications(in_wbParentStart:wblnit_StartTime, in_wbParentPath:wblnit_Path $\verb|c:wblnit_FinalExec|, in_wbHandleError: "Rethrow", io_Audit: wblnit_Audit, \\$ in_Config:Config) 皂 Log success + wbLogging(in_wbType:wblnit_Type, in_wbStart:wblnit_StartTime, in_wbPath:wblnit_Path, in_wbChildAudit:wblnit_Audit, in_wbParentStart:wbMain_StartTime, in_wbFinalExec:True, in_wbHandleError:wblnit_HandleError, in_LogMessage:["Work Block "+ wblnit_Type.toString + " successful"], in_SuppressLogging: [if(Config.Item("wbInit_SuppressSuccessful").ToString.trim.ToLower = "true", true, false)], io_ParentAudit.wbMain_Audit, io_Exception:) Catch all exceptions and log them. Output exception object Error, Recover System.Exception: Log failure 钇 wbLogging(in_wbType:wbInit_Type, in_wbStart:wbInit_StartTime, in_wbPath:wbInit_Path, in_wbChildAudit:wbInit_Audit, in_wbParentStart:wbMain_StartTime, in_wbFinalExec:True, ror wblnit_HandleError, in_LogMessage:exception.ToString, in_SuppressLogging:False, io_ParentAudit:wbMain_Audit, io_Exception:exception) <>State Machine - Get Transaction Data State>> <<State Machine - Process Transaction State>> <<Framework Layer>> <<Framework Layer>> wbGetTransactionData wbProcess + wbGetSetTransactionData_StartTime: DateTime = now() + wbGetSetTransactionData_Type: String = Config.Item("wbGetTransactionData_Type").ToString + wbProcessTransaction_StartTime: DateTime = DateTime.now() + wbProcessTransaction_Type: String = Config.Item("wbProcessTransaction_Type").ToString + wbGetSetTransactionData_Path:String = wbMain_Path+"|"+wbGetSetTransactionData_Type + wbGetSetTransactionData_Audit: List<Dictionary<String, Object>> = New List(of Dictionary(Of String, Object)) + wbProcessTransaction_Path:String = wbMain_Path+"I"+wbProcessTransaction_Type + wbProcessTransaction_Audit: List<Dictionary<String, Object>> = New List(of Dictionary(Of String, Object)) Try executing Components: **New Transaction** Try executing Components: 包 耟 System - reserved System - reserved + OrchestratorStop () + GetQueueRetries(in_TransactionItem:TransactionItem, io_SystemReserved:SystemReserved, io_Config:Config) 包 包 **Data Layer Interface Business Process Layer Interface** + GetTransactionData(in_wbParentStart:wbMain_StartTime, in_wbParentPath:wbGetSetTransactionData_Path, in_wbFinalExec:wbGetSetTransactionData_FinalExec, in_wbHandleError:"Rethrow", io_Audit:wbGetSetTransaction_Config:Config, io_TransactionItem:TransactionItem, io_TransactionData:TransactionData, in_TransactionNumber:cint(SystemReserved.Item("TransactionNumber,")), $- \textbf{Process} (in_wbParentStart: wbMain_StartTime, in_wbParentPath: wbProcessTransaction_Path, in_wbParentPath, in_wbPath, in_wbPa$ wbFinalExec:wbProcessTransaction_FinalExec, in_wbHandleError:"Rethrow", Audit:wbProcessTransaction_Audit, in_Config:Config, io_TransactionItem:Tran_TransactionNumber:cint(SystemReserved.Item("TransactionNumber;")), udit:wbGetSetTransactionData Audit, Error, Continue n_RetryNumber:cint(SystemReserved.Item("RetryNumber,"))) _RetryNumber:cint(SystemReserved.Item("RetryNumber,"))) 釦 Log success - wbLogging(in_wbType:wbProcessTransaction_Type, in_wbStart:wbProcessTransaction_StartTime, + wbLogging(in_wbType:wbGetSetTransactionData_Type, in_wbStart:wbGetSetTransactionData_StartTime, n_wbPath:wbProcessTransaction_Path, n_wbChildAudit:wbProcessTransaction_Audit, in_wbParentStart:wbMain_StartTime, in_wbFinalExec:True, _wbPath:wbGetSetTransactionData_Path, in_wbChildAudit:wbGetSetTransactionData_Audit, in_wbParentStart:wbMain_StartTime, _wbFinalExec:True, in_wbHandleError:wbGetSetTransactionData_HandleError, in_LogMessage:"successful", in_SuppressLogging in wbHandleError:wbProcessTransaction_HandleError, in LogMessage:"successful", in SuppressLogging: [if(Config.Item("wbProcessTransaction_SuppressSuccessful").ToString.trim.ToLower = "true", true, [if(Config.Item("wbGetSetTransactionData_SuppressSuccessful").ToString.trim.ToLower = "true", true, false)], D_ParentAudit:wbMain_Audit, io_Exception:) false)], io_ParentAudit:wbProcessTransaction_Audit, io_Exce Success Catch all exceptions and log them. Output exception object Catch all exceptions and log them. Output exception object 囙 System.Exception: Log failure System.Exception: Log failure $+ \textbf{wbLogging} (in_wbType: wbGetSetTransactionData_Type, in_wbStart: wbGetSetTransactionData_StartTime, \\$ + wbLogging(in_wbType:wbProcessTransaction_Type, in_wbStart:wbProcessTransaction_StartTime, in_wbPath:wbGetSetTransactionData_Path, in_wbChildAudit:wbGetSetTransactionData_Audit, in_wbParentStart:wbMain_StartTime, in_wbFinalExec:True, in_wbHandleError:wbGetSetTransactionData_HandleError, in_wbPath:wbProcessTransaction_Path, in_wbParentStart:wbMain_StartTime, in_wbFinalExec:True, n_LogMessage:exception.ToString, in_SuppressLogging:False, io_ParentAudit:wbMain_Audit, io_Exception:exception) in_wbHandleError:wbProcessTransaction_HandleError, in_LogMessage:exception.ToString, in_SuppressLogging:False, io_ParentAudit:wbMain_Audit, io_Exception:exception) <<State Machine - End Process State>> <<Framework Layer>> >Errors, Abort Try executing Components: No Data 包 System - reserved System - reserved << Business Process Layer>> + CloseAllApplications(in_wbParentStart:wbMain_StartTime, in_wbParentPath:wbMain_Path, in_wbFinalExec:wbMain_FinalExec, in_wbHandleError:"Rethrow", io_Audit:wbMain_Audit, in_Config:Config) >Errors, Abort Data Error Log success 包 - wbLogging(in_wbType:wbMain_Type, in_wbStart:wbMain_StartTime, in_wbPath:wbMain_Path, n_wbChildAudit:wbMain_Audit, in_wbParentStart:in_wbParentStart, n_wbFinalExec:True, in_wbHandleError:in_wbHandleError, in_LogMessage:"successful", n_SuppressLogging:false, io_ParentAudit:io_Audit, io_Exception:) Catch all exceptions and log them. Output exception object input argument, Framework Layer 钌 System.Exception: Log failure ed output argument + wbLogging(in_wbType:wbMain_Type, in_wbStart:wbMain_StartTime, in_wbPath:wbMain_Path, in_wbChildAudit:wbMain_Audit, in_wbParentStart:in_wbParentStart, in_wbFinalExec:True, in_wbHandleError, in_LogMessage:exception.ToString, in_SuppressLogging:False, io_ParentAudit:io_Audit, in_wbParentStart. ourple io argument. Data Laver **Business Process Layer** o_Exception:exception)