An introduction to WESS

Dr. Yonglin Lei National University of Defense Technology, Changsha, China 2015.6

Outline

- Positioning WESS
- Software architecture
- Application flowcharts
- Main components

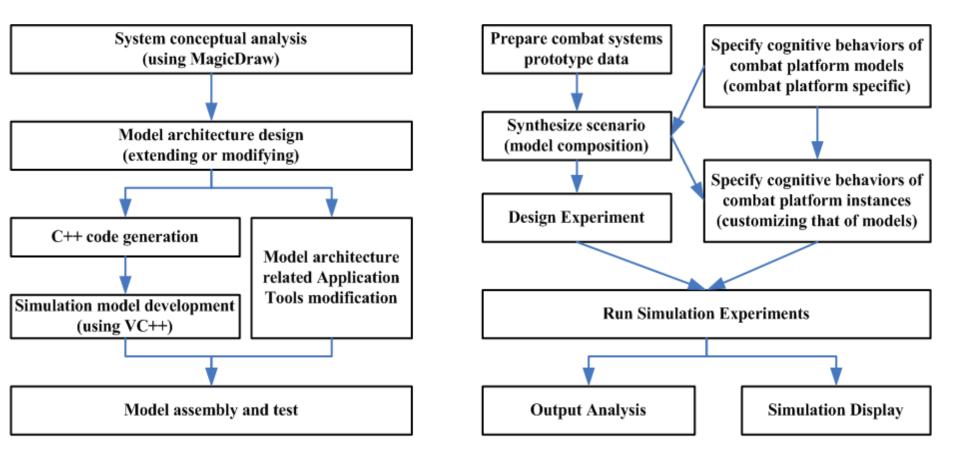
Positioning WESS

- WESS(Weapon Effectiveness Simulation System) is a generic and extensible combat system effectiveness simulation system.
- It is an engagement-level simulation system applied to acquisition and overall design of complex combat systems.
- It provides a platform-centric model architecture with net-centric modeling support based on SMP and other MDE technologies.

WESS software architecture

Application Composition	Net centricAirMissile DefenseCombat	Surface Submar Combat Comba		Helicopter Anti-Sub	Air Ground Combat	Laser Weapo Experiment	n	
Application Tools	Prototype Data Editor Editor	Cognitive Behav with CSCBML F		OOE Editor	2D Viewer	3D Viewer	Output Analyzer	
Model	Abstract Physical Behav		ictural Model Arc		gnitive Behavi	or Model Tem	blates (.py)	
Architecture	Abstract Physical Behavior Models of DMA Cognitive Cognitive Concrete Model Components of AMA (.dll) Modeling Interface Application-Specific Cognitive Behavior Models (.py)							
Modeling Services	Environment Entity manager manager	Sensor Proto manager mana	- Arbitrator			Event Ta anager man	1 5	
· · · Development & Runtime	System analysisModel designModel verification	on Code Mod generation develop	lel Model assembly S	Decision Scripts Edito		1	Data corder Playback tool	
Data & Resources		nario les Files	OpenFlight Ico files file			Analysis P Reports	layback files files	
Infrastructure & Platform		crosoft. ual C++	Boost Python	imlink/RTI Network	Altova XML	МарХ	Open Scene Graph	

WESS application flowcharts



Workflow of Model Architecture Maintaining

Workflow of Composition and Application

Main components

- 1. Composable simulation development toolset
- 2. Simulation application and analysis toolset
- 3. Generic model architecture
- 4. Model component library
- 5. A set of cognitive decision model scripts

1. Composable simulation development toolset

Functions

- Based on Simulation Modeling Platform(SMP)
- Modeling the model architecture of WESS and its components
- Enable incorporation, extension, and evolvement of WESS simulation models.
- Provide capabilities for simulation model design, verification, assembly, and code generation.

就 模型开发集成环境

文件:

编辑 模型设计 帮助

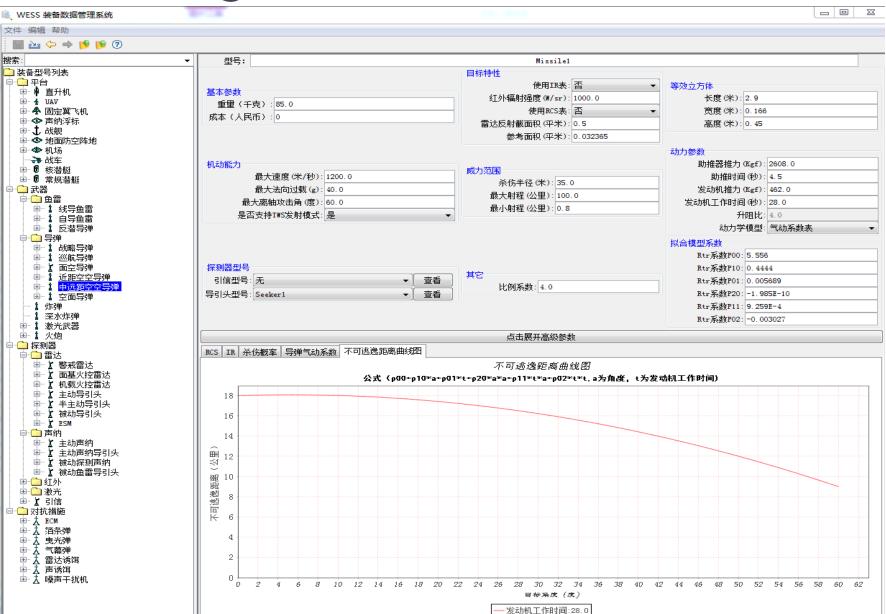


i 💼 • 🗁 🖫 🕼 🗶 % i 🞸 🐃 🚟 🗞 🦕 □ Pr... □ Pl... □ Pl... □ Pl... □ Pl... □ Pl... □ \t... □ \t... 🎁 CESS. catalogue 🔀 #include "BaseModel/IWeaponLockSubscriber.h"
#include "Sensor/tmSensor.h"
#include "Weapon/tmWeapon.h"
#include "CounterMeasure/tmCounterMeasure.h"
#include "Gun/tmGun.h" ^ 😑 🎁 file:/C:/WESS/CESS.catalogue ~ 😑 🔶 Document Root 🖃 🚸 <catalogue> Catalogue CESS #include "Sensor/tmSensorForward.h" #include "Interaction/DataType/DataTypeNamespace.h" #include "Smp/SimpleTypes.h" #include "CommonDataType/CommonDataTypeNamespace.h" 🗄 🔶 Namespace BaseModel 🗄 🔶 Namespace CommonDataType 🗄 🔶 Namespace CounterMeasure 🗄 🔶 Namespace Group namespace Platform 🗄 🔶 Namespace Gun 🗄 🔶 Namespace Interaction class tmPlatform: virtual public ::Smp::IDynamicInvocation, 🖃 🚸 Namespace Platform virtual public ::Smp::Mdk::Management::ManagedModel, 🖮 🚸 Interface IAirbase virtual public ::Smp::Mdk::Composite, virtual public ::Smp::Muk::Management::EventProvider, virtual public ::Smp::Mdk::Management::EventConsumer, virtual public ::BaseModel::tmSimEntity, 🖮 🔶 Model tmPlatform 💼 🚸 Container CounterMeasureList 🛓 🚸 Container GunList virtual public ::BaseModel::IDecisionable, virtual public ::Group::IGroupNode, 😟 🚸 Container SensorList virtual public ::BaseModel::IWeaponLockSubscriber 🗄 🔶 Container WeaponList { 🗄 🔶 Event Sink DamageResultSink // ----- Constructors/Destructor ------💼 🔶 Event Sink LockedBvRadarSink 🗄 🔶 Event Sink ReleaseByRadarSink public: tmPlatform(void); 🗄 🔶 Event Source CloseJammerSource tmPlatform (Smp::String8 name, Smp::String8 description, Smp::ICompos 🗄 🔶 Event Source FireBallisticGunSour virtual ~tmPlatform(void); 🛓 🚸 Event Source FireLaserGunSource 💼 🔶 Event Source LaunchSBMSource // ----- Fields ------🗄 🔶 Event Source OpenJammerSource private: 🗄 🔶 Event Source SensorOffSource // 巡航速度(m/s) 🗄 🔶 Event Source SensorOnSource ::Smp::Float32 CruiseSpeed; public: 🗄 🔶 Field CruiseSpeed ::Smp::Float32 GetCruiseSpeed() const {return CruiseSpeed;} 🗄 🔶 Field DHeight void SetCruiseSpeed(const ::Smp::Float32& newValue) {CruiseSpeed = newValue 🗄 🔶 Field DLength private: 🗄 🔶 Field DWidth // 最大航速(m/s) ::Smp::Float32 MaxSpeed; 🗄 🔶 Field MaxSpeed public: 표 🔶 Field MissionName ::Smp::Float32 GetMaxSpeed() const {return MaxSpeed;} 💼 🚸 Field rBallisticCountFired void SetMaxSpeed(const ::Smp::Float32& newValue) {MaxSpeed = newValue;} 🗄 🔶 Field rCMLaunchCount private: 🗄 🔶 Field rDamageLevel // 使命名字(从想定中解析。动态创建的子平台的同父平台的。) 🗄 🔶 Field rFirstCMLaunchTime ::CommonDataType::NameString MissionName; public: 🗄 🔶 Field rFirstWeaponLaunchTime 🗄 🔶 Field rFoundTargetDistance void SetMissionName(const ::CommonDataType::NameString& newValue) {Mission} 🗄 🔶 Field rFoundTargetTime private: 🗄 🔶 Field rHitTargetCount // 是否完成任务 ::Smp::Bool rMissionSuccess; 표 🔶 Field rHitWeaponCount < > > <

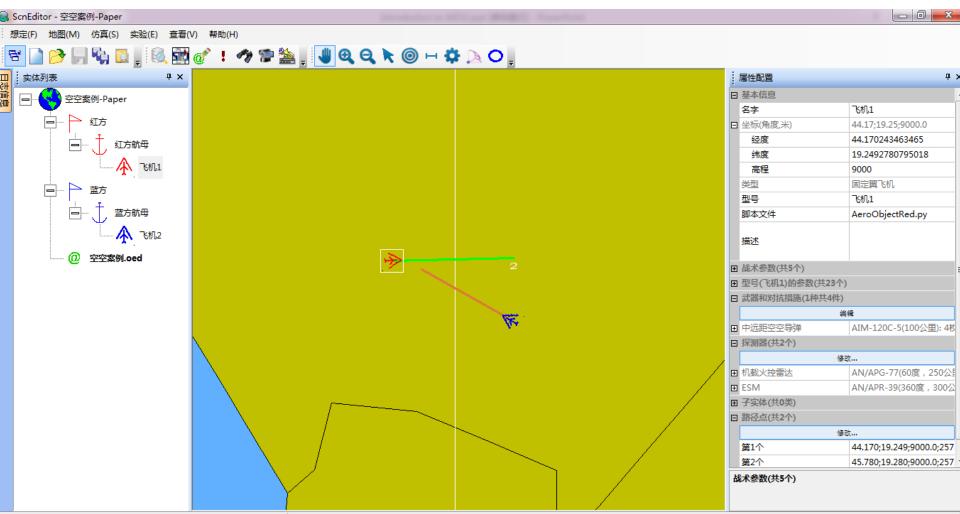
2. Simulation application and analysis toolset

- Component tools
 - DataManager: manage combat system parameters
 - ScnEditor: scenario editor
 - ScriptEditor: cognitive behavior model editor
 - DoeEditor: simulation experimental design
 - Simulator: batch runs with Monte-Caro sampling
 - SimDisplay: simulation display in 2D and 3D
 - SimLogger: log and playback simulation data
 - OutputAnalyzer: script-based output analyzer

DataManager



ScnEditor



ScriptEditor

	ScriptEditor - E:\Sim2000.SMP2\Test\Scripts\AeroObject.py		
SubAntSufface.py AeroObject.py × 4 111 if (FL:FetchEventBaraBaol("MissileWarning", "ByFireBadar")): msIrack = PI.SetchEventBaraBaol("MissileWarning", "MissileID") msIrack = PI.SetchEventBaraBaol("MissileWarning", "MissileID") msIrack = PI.Settrack(MissileNoid)(PI.Debug("音ど音号等場, SEK <addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PI.E.bebug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PI.E.bebug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String P.RTs.tht P.MissileGuide: int P.MissileGuide: int P.RTs.tht 111 else: PI.Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PI.Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PI.Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL.Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug("EdtFormInfo) 112 FL:Debug(TESERS=FL, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug(TESERS=FL, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug(TESERS=FL, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug(TESERS=FL, SEK<addk,:%s->%dE等操"%%(C_EWERAnge/1000, Phase2String PL:Debug(TESERS=FL, SEK<addk,:%s-%desers PL:Debug(TESERS=FL, SEK<addk,:%s-%desers PL:</addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s->	件编辑 视图 搜索 文档 工具 配置 Python 窗口 帮助		
SubAntSufface.py AeroObject.py × 4 111 if (FL:FetchEventBaraBaol("MissileWarning", "ByFireBadar")): msIrack = PI.SetchEventBaraBaol("MissileWarning", "MissileID") msIrack = PI.SetchEventBaraBaol("MissileWarning", "MissileID") msIrack = PI.Settrack(MissileNoid)(PI.Debug("音ど音号等場, SEK <addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PI.E.bebug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PI.E.bebug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String P.RTs.tht P.MissileGuide: int P.MissileGuide: int P.RTs.tht 111 else: PI.Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PI.Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PI.Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL.Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug("音ど音号等場, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug("EdtFormInfo) 112 FL:Debug(TESERS=FL, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug(TESERS=FL, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug(TESERS=FL, SEK<addk,:%s->%dE等操"%(C_EWERAnge/1000, Phase2String PL:Debug(TESERS=FL, SEK<addk,:%s->%dE等操"%%(C_EWERAnge/1000, Phase2String PL:Debug(TESERS=FL, SEK<addk,:%s-%desers PL:Debug(TESERS=FL, SEK<addk,:%s-%desers PL:</addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-%desers </addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s-></addk,:%s->	▼ 🖆 🚰 🖶 🕼 🚳 🐁 ங 💼 🗠 🗠 🛤 🎲 😓 ▼ 🗇 ▼ 💕 🖽 🖽 🥪 沙 🥂 🛩 🗐	J 🕨 🕨 🔳	🔗 🥵 💝
11 (F). FetchbvenEParaBool ("MissileWarning", "ByFireRadar")): MissileID = FI. FetchbvenEParaBool ("MissileWarning", "MissileID") miTrack = FI. GetTack (MissileID) FI. Ebeta (MissileIC) FI. Ebeta (MissileVardi() FI. Ebeta = P_AvoidMissile		<u></u> ; · · ·	
155 MissileID = FI.FetchEventPeraltr ("KissileWarning", "MissileID") 157 Initrack = FI.detTack(MissileID) 157 If (PI.Debug("atchewsParaltr("KissileWarning", "MissileID") 157 FI.Debug("atchewsParaltr("kissileWarningOuard") 158 PI.Debug("atchewsParaltr("KissileWarningOuard") 159 PI.ScheduleGuardEvent("RadarWarningOuard", "RadarWarningOuardHandler,", "RadarWarning(Platf") 150 PI.Debug("atchewsParaWateWate("RadarWarningOuard") 151 PI.Debug("atchewsParaWateWate("RadarWarningOuard") 152 PI.Debug("atchewsParaWateWate("RadarWarningOuard") 153 PI.Debug("atchewsParaWateWateWateWateWateWateWateWateWateWat			4
<pre>set malTrack = PI.GetTrack(MissIleID) set [PI.Debug("當注意警导课,距离(AddW, is=>/規題导彈"%(C_BNBRange/1000,Phase25tring PI.Debug("當注意警导课,距离(AddW, is=>/規題导彈"%(C_BNBRange/1000,Phase25tring PI.Debug("高注意警导课)器(AddW, 運動率"%(C_BNBRange/1000,Phase25tring(PlatformInfo) FI.Debug("高注意警导课)器(AddW, 運動率"%(C_BNBRange/1000,Phase25tring(PlatformInfo) FI.Debug("高注意警导课,距离(AddW, 運動率"%(C_BNBRange/1000,Phase25tring(PlatformInfo) FI.Debug("高注意警导课,距离(AddW, 運動率"%(C_BNBRange/1000,Phase25tring(PlatformInfo) FI.Debug("高注意警导课,距离(AddW, is=>/規題导彈"%(C_BNBRange/1000,Phase25tring(PlatformInfo) FI.Debug("高注意警导课,距离(AddW, is=>/規題导彈"%(C_BNBRange/1000,Phase25tring(PlatformInfo) FI.Debug("高注意警导课,距离(AddW, is=>/規題导彈"%(C_BNBRange/1000,Phase25tring(PlatformInfo) FI.Debug("高注意警导课,距离(AddW, is=>/規題导彈"%(C_BNBRange/1000,Phase25tring(PlatformInfo) FI.Debug("高注意警导课,距离(AddW, is=>/規題导彈"%(C_BNBRange/1000,Phase25tring(PlatformInfo) FI.Debug("富注意警导彈,距离(AddW, is=>/規題專理"%(C_BNBRange/1000,Phase25tring(PlatformInfo) FI.Debug("富注意會导彈,距离(AddW, is=>/Latting(FI)) FI.Debug("電話意")) FI.Debug("電話意")) FI.Debug("電話意")) FI.Debug("電話ECHTING) FI.Debug("電話ECHTING)) FI.Debug("電話ECHTING) FI.Debug("電話ECHTING)) FI.Debug("完TATTING)) FI.Debug("完TATTING)) FI</pre>		^	ø
11 [P1.DistanceTo (malTrack) <-C_INNRange):			B. Drag August int
PI.Debug("電法含譽与美,提薦《KdKM,:%s=>规避导弹"%(C_KMMRange/1000,Phase2String) P.MissileGuide:int PI.Debug("電法含影与此)(); P.MissileGuide:int PI.Debug("電法含影力)(); P.MissileGuide:int PI.Debug("電法含影力)(); P.MissileGuide:int PI.Debug("電法含影力)(); P.MissileGuide:int PI.Debug("電法含影力)(); P.MissileGuide:int PI.Debug("電法含影力)(); P.MissileGuide:int PI.Debug("電法含影影力)(); P.MissileGuide:int PI.Debug("電法含影力)(); P.MissileGuide:int PI.Debug("電法合影()			
19 PI.AutoMissileAvoid() P.AutoMissileAvoid() 19 PI.AutoMissileAvoid() P.RTB: int 10 PI.ScheduleGuardEvent("RadarWarningGuard", "RadarWarningGuardHandler,", "RadarWerning(Plattor 101 PI.ScheduleGuardEvent("RadarWarningGuard", "RadarWarningGuardHandler,", "RadarWerning(Plattor 101 PI.Debug("雪达希景·ks=>从短海/导弹, 距离>kd0%, 距离+sd0%, mas+sd0%,		se2String	
Image = Paction FileName = P_action Image = P_action FileName PileName Image = P_action FileName FileName Imator FileName			
##2 PI.ScheduleGuardZvent("RadarWarningGuardIandler", "RadarWe / AvoidMissile(PlatformInfo) ##2 PI.Debug("雪法探测导弹,距离>%100,Phase2String(PlatformInfo) / BreakLock(PlatformInfo) ##2 PI.Debug("与洪音警:%5=>规键导弹"%Fhase2String(PI.Phase)) / DragAway(PlatformInfo) ##2 PI.Debug("与洪音警:%5=>规键导弹"%Fhase2String(PI.Phase)) / DragAway(PlatformInfo) ##2 PI.Debug("与洪音警音"%Fhase2String(PI.Phase)) / DragAway(PlatformInfo) ##3 Def RadarWarningGuardIandler(PlatformInfo): / DragAway(PlatformInfo, TargetID) ##4 Def RadarWarningGuardIandler(PlatformInfo): / EnterGVRRight(PI) ##1 EnterGVRRight(PI) / EnterGVRRight(PI) / PlatformInfo.Debug("雪波音》字》、服高く4000, %ts=>-½%20~%%%(C_ENMRange/1000, Phase2String(PlatformInfo) / InitDecision(PlatformInfo) // PlatformInfo.RemoveExternalEvent("RadarWarningGuard") / EnterGVRRight(PlatformInfo) / InitDecision(PlatformInfo) // PlatformInfo.RemoveExternalEvent("RadarWarningGuard") / faitaformInfo.GetClosestWarningMissile() / InitDecision(PlatformInfo) // BlatformInfo.Debug("雪波音響力, 服高く4000, %ts=*%(distance, Phase2String(PlatformInfo. / MissileJaunchPlatformInfo) // PlatformInfo.Debug("雪波音響力, 服高く4000, %ts=*%(distance, Phase2String(PlatformInfo. / MissileJaunchPlatformInfo) // PlatformInfo.Debug("雪波音響力, 服高く4000,	180 - PI.Phase = P AvoidMissile		P_RTB : int
<pre>isa Fl.Debug("當达探測导弹,距离>4204%,继续\$*\$(C_RNMRange/1000,Phase2String(PlatformInfo) ist else: else: fl.Debug("导弹音警:\$s=>规型导弹"\$Phase2String(Pl.Phase)) Fl.Debug("导弹音警:\$s=>规型语导弹"\$Phase2String(Pl.Phase)) Fl.Debug("导弹音警:\$s=>规型语导弹"\$C_RNMRange/1000,Phase2String(PlatformInfo) fl.Phase = P_AvoidMissile fl.Debug("電达音警导弹,距离(*ddWA,*\$s=>规型导弹"\$(C_RNMRange/1000,Phase2String) fl.Debug("電达音警导弹,距离(*ddWA,*\$s=>规型导弹"\$(C_RNMRange/1000,Phase2String) fletformInfo.Debug("富达音警导弹,距离(*ddWA,*\$s=>规型导弹"\$(C_RNMRange/1000,Phase2String) fletformInfo.Debug("富达音警导弹,距离(*ddWA,*\$s=>规型导弹"\$(C_RNMRange/1000,Phase2String) fletformInfo.Debug("富达音警导弹,距离(*ddWA,*\$s=>规型导弹"\$(C_RNMRange/1000,Phase2String) fletformInfo.Debug("富达音警导弹,距离(*ddWA,*\$s=>规型语导弹"\$(C_RNMRange/1000,Phase2String) fletformInfo.Debug("富达音警导弹,距离(*ddWA,*\$s=>规型导弹"\$(C_RNMRange/1000,Phase2String) fletformInfo.Debug("富达音警导弹,距离(*ddWA,*\$s=>规型语导弹"\$(C_RNMRange/1000,Phase2String) fletformInfo.Debug("富达音警导弹,距离(*ddWA,*\$s=>规型导弹"\$(C_RNMRange/1000,Phase2String) fletformInfo.Debug("富达音警导弹,距离(*ddWA,*\$s=>规型导弹"\$(C_RNMRange/1000,Phase2String) fletformInfo.Debug("富达音警导弹,距离(*ddWA,*\$s=>规型导弹"\$(C_RNMRange/1000,Phase2String) fletformInfo.Debug("富达音警导弹,距离(*ddWA,*\$s=>规型导弹"\$(C_RNMRange/1000,Phase2String) fletformInfo.DistanceTo(Inissile) fletformInfo.DistanceTo(Inissile) fletformInfo.DistanceTo(Inissile) fletformInfo.DistanceTo(Inissile) fletformInfo.Debug("富达音響导弹,距离(*ddWA,*\$s"\$(distance,Phase2String(PlatformInfo). fletformInfo.Debug("完达音響导弹,距离(*ddWA,*\$s"\$(distance,Phase2String(PlatformInfo). fletformInfo.Debug("完达音響导弹,距离(*ddWA,*\$s"\$(distance,Phase2String(PlatformInfo). fletformInfo.Debug("完达音響导弹,距离(*ddWA,*\$s"\$(distance,Phase2String(PlatformInfo). fletformInfo): fletformInfo.Debug("完计音频性的</pre>			P_TacticFight : int
<pre>138 PI.Debug("電法探测导弹,距离>4d84,继续***(C_NWRRange/1000,Phase2String(Platfor else: PI.Debug("导弹合警:%s>规避导弹"*Phase2String(PI.Phase)) PI.AutoMissileAvoid() PI.Phase = P_AvoidMissile +#雷达探测导弹->采取机造机dr(PlatformInfo): PlatformInfo.Debug("高达含警导弹,距离<4k24,*s=>规避导弹"*(C_NWMRange/1000,Phase2String(PlatforMInfo) / EnterBVRFight(PlatformInfo) / InitDecision(PlatformInfo) / InitDecision(PlatformInfo) / InitDecision(PlatformInfo) / InitDecision(PlatformInfo) / InitBecision(PlatformInfo) / MissileAarNaringGuardTunfo) / RadarWarningGuardTunfo) / RadarWarningGuardTunfo) / RadarWarningGuardTunfo) / RadarWarningGuardTunfo) / RadarWarningGuardTunfo) / RadarWarningGuardTunfo) / TargetDebug("美術ECM") / #1[Pl.GetLimBConflict / #1[Pl.GetLimBConflict] / #1[Pl.G</pre>			f AvoidMissile(PlatformInfo)
14 □ else: 15 □ Else: 15 □ FI. Debug("导弹合整:%s=>规避导弹"%Phase2String(PI.Phase)) 15 □ FI. AutoMissileAvoid() 15 □ FI. Debug("导弹合整:%s=>Mu测点 15 □ FI. Debug("导弹合整:%s=>Mu测点 15 □ Gef. RadatWarningGuardHandler(PlatformInfo): 15 □ PlatformInfo. These P_AvoidMissile 15 □ PlatformInfo. Phase = P_AvoidMissile 15 □ PlatformInfo. RemoveExternalEvent("RadatWarningGuard") 15 □ return 15 □ PlatformInfo.DistanceTo (missile) 15 □ fissile = PlatformInfo.DistanceTo (missile) 15 □ fissileGuide(PlatformInfo) 15 □ fissile = PlatformInfo.DistanceTo (missile) 16 □ fissileGuide(PlatformInfo) 17 □ fissileWarningGuardFunc(PlatformInfo): 18 □ fissileWarningGuardFunc(PlatformInfo): 19 □ <td></td> <td>ng(Platfc</td> <td></td>		ng(Platfc	
if initiality (Pinder Markey and Minicipal Part Markey) (Pinder Pinder Pind			
## PI.Phase = P_AvoidMissile ## FI.Phase = P_AvoidMissile ## ## <tr< td=""><td></td><td>-</td><td></td></tr<>		-	
188		=	-
159<	_		
19 def RadarWarningGuardHandler(PlatformInfo):			_ · · · ·
191 PlatformInfo.Debug("雷达告警导弹,距离<%dKM,:%s=>规避导弹"%(C_RWMRange/1000,Phase2Stri f FormFly(PlatformInfo) 192 PlatformInfo.AutoMissileAvoid() f GuideOverHandler(PlatformInfo) 193 PlatformInfo.AutoMissileAvoid() f InitDecision(PlatformInfo) 194 PlatformInfo.RemoveExternalEvent("RadarWarningGuard") f InitDecision(PlatformInfo) 195 return f EditAmane (PlatformInfo) 196 def RadarWarningGuardFunc(PlatformInfo): missile = PlatformInfo.Debug("雷达告警导弹,距离%dKM,:%s"%(distance, Phase2String(PlatformInfo.F 196 def RadarWarningOverHandler(PlatformInfo): f MissileLaunchHintHandler(PlatformInfo) 197 #@达探测导弹,计算距离 f MissileWarningOverHandler(PlatformInfo) 198 D def MissileWarningOverHandler (PlatformInfo): f MissileWarningOverHandler(PlatformInfo) 199 #PiletformInfo.Debug("富达告警导弹,距离%dKM,:%s"%(distance, Phase2String(PlatformInfo.F f MissileWarningOverHandler(PlatformInfo) 107 PlatformInfo.Debug("富达告警导弹,距离%dKM,:%s"%(distance, Phase2String(PlatformInfo.F f MissileWarningGuardFunc(PlatformInfo) 108 def MissileWarningOverHandler(PlatformInfo): f RadarOverHandler(PlatformInfo) f RadarWarningGuardFunc(PlatformInfo) 109 #FF = toPlatformInfo.DistoncetormInfo.F f RadarWarningGuardFunc(PlatformInfo) f			f EnterCACFight(PlatformInfo, TargetID)
134 PlatformInfo.Phase = P_AvoidMissile 135 PlatformInfo.Phase = P_AvoidMissile 136 PlatformInfo.RemoveExternalEvent("RadarWarningGuard") 137 return 138 initDecision(PlatformInfo) 139 + # # 达探测导弹, 计算距离 139 + # # 达探测导弹, 计算距离 139 + # # 达探测导弹, 计算距离 139 - def RadarWarningGuardFunc(PlatformInfo): 131 PlatformInfo.Debug(" 當法告答导弹, 距离%dKM,:%s"% (distance, Phase2String(PlatformInfo.F 141 - # # # # # # # # # # # # # # # # # # #		hase2Stri	f FormFly(PlatformInfo)
193 PlatformInfo.Phase = P_AvoidNissile 194 PlatformInfo.Phase = P_AvoidNissile 194 PlatformInfo.RemoveExternalEvent("RadarWarningGuard") 195 return 196 # # 达探测导弹, it 算距离 197 # 都 达探测导弹, it 算距离 198 = def RadarWarningGuardFunc(PlatformInfo): 199 missile = PlatformInfo.GetClosestWarningMissile() 191 # PlatformInfo.Debug(" 雷达音警导弹, 距离 \dfkM, : \s m \dfk (distance, Phase2String (PlatformInfo).F 192 # PlatformInfo.Debug(" 雷达音警导弹, 距离 \dfkM, : \s m \dfk (distance, Phase2String (PlatformInfo).F 193 # PlatformInfo.Debug(" 雷达音警导弹, 距离 \dfkM, : \s m \dfk (distance, Phase2String (PlatformInfo).F 194 # PlatformInfo.Debug(" 雷达音警母导弹, 距离 \dfkM, : \s m \dfk (distance, Phase2String (PlatformInfo).F 195 = def MissileWarningOverHandler(PlatformInfo) 196 = HissileWarningOverHandler(PlatformInfo): 197 = PlatformInfo 198 = def MissileWarningGuardFunc("ECM")>0): 198 = def MissileWarningGuardFunc("ECM")>0): 199 = PlatformInfo 199 # PlatformInfo 199 # PlatformInfo 190 # PlatformInfo 191 <	192 PlatformInfo.AutoMissileAvoid()		f GuideOverHandler(PlatformInfo)
194 Platforminfo.RemoveExternalEvent("RadarwarningGuard") 195 return 196 return 197 #雷达探测导弹,计算距离 198 def RadarWarningGuardPunc(PlatformInfo): 199 missile = PlatformInfo.GetClosestWarningMissile() 199 distance = PlatformInfo.DistanceTo(missile) 190 #PlatformInfo.Debug("當法告警导弹,距离%dkM,:%s"%(distance,Phase2String(PlatformInfo).F 190 #SpiteSamp 191 #Gifterminfo.Debug("當法告警导弹,距离%dkM,:%s"%(distance,Phase2String(PlatformInfo).F 192 tetrn distance < C_RWMRange	<pre>193 PlatformInfo.Phase = P_AvoidMissile</pre>		
<pre>1 continue 1 continue 1 f LockByRadarOverHandler(PlatformInfo) 1 f LockByRadarOverHandler(PlatformInfo) 1 missile = PlatformInfo.GetClosestWarningMissile() 1 missile = PlatformInfo.DistanceTo(missile) 1 f LockByRadarOverHandler(PlatformInfo) 1 missile = PlatformInfo.DistanceTo(missile) 1 missile = PlatformInfo.Debug("當达告警导弹,距离をMM,:%s"%(distance,Phase2String(PlatformInfo.F 1 f PlatformInfo.Debug("當达告警导弹,距离をMM,:%s"%(distance,Phase2String(PlatformInfo.F 1 missileWarningOverHandler(PlatformInfo) 2 missileWarningOverHandler(PlatformInfo): 2 missileWarningOverHandler(PlatformInfo): 3 missileWarningOverHandler(PlatformInfo): 4 WeaponFailedHandler(PlatformInfo): 4 WeaponFailedHandler(PlatformInfo): 4 WeaponFailedHandler(PlatformInfo): 4 WeaponFailedHandler(PlatformInfo): 4 WeaponFailedHa</pre>			
197 #雷达探测导弹,计算距离 197 #雷达探测导弹,计算距离 198 def RadarWarningGuardFunc(PlatformInfo): 199 missile = PlatformInfo.GetClosestWarningMissile() 201 distance = PlatformInfo.DistanceTo(missile) 202 #PlatformInfo.Debug("雷达告警导弹,距离%dKM,:%s"%(distance,Phase2String(PlatformInfo.F 203 #PlatformInfo.Debug("雷达告警导弹,距离%dKM,:%s"%(distance,Phase2String(PlatformInfo.F 204 #导弹告警察除 205 def MissileWarningOverHandler(PlatformInfo): 206 #PitE toPlatformInformInterface.toPlatformInfo): 207 PI = PlatformInfo 208 #if(PI.GetItemCount("ECM")>0): 209 #PI.Debug("关闭ECM") 209 #PI.SlockByRadar()): 201 if(PI.IsLockByRadar()): 202 PI.Debug("异弹告警察除:			
<pre>198 日 def RadarWarningGuardFunc(PlatformInfo): missile = PlatformInfo.GetClosestWarningMissile() distance = PlatformInfo.DistanceTo(missile) #PlatformInfo.Debug("審法告答导弹,距离%dKM,:%s"%(distance,Phase2String(PlatformInfo.F return distance < C_RWMRange #导弹告警解除 004 #导弹告警解除 005 日 def MissileWarningOverHandler(PlatformInfo): 006 #PI = toPlatformInterface.toPlatformInterface PI = PlatformInfo #FI.Debug("关闭ECM") #PI.CloseECM() 10 #PI.CloseECM() 11 PI.Debug("异弹告警解除:%s=>提脱锁定"%Phase2String(PI.Phase)) 15 U.Debug("异弹音警解除:%s=>提脱锁定"%Phase2String(PI.Phase)) 15 U.Debug("异弹音警解除:%s=>提脱锁定"%Phase2String(PI.Phase)) 15 U.Debug("异弹音警解除:%s=>提脱锁定"%Phase2String(PI.Phase)) 15 U.Debug("异弹音警解除:%s=>提脱锁定"%Phase2String(PI.Phase)) 15 U.Debug("异弹音警解除:%s=>提脱锁定"%Phase2String(PI.Phase)) 15 U.Debug("异弹音警解除:%s=>提脱锁定"%Phase2String(PI.Phase)) 15 U.Debug("异弹音警解除:%s=>提脱锁定"%Phase2String(PI.Phase)) 15 U.Debug("异弹音警解除:%s=>提脱锁c(U.f) 15 U.Debug("异弹音警解除:%s=>提脱锁c(U.f) 15 U.Debug("异弹音警解除:%s=>提脱锁c(U.f) 15 U.Debug("异弹音警解除;%s=>提脱锁c(U.f) 15 U.Debug("异弹音警解除;%s=>提脱锁c(U.f) 15 U.Debug("异弹音警解除;%s=>提脱锁c(U.f) 15 U.Debug("异弹音警解除;%s=>U.f) 15 U.Debug("异弹音警解除;%s=>提脱锁c(U.f) 15 U.f) 15 U.debug("异弹音管容解除;%s=>URD;%s=>U.f) 15 U.f) 15 U.f)</pre>			· · · · · · · · · · · · · · · · · · ·
<pre>missile = PlatformInfo.GetClosestWarningMissile() distance = PlatformInfo.DistanceTo(missile) #PlatformInfo.Debug("雷达告警导弹,距离%dKM,:%s"%(distance,Phase2String(PlatformInfo.F return distance < C_RWMRange #导弹告警解除 #导弹告警解除 #导弹告警解除 Get MissileWarningOverHandler(PlatformInfo): #PI = tcPlatformInterface.tcPlatformInterface PI = PlatformInfo #PI = tcPlatformInfo #PI = tcPlatformInfo #PI = platformInfo #PI = tcPlatformInfo #PI = tcPlatformInfo #PI = platformInfo #PI = platformInfo #PI = platformInfo #PI = ccPlatformInfo #PI = tcPlatformInfo #PI = platformInfo #PI = ccPlatformInfo #PI = platformInfo #PI = platformInfo #PI = platformInfo #PI = ccPlatformInfo #PI = platformInfo #PI = ccPlatformInfo #PI = platformInfo #PI = platformInfo #PI = platformInfo #PI = ccPlatformInfo #PI = platformInfo #PI = platform</pre>			f MissileGuide(PlatformInfo)
200 distance = PlatformInfo.DistanceTo(missile) f MissileWarningHandler(PlatformInfo) 201 #PlatformInfo.Debug("雷达告警导弹,距离%dKM,:%s"%(distance,Phase2String(PlatformInfo.F f MissileWarningOverHandler(PlatformInfo) 202 #导弹告警解除 f MissileWarningOverHandler(PlatformInfo): f 203 #导弹告警解除 f RadarOrientMainTarget(PlatformInfo) 204 #导弹告警解除 f RadarOrientMainTarget(PlatformInfo) 205 def MissileWarningOverHandler(PlatformInfo): f RadarWarningGuardFunc(PlatformInfo) 206 #PI = tcPlatformInfo f RadarWarningGuardFunc(PlatformInfo) 206 #PI = platformInfo f RadarWarningGuardHandler(PlatformInfo) 205 #PI = blatformInfo f RadarWarningGuardHandler(PlatformInfo) 206 #PI = blatformInfo f RtB(PlatformInfo) 207 PI = PlatformInfo f StepDecision(PlatformInfo) 208 #PI.Oebug("关闭ECM") f TargetFoundHandler(PlatformInfo) 209 #PI.Debug("关闭ECM") f TargetLostHandler(PlatformInfo) 211 f If (PI.IsLockByRadar()): f WeaponFailedHandler			f MissileLaunchHintHandler(PlatformInfo)
<pre>201 #PlatformInfo.Debug("雷达告警导弹,距离%dKM,:%s"%(distance,Phase2String(PlatformInfo.F return distance < C_RWMRange</pre> <pre> 4 #导弹告警解除 4 #导弹告警解除 5 def MissileWarningOverHandler(PlatformInfo): 6 def MissileWarningOverHandler(PlatformInfo): 7</pre>			f MissileWarningHandler(PlatformInfo)
202 return distance < C_RWMRange		ormInfo.F	f MissileWarningOverHandler(PlatformInfo
203 上 204 #导弹告警解除 205 def MissileWarningOverHandler(PlatformInfo): 206 #PI = tcPlatformInterface.tcPlatformInterface 207 PI = PlatformInfo 208 #if(PI.GetItemCount("ECM")>0): 209 #PI.Debug("关闭ECM") 210 #PI.CloseECM() 211 If(PI.IsLockByRadar()): 212 if(PI.IsLockByRadar()): 213 PI.Debug("异弹告警察院:%3=>探脱锁定"%Phase2String(PI.Phase))			-
204 非子理官答解除 205 def MissileWarningOverHandler (PlatformInfo): 206 #PI = tcPlatformInterface.tcPlatformInterface 207 PI = PlatformInfo 208 #if(PI.GetItemCount("ECM")>0): 209 #PI.Debug("关闭ECM") 210 #PI.CloseECM() 211 if(PI.IsLockByRadar()): 212 if(PI.IsLockByRadar()): 213 PI.Debug("异弹告警察院:\$a=>探脱锁定"\$Phase2String(PI.Phase))	203 L		• • • • •
206 #PI = tcPlatformInterface.tcPlatformInterface 207 PI = PlatformInterface.tcPlatformInterface 208 #PI = tcPlatformInterface.tcPlatformInterface 209 #PI.Debug("关闭ECM") 210 #PI.CloseECM() 211 if (PI.IsLockByRadar()): 212 if (PI.IsLockByRadar()): 213 PI.Debug("异弹告擎经险:%a=>探脱锁定"%Phase2String(PI.Phase))			
207 PI = PlatformInfo 208 #if(PI.GetItemCount("ECM")>0): 209 #PI.Debug("关闭ECM") 210 #PI.CloseECM() 211 if(PI.IsLockByRadar()): 212 if(PI.IsLockByRadar()): 213 PI.Debug("异弹告警察院:%a=>提脱锁定"%Phase2String(PI.Phase))	T · · ·		
208 #if(PI.GetItemCount("ECM")>0): 209 #PI.Debug("关闭ECM") 210 #PI.CloseECM() 211 if(PI.IsLockByRadar()): 212 if(PI.IsLockByRadar()): 213 PI.Debug("异弹告擎经险:%a=>提脱锁定"%Phase2String(PI.Phase))			
209 #PI.Debug("关闭ECM") 210 #PI.CloseECM() 211 if (PI.IsLockByRadar()): 212 日 if (PI.IsLockByRadar()): 213 PI.Debug("导弹告擎经险:%a=>提脱锁定"%Phase2String(PI.Phase))			f RTB(PlatformInfo)
210 #PI.CloseECM() f TargetFoundHandler(PlatformInfo) 211 if (PI.IsLockByRadar()): f TargetLostHandler(PlatformInfo) 212 日 if (PI.IsLockByRadar()): f WeaponFailedHandler(PlatformInfo) 213 PI.Debug("导弹告警经险:%a=>探脱锁定"%Phase2String(PI.Phase)) f WeaponFailedHandler(PlatformInfo)			f StepDecision(PlatformInfo)
211 f TargetLostHandler(PlatformInfo) 212 □ if (PI.IsLockByRadar()): f WeaponFailedHandler(PlatformInfo) 213 □ PI.Debug("导弹告警経险:%3=>探脱锁定"%Phase2String(PI.Phase)) f WeaponFailedHandler(PlatformInfo)			f TargetFoundHandler(PlatformInfo)
212 日 if (PI.IsLockByRadar()): 213 日 PI.Debug("导弹告警经院:%a=>提脱锁定"%Phase2String(PI.Phase))			
212 PT. Debug ("导弹告警超除:%a=>探脱绑定"%Phase2String (PT. Phase))	I		
J WeaponSuccessHandler(PlatformInto)	213 PT.Debug("导弹告警解除:≵s=>掃脱锁完"≵Phase2String(PT.Phase))	-	
		+	J vveaponSuccessHandler(PlatformInfo)

DoeEditor

DUELUIL	U I			实验	1 2.	200.	100.	
WESS仿真实验设计 - AirCombatExample.oed				实验	2 2.	200.	150.	
基本信息、实验因子、实验方案、实验响应				实验	3 2.	200.	50.	
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □		因子水	₽设置	实验4	4 2.	300.	100.	
AirCombatExample 因子名称	因子代号 数据类型▲ 因子名称 所儿 最大探测距离 ESI	属対象 作用范围 水平个数 序号 M1 型号 5 1	水平值	实验	5 2.	300.	150.	
角度分辨率	AngleResoluition float	2	112.5 175. 237.5	实验		300.	50.	-11
B RedCarrier 接收传播因子	ReceiveFactor float	4 5	300. 50.	实验		200.	100.	- E
FighterB 接收损耗	ReceiveLoss float							-82
	ReceiveAntennaGain float			实验		200.	150.	-88
▲ 扫描周期 ESM2(型号)	ScanCycle double			实验	9 3.5	200.	50.	-82
▲ □□□□2至57 → Seeker2(型号)	AntennaHeight float SignalProcessLoss float			实验	10 3.5	300.	100.	- 11
△ III 18 52(1323)496 Missile4(型号)	SignalBandwidth float			实验	11 3.5	300.	150.	- 11
▲ In in 5 m 300 ■ ▲ 藍方 重量	Weight float			实验	12 3.5	300.	50.	
■ ↓ BlueCarrier 最大俯仰角	MaxPitch float			实验	13 5.	200.	100.	
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	MaxTrackNumber int			实验		200.	150.	-11
Radar1(型号) ✓ 最大探测距离	HavBaran Class	请选择	水平 更改水平值					
△ 最小確仰角 ESM1(型号)	WESS仿真实验设计 - 空空案例.oed							×
△ 量子			1					
△ 最小探测距离 ▲ Missile3(型号)	空空案例-Paper	数据代号	数据名称	数据类别	<u>数据名称</u> 数据类型 各种雷达锁定 终态数据	型 <u>所属対象</u> 居飞机1	▼ ■ ★ ■ ★ ●	^
	1 日本	✓ rLockTargetTimes	各种雷达锁定对方的次数	终态数据 i	首次锁定目标 终态数1 首次收到被敌 终态数1	居 飞机1 居 飞机1	实例 实例	
	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	✓ rFirstLockTime	首次锁定目标时刻	终态数据(首次锁定目标 终态数技 首次收到被敌 终态数技	居 飞机1 居 飞机1	实例 实例	
		 ✓ rFirstRadarWarningTime ✓ rFirstLockDistance 	首次收到被敌雷达锁定报告时刻	终态数据 (终态数据 f	首次被导弹锁 终态数打 首次被导弹锁 终态数打	居 飞机1 居 飞机1	()() ()()()()()()()()()()()()()()()()()	
		✓ rFirstLockDistance ✓ rFirstRadarWarningDistance	首次锁定目标距离 首次收到被敌雷达锁定报告距离		是否完成任务 终态数指 完成任务时间 终态数指	居 飞机1 居 飞机1	()() 实例 实例	
		✓ rFirstMissileWarningTime	首次被导弹锁定时刻	终态数据(被命中武器数 终态数	居 飞机1 居 飞机1	实例	
	X AN/AF	✓ rFirstMissileWarningDistance	首次被导弹锁定距离	终态数据(最后丢失目标 终态数指 发射武器数量 终态数据	居 飞机1 居 飞机1	实例	
	🖌 WGU-	sAccelN	法向加速度	状态数据 f	首次发射武器 终态数 发射炮弹数量 终态数	名 3011 居 飞机1 居 飞机1	实例	
	AIM-1	sClimbRate	爬升率	状态数据 f	发射对抗措施 终态数扎	名 (101,1 居 飞机1 居 飞机1	实例	
		sRemainFuel	剩余燃油	状态数据(首次发射对抗 终态数据 首次发现目标 终态数据		实例 目	
	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	sNy	法向过载	状态数据 f	 設伤等级 终态数据 武器锁定对方… 终态数据 	居 飞机1 居 飞机1	实例	
	□ □ □ □ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	sNh	法向过载的水平分量	状态数据 f	 击毀目标数量 终态数 首次发现敌方… 终态数 	居 飞机1 居 飞机1	头例 实例	E
		sNv	法向过载的垂直分量	状态数据 f	首次发现敌方 终态数打 首次发射距离 终态数打	居 飞机1 居 飞机1	实例 实例	
		sNx	切向过载	状态数据 f	运行最远距离 终态数据 总行程距离 终态数据	居 飞机1 居 飞机1	实例 实例	
	AN/AP	✓ rMissionSuccess	是否完成任务	终态数据	首次被敌方发 终态数1 首次被敌方发 终态数1	居 飞机1 居 飞机1	实例 实例	
	¥ 9B-11	✓ rMissionSuccessTime	完成任务时间	终态数据(居 飞机1 居 飞机1	实例 实例	
	R-27E	✓ rHitWeaponCount	被命中武器数	终态数据 i	加入仿真时刻 终态数据 退出仿真时刻 终态数据	居 飞机1 居 飞机1	实例 实例	
	±	✓ rFoundTargetTime	首次发现目标时间	终态数据(工作时长 终态数排 首次发现目标 终态数排	居 飞机1 居 AN/APG-77	实例 	
		✓ rLostTargetTime	最后丢失目标时间	终态数据(最后丢失目标 终态数打 首次发现目标 终态数打	居 AN/APG-77 居 AN/APG-77		
		✓ rWeaponCountLaunched	发射武器数量	终态数据 i 🗸	发现目标平均 终态数技	名N/APR-39		
				•				
		添加表达式终值全部终值	全部	全部不选	清除状态数据	据 清除金	全部	
	•		"					•
							1	

×

1992. 4

雷达反射截面积_想定1.红方.苏27长机 最大探测距离_R...

WESS仿真实验设计 - 想定1.oed

┌实验方案列表・

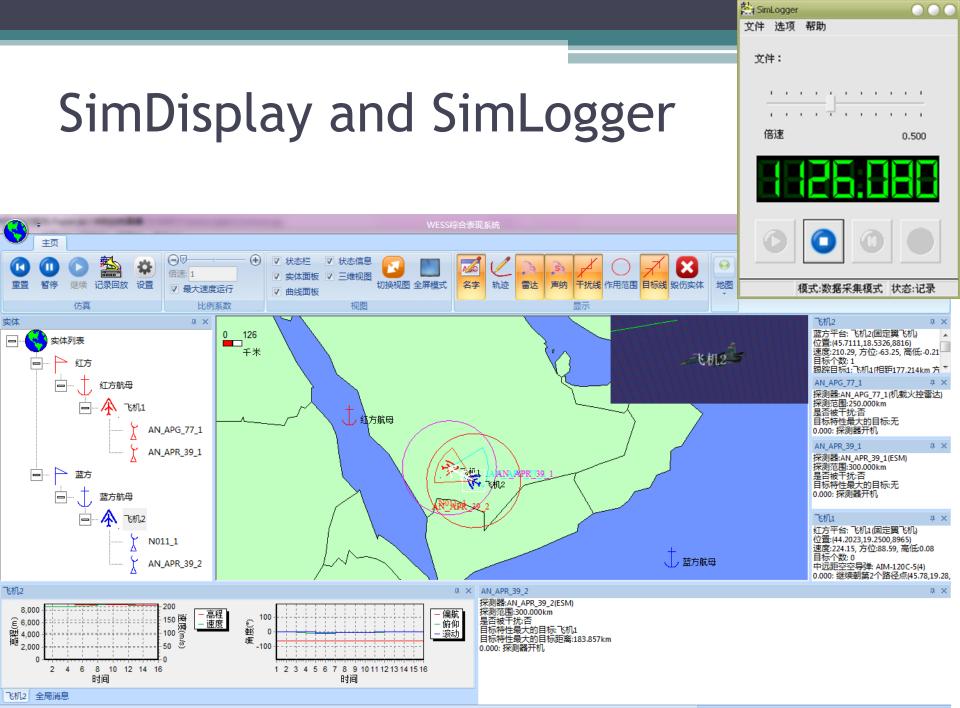
全面设计表

基本信息 实验因子 实验方案 实验响应

Simulator

🎬 AirCombatExample - WESS仿真器

文件(亚) 查看(亚) 运控(亚) 编辑(亚) 帮助(吐)							
🔁 🔚 💿 🕕 🕥 📿 🤗 🎘	🎇 🚱 🎲 🕜 🗸 🗹 分布表现 🗹 实时运行 1 🔹 🔹 倍实时 🗸						
仿真配置与控制 · · · · · · · · · · · · · · · · · · ·	高低-15.73度,目标特性0;信噪比0		模型数据				Ψ×
	[ESM2_1] : 300.200: 目标【FighterA】的类型识别为:[飞机]!		名称	值	类型	类别 ▽	
框架文件 案例\AirCombatExample.assembly 🔍	[Missile3_2] : 301.000: 模型创建(ID=16,Missile3_2,tmFarAAM)		sLat	19.166637	Float32	状态	
	[Missile3_2] : 301.000: 注意: 未装配引信, 导引头将控制引爆战斗部!		sLon sSpeed	46. 135422 199. 372559	Float32 Float32	状态	=
模型路径 C:\WESS\SimModels	[Seeker1_2] : 301.000: 模型创建(ID=17,Seeker1_2,tmActiveRadarSeeker)		sAzimuth	82.653549	Float32	状态	
	[FighterA] : 301.000: 建立火控关系: [Radar1_1]+[Missile3_2]->[FighterB]		sAlt	8988.966797	Float32	状状状状状状状状状状状状状状状状状状状状状	_
实验文件 C:\WESS\Projects\空空案例\AirC 🔍	[FighterA]: 301.000: 目标1点钟方向,距离36.990km		sElevation sRoll	-3.526640 -65.685135	Float32 Float32	\ √恣 状态	
	[FighterA] : 301.000: [脚本] 向(FighterB)发射导弹:规避导弹		sPitch	-3.526640	Float32	状态	
-AirCombatExample	[FighterA]: 301.000: 导弹Missile4_1距离18780.2方位0.532963转弯过载-5		sHeading sVoyage	82.653549 74828.507813	Float32 Float32	状态	
合 红方	[FighterA] : 302.000: 朝目标FighterB发射Missile3型中远距空空导弹[Missile3 _2]		sAccel	1.363657	Float32	状态	
-RedCarrier	_ ^{2]} [Missile3 2] : 302.000: 预定攻击目标为: FighterB		sTurnRate	-3.452971	Float32	状态	
-FighterB	[Missile3_2]: 302.000: 坐标原点为: 46.122565,19.168440,9006.322364		sThrust sDrag	49411.031250 34136.628906	Float32 Float32	状态 状态	
Radar2 1	[Missile3_2]: 302.000: 发射方向: 俯仰0度, 方位126度		sMostRiskTa	Missile4_1	NameString	状态	
ESM2 1	[Missile3_2] : 302.000: 预测目标点为: (46.2255,18.8511,8985)		sNextWPLon sNextWPLat	46.500000 19.280001	Float32 Float32	状态	
⊡ Missile4 1	[FighterA] : 302.000: 导弹Missile4 1距离17544.5方位0.794871转弯过载-5		sNextWPAlt	9000.000000	Float32	状态	
Seeker2_1	[Radar2_1] : 302.010: 发现目标【Missile3_2】, 距离: 36558米, 方位7.49度;			2	Int32	状态	
□	高低17.08度,目标特性0.512603;信噪比9.5652		sRemainTorp sRemainCMCount	0	Int32 Int32	(八念) 状态	
- BlueCarrier	[Radar2_1] : 302.010: 目标【Missile3_2】的类型识别为:[拦截弹]!		sTarzetCount	2	Int32	状态	~
- FighterA	[FighterB] : 302.010: [脚本] 发现目标(Missile3_2),继续远距空战		模型消息				Ψ×
_Radar1_1	[FighterA] : 303.000: 导弹Missile4_1距离17437.6方位1.13645转弯过载-		0.000: 模型创刻				^
-ESM1_1	4.86355		0.000: 导入默ì		SS\Scripts\Ae:	roObject.j	ру】
🖻 Missile3_1	[FighterA] : 304.000: 导弹Missile4_1距离17360.1方位1.58076转弯过载- 4.41924		0.000: 导入成]				
Seeker1_1	[FighterA] : 305.000: 导弹Missile4 1距离16791.2方位2.14908转弯过载-		0.000: 继续朝第				
⊨ Missile3_2	3.85092		0.000: 初始位5		8,9000),初速!	夏(257.22)	2),万位
Seeker1_2	[FighterA] : 306.000: 导弹Missile4_1距离16767.6方位2.71917转弯过载-		0.000: 执行任务		- 10 00 0000		
	3.28083	L L	0.000: 初始化, 0.000: [脚本]	, ヨ則12面(45.) - 委決	5, 19. 28, 9000)		_
	[Missile3 2]: 306.500: 助推段结束,进入中段		0.000: [jup本] 239.800: 注意:		插为『Frighter	ъВ	
	[FighterA] : 307.000: 导弹Missile4_1距离16764.4方位3.08817转弯过载-		239.800: (聖恩: 239.800: [脚本				交战
	2.91183		240.000:[脚本				
	[FighterA] : 308.000: 导弹Missile4_1距离16775.4方位3.42171转弯过载-		246.000: [脚本				
J	2.57829	2	246.000: [脚本	:] 雷达探测导。	弹,距离>20KM,	继续远距空	战
	[FighterA]: 309.000: 导弹Missile4_1距离16800.6方位3.7437转弯过载-2.2563		290.000: 建立:			ile3_1]->	[Fight:
	[Radar2_1] : 309.090: 丢失目标: FighterA, 距离(34150米), 原因是:未能进入		290.000: 目标(
	候选探测目标列表		290.000: [脚本				
		v 2	290.000: 目标F	FighterB距离43	2931.2方位0.0	585821转弯	Sift截N ≚
TESS效能仿真系统	」 ————————————————————————————————————		•		当前步长:0.030	309, 330	-
				-			11



OutputAnalyzer-analysis script

💥 ScriptE	ditor - E:\Sim2000.SMP2\Test\Scripts\DefaultAnalysis.py		
文件 编辑	目 视图 搜索 文档 工具 配置 Python 窗口 帮助		
	C = Ø Ø Ø Å Pa Ra \\ \\ \ \ \ A \\ \ \ \ \ \ \ \ \ \ \ \		= @ @ ¹⁵
SubA	ntiSurface.py AeroObject.py AnalysisBase.py DefaultAnalysis.py ×		4 ▷ ▼
85	Scenario=ExperimentInfo.GetScenario()		(¢)
86	Doe=ExperimentInfo.GetDOE()		
87	#		from AnalysisCommon import *
88	mydir=GetDefReportDir(ExperimentInfo)		💖 import os
90	<pre>#templatefile=ExperimentInfo.DefScriptDir()+'\\defaulttempl</pre>	ate.dot! #模板文	🟨 😵 import time
91	templatefile=ExperimentInfo.DefScriptDir()+'\\templateforma		😵 import win32com.client
92	word = WordWrap(templatefile) #新建空白word文档,应用	指定模板	AeroCoeffTableOptDict : dict
93	word.deleteAll()	的最尾端	IRTableOptDict : dict
94	word.pageSetup()		PlatformParaOptDict : dict
95	word.saveAs(ExperimentInfo.ReportPathName())		RCSTableOptDict : dict
96	filePathName=word.getFilePathName()		
97	sel = word.wordSel #获取Selection对象		ResponseOptDict : dict
98	doc =word.wordDoc		v SensorParaOptDict : dict
100	#		ThrustTableOptDict : dict
101	word.addStyledPara(Text=Scenario.Name+'综合分析报告',Style=	•标题•) #报告标题	ŧ. 旭 v WeaponParaOptDict : dict
102	Time=time.localtime(time.time())	#生成报告的8	
103	word.addNewParagraph(Text='%s年%s月%s日,星期%s'%(Time[0],T	ime[1], Time[2], Nu	umber: v WordApp.Visible : bool
104	<pre>word.addNewParagraph(Text='By '+Doe.Author,Alignment=1)</pre>	#实验设计人	f Generate(ExperimentInfo)
105			J Generate(Experimentatio)
106	#基本信息	1- 1-	
107	word.addNewParagraph(Text='基本信息',LineSpacing = 2,Style=	'	,Bold
108	word.addNewParagraph(Text='测试想定:'+Scenario.Name) word.addNewParagraph(Text='保存路径:'+word.getSaveDir())	≠想定名称 #想定和报告	1. 吃 你
109 110	word.addNewParagraph(lext='%%4+#84%:'+word.getSaveDir())	带水品 从已不过了风 田	1 #67 10C.
110	#		
112	word.addNewParagraph(Text='想定配置和装配参数',LineSpacing	= 2.Stvle='标题 1	l'.Si:
113	EquipmentItemPrint(doc, ExperimentInfo)	#装备型号配	
114	WaypointPrint (doc, ExperimentInfo)	#平台航路设	
115	PlatformParaPrintAll(doc,ExperimentInfo,PlatformParaOptDict) #平台参数	
116	<pre>#RCSTablePrintAll(doc,ExperimentInfo,RCSTableOptDict)</pre>	#RCS表	
117	#AeroCoeffTablePrintAll(doc,ExperimentInfo,AeroCoeffTableOp		
118	<pre>#ThrustTablePrintAll(doc,ExperimentInfo,ThrustTableOptDict) #ThrustTablePrintAll(doc,ExperimentInfo,ThrustTableOptDict)</pre>		
119 120	<pre>#IRTablePrintAll(doc,ExperimentInfo,IRTableOptDict) WeaponParaPrint(doc, ExperimentInfo,WeaponParaOptDict)</pre>	♯IR表 #武器参数	
120	SensorParaPrint(doc, ExperimentInfo,SensorParaOptDict)	#探测器参数	
122	Sensorrararrino (ass, Experimentinio, Sensorraraoposiso)	#1/10/01/08/22/350	
123	#		
124	newpara=AddNewParagraph(doc,Text='脚本配置',LineSpacing = 2	,Style='标题 1',s	Size=:
125	ScriptPrint(doc, ExperimentInfo) #脚本信息		
126			
127	│ #实验设计及相应		==
128	newpara=AddNewParagraph(doc,Text='实验设计和响应',LineSpaci	ng = 2,Style='标版	题 1'
100		11 列: 35	Selected: 0 换行符 : \r\n 编码 : cp936

OutputAnalyzer-generated report

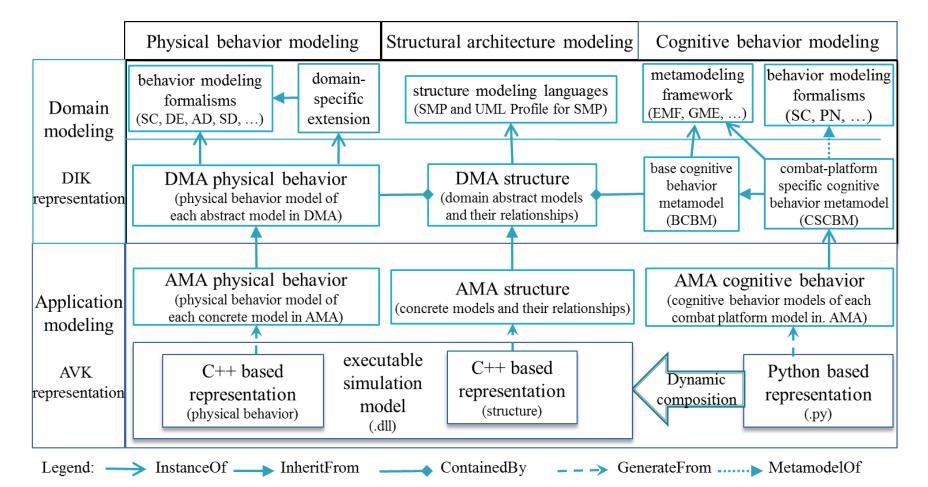
. <u>.</u>	空空案例.doc	[兼容模式] - Word		? 🕋 –
之件 开始 插入 设计 页面布局	引用 邮件 审阅 视图			
	*** H MUMB (****	,神风,间风飞水,水	x (mp) p	
寻航 ▼×				
	4.2.1 平台: 航母	奕+/		
奥索文档 🔎 🔹	红方航母(共1个)↩	1	蓋方航母(共1个)↩	4
天题 页面 结果	22.3 , 38.0 , 0.0 ,	0.00	4.0 , 58.0 , 0.0 , 0.040	4
3.2 飞机各种雷达锁定对方的次数分析	له			
3.3 飞机武器锁定对方的次数分析		977, in the constant of the co		
3.4 命中概率分析	4.2.2 平台:固定	農№111尖4		
▲ 3.5 武器制导历程:	飞机1(共2个)√	-	飞机 2 (共 2 个)↔	
▲ 3.5.1 武器制导历程_飞机2:				
3.5.1.1 方案序号_1				_
4 想定配置和装配参数	19.2492780795, 44.17	702434635 , 9000.0 , 257.2224 1	8.5187684255 , 45.7386949765 , 9000	0.0, 257.2224
▲ 4.1 装备(型号)配系信息				
4.1.1 装备 (型号) 配系: 航母类	19.28 , 45.78 , 9000	.0, 257.222¢ 1	9.1449195575 , 44.5678790583 , 9000	0.0, 257.22244
4.1.2 装备 (型号) 配系: 固定翼飞机类	له ا			
▲ 4.2 平台航路(经度,维度,高度(米)				
4.2.1 平台:航母类	4.3 平台参数配置。	h		
4.2.2 平台:固定翼飞机类				
▲ 4.3 平台参数配置	4.3.1 平台配置:魚	亢母类↩		
4.3.1 平台配置: 航母类	〈当前平台类型没有定〉	刻華打印的余勝山		
4.3.2 平台配置: 固定翼飞机类		01371J 47928780274		
▲ 4.4 武器型号参数 4.4.1 参数:中远距空空导弹	4.3.2 平台配置: 國			
4.4.1 参数: 中心起空空守理 ▲ 4.5 探测器型号参数				
4.5.1 参数: ESM	参数名称	飞机 1₽	飞机 2∻	1
4.5.2 参数: 机载火控雷达	长度↩	18.924	18.924	
		05000000.0	05000000.0	
5.1 决策脚本:航母类	成本↩	250000000	250000000₽	1
	 高度₽	5.08₽	5.08₽	

3. Generic model architecture

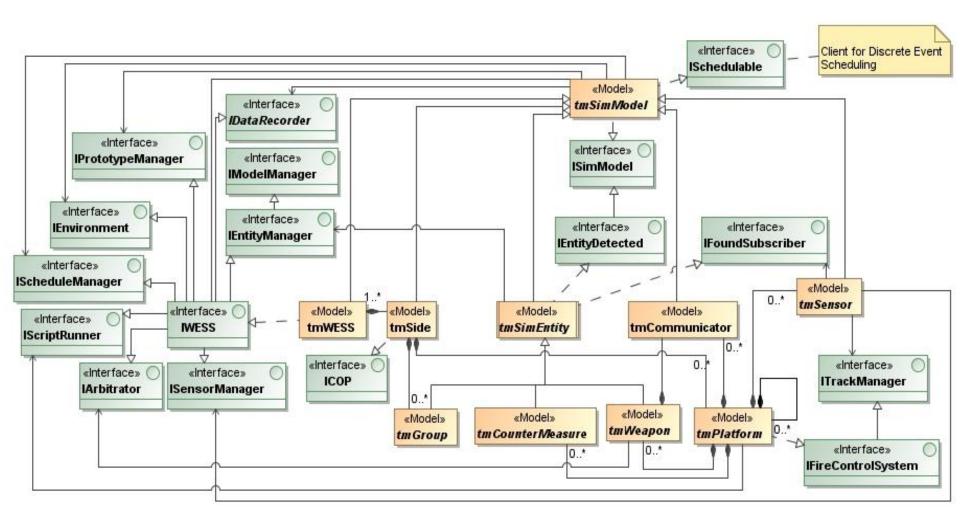
- Rational
 - Targeted to engagement-level simulation, few to few
 - Platform-centric combat with net-centric support
 - Separation Domain MA from Application MA
 - All models within DMA are abstract, i.e. non-instantiable
 - All models within AMA are concrete, i.e. instantiable
 - AMAs inherit the DMA
 - Separation cognitive behaviors from physical behaviors
 - Stable physical behaviors represented in C++
 - Variable cognitive behaviors represented in Python
 - Cognitive modeling interface
 - Formalized structure representation based on SMP
 - Quasi-formalized behavior representation

3. Generic model architecture

Modeling framework of WESS model architecture



Top view of the model architecture



Top view of the model architecture

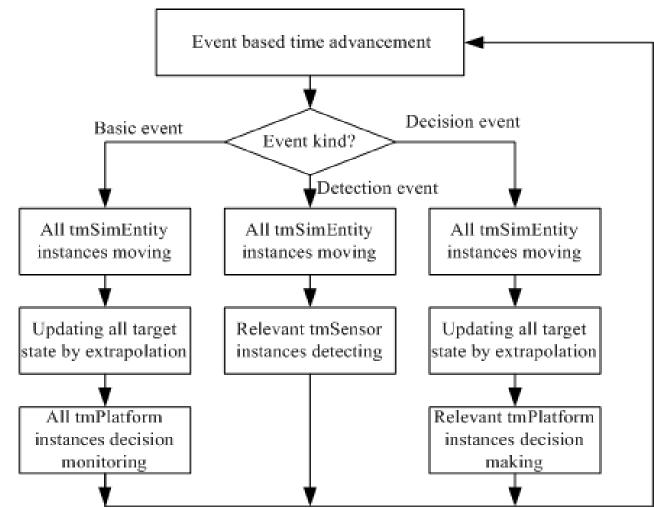
Abstract event-based time advancement algorithm

- Basically, the motion model of every entity, i.e. subclass of *tmSimEntity* model, is continuous. To advance time, the simulation time should be discretized with a basic step. At the functional modeling level, the detection model of a sensor is discrete time, probably with time-varying steps. For the platform model excluding motion, another important functionality is to invoke its cognitive behavior model for tactical decision-making over all simulated time when the platform is "alive" in the mission. To represent these cognitive behaviors, both discrete time and discrete event time advance mechanisms are demanded. The former is applied generally for regular situation monitoring; whereas the latter is useful to schedule a decision point ahead of a certain time already known in the moment or given the quantified condition under which a decision would be made.
- To precisely and efficiently model these different time advance requirements, a discrete event time advance mechanism is chosen by the simulator. Each model will register its time advance requirements to simulator in terms of simulator events for discrete event or cycle events for discrete time. For this sake, two kinds of events are available for registration. One is **decision event**, and the other is **detection event**. For the continuous time advance requirement, a basic step specified by the user is used to register the third event kind, i.e., **basic event**. When a simulator event is triggered, depending on its kind, different execution sequences happen as shown in next slide.

3. Generic model architecture

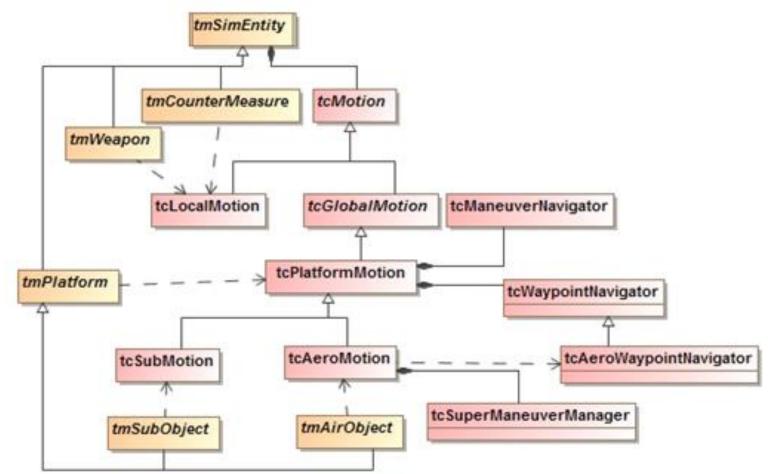
Top view of the model architecture

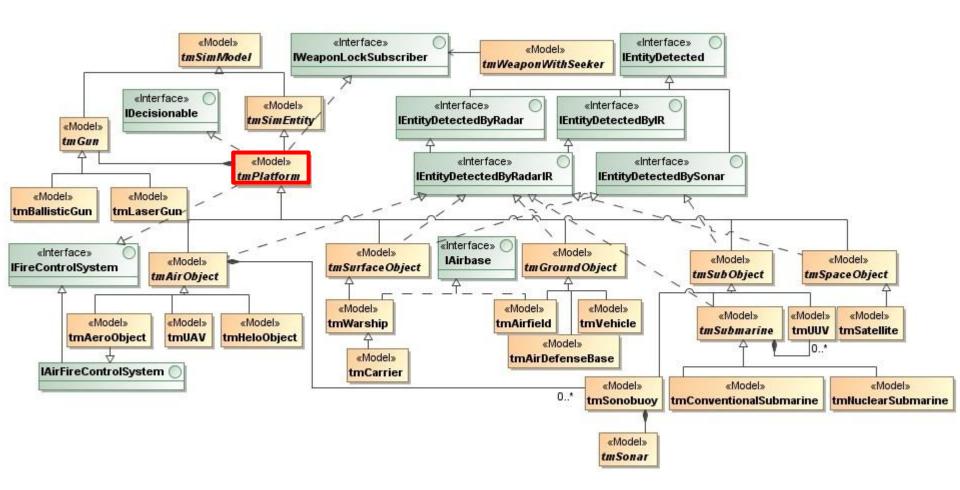
• Abstract event-based time advancement algorithm



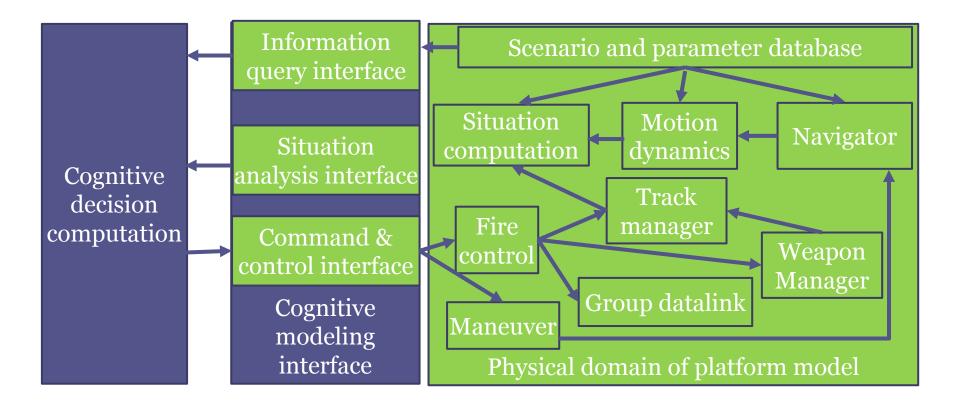
Top view of the model architecture

Abstraction-oriented motion modeling

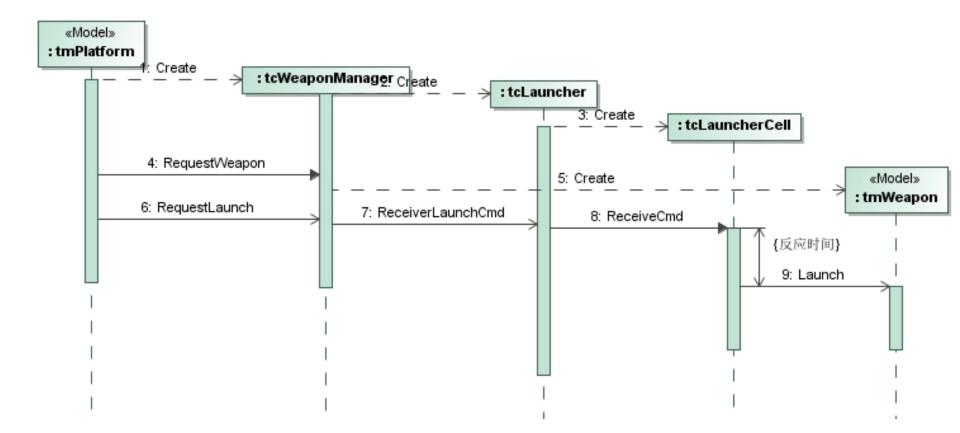




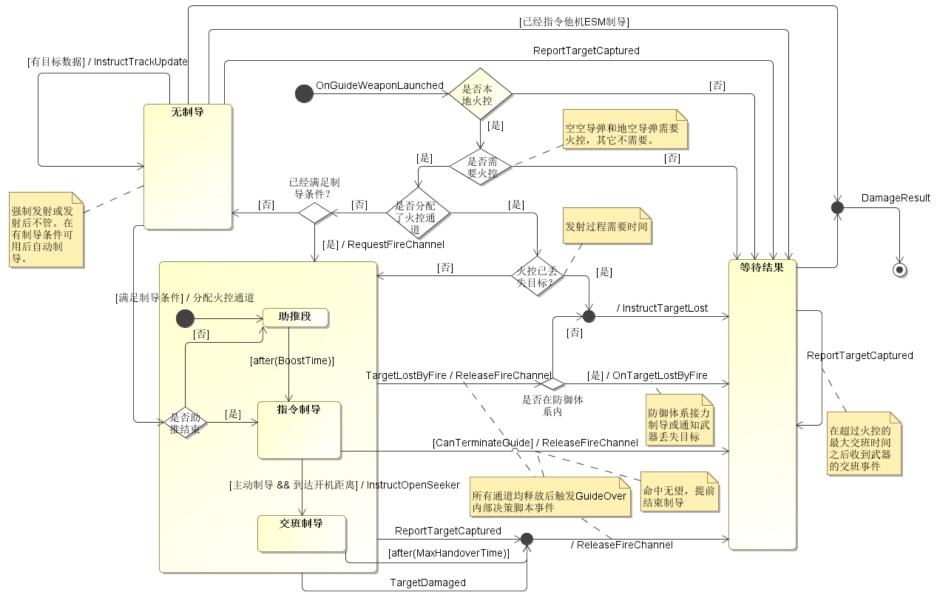
Modeling framework of combat platform models



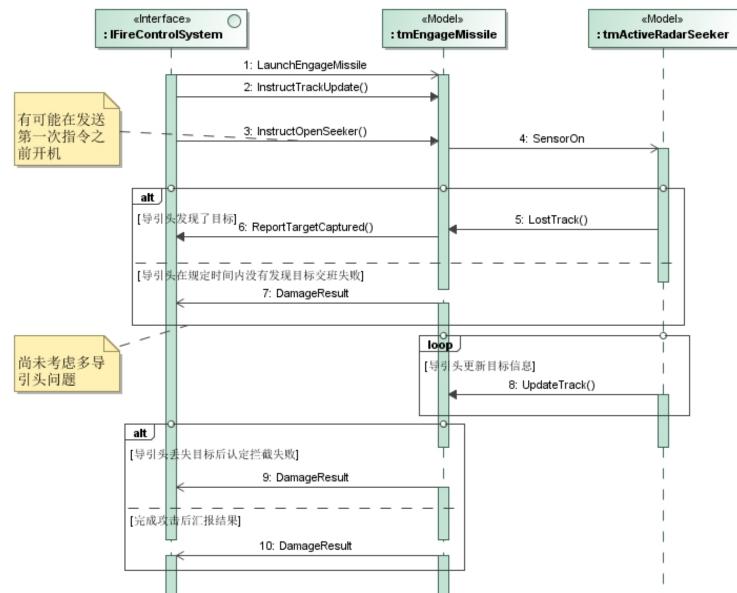
Weapon launch management



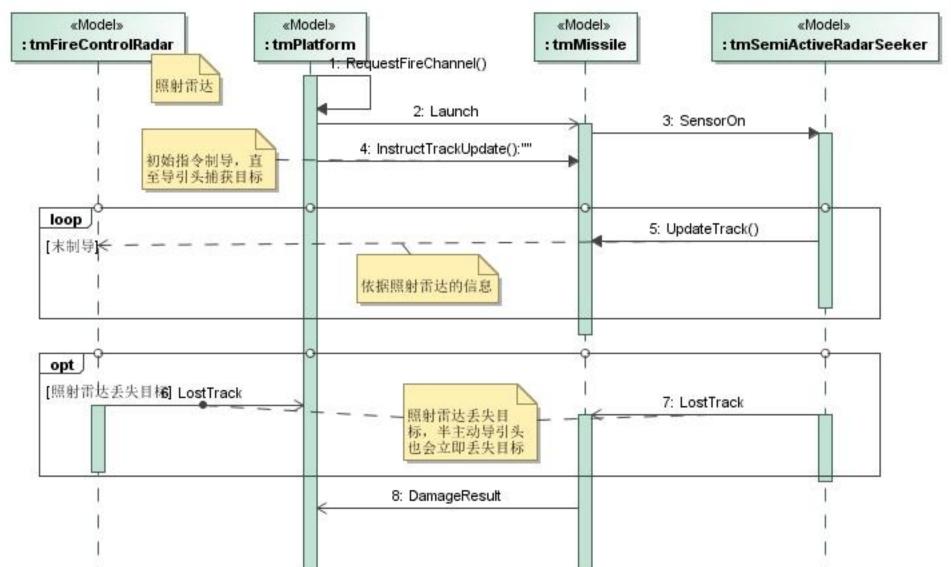
Fire control weapon management



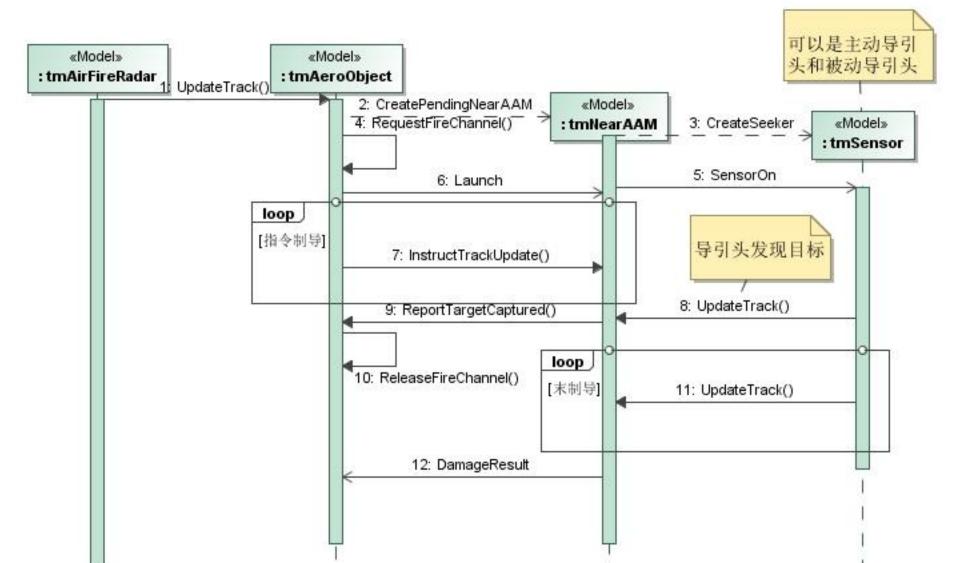
Missile guiding (active radar seeker)



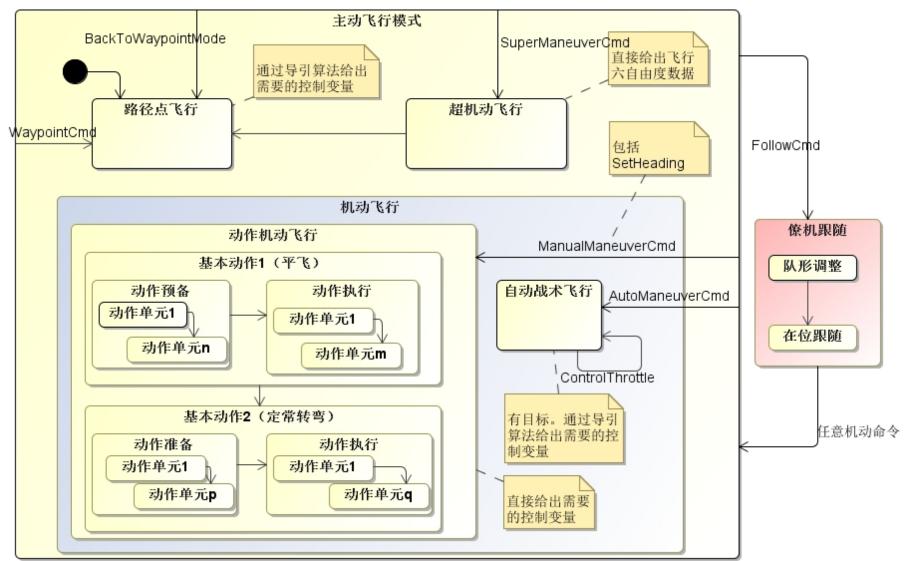
Missile guiding (semiactive radar seeker)



Missile guide (near air to air missile)

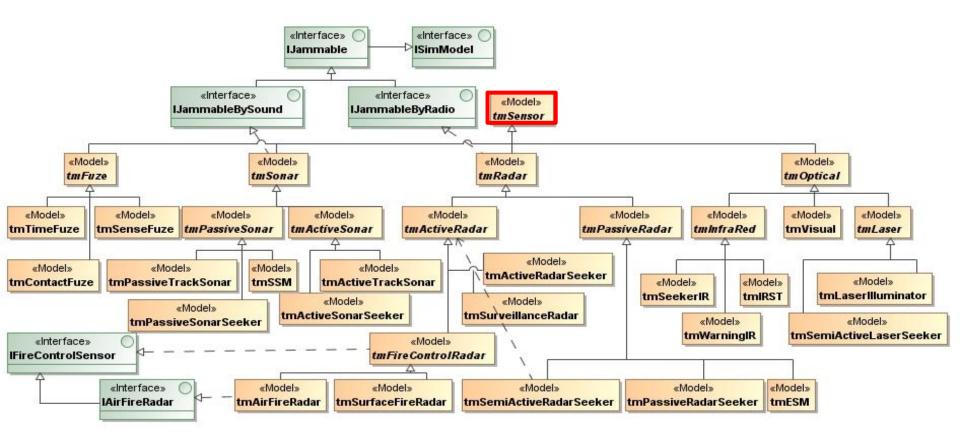


Platform motion statecharts

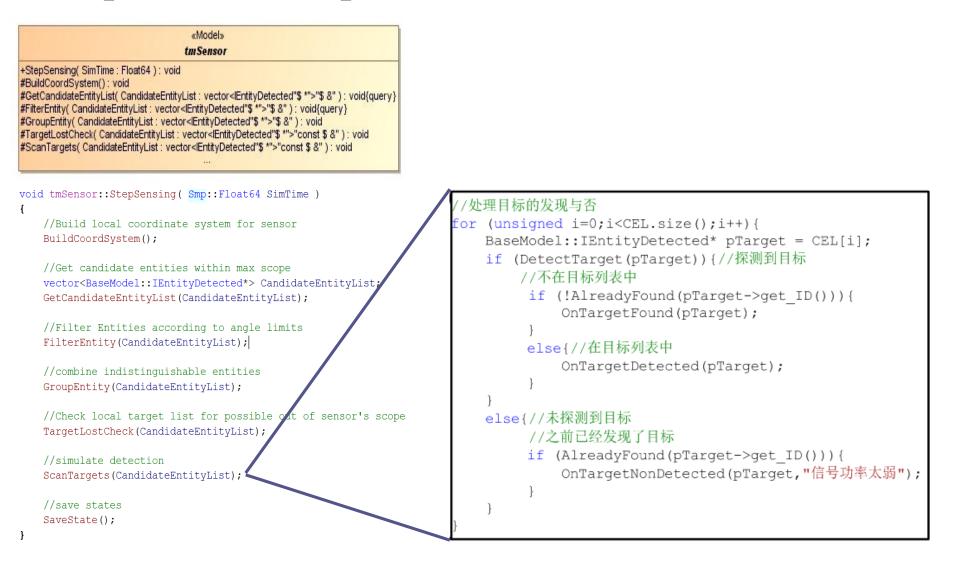


3. Generic model architecture

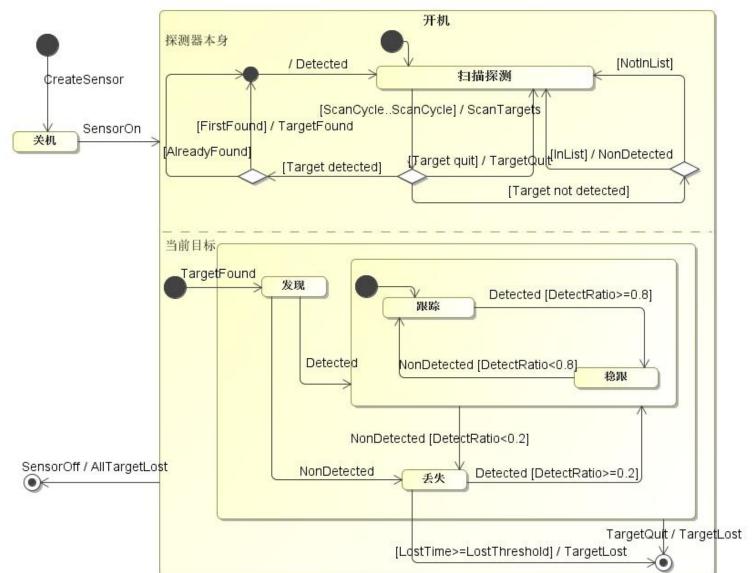
Sensor model architecture



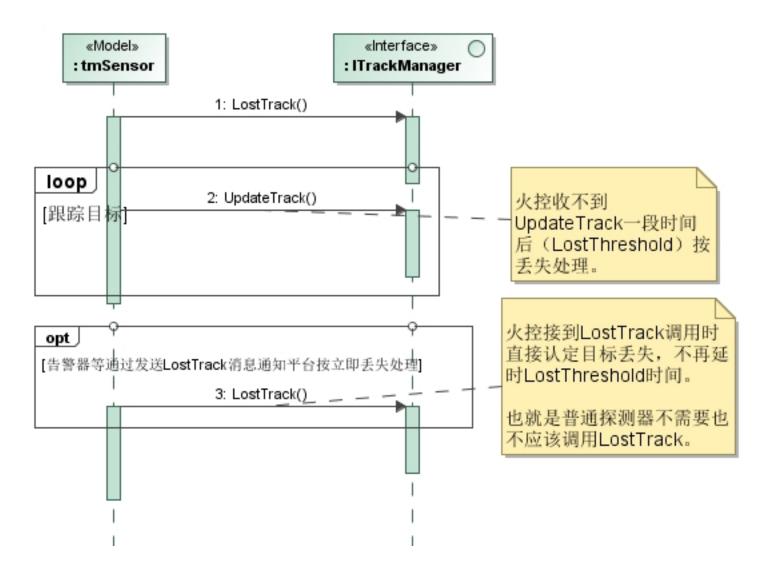
Template Method pattern based detection model



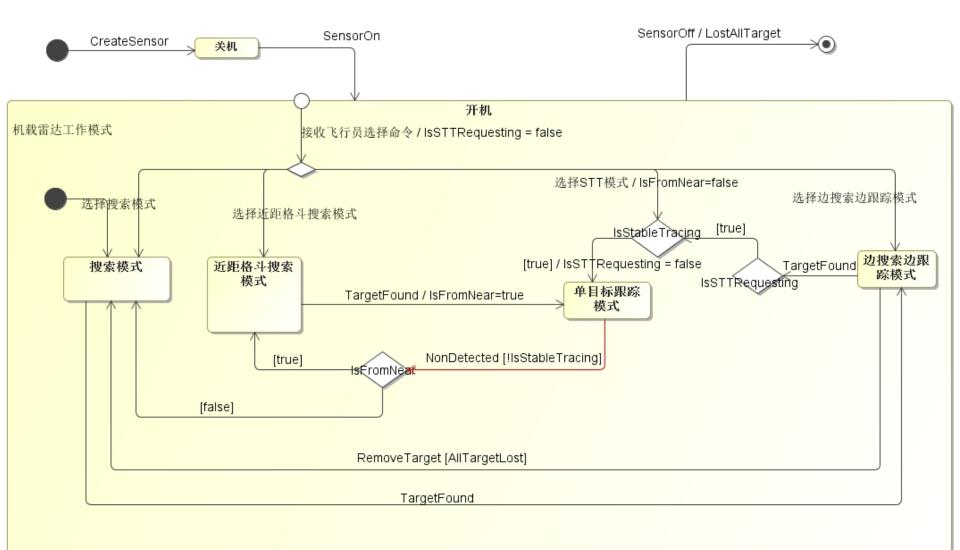
Abstract Statecharts of sensor models



Track report

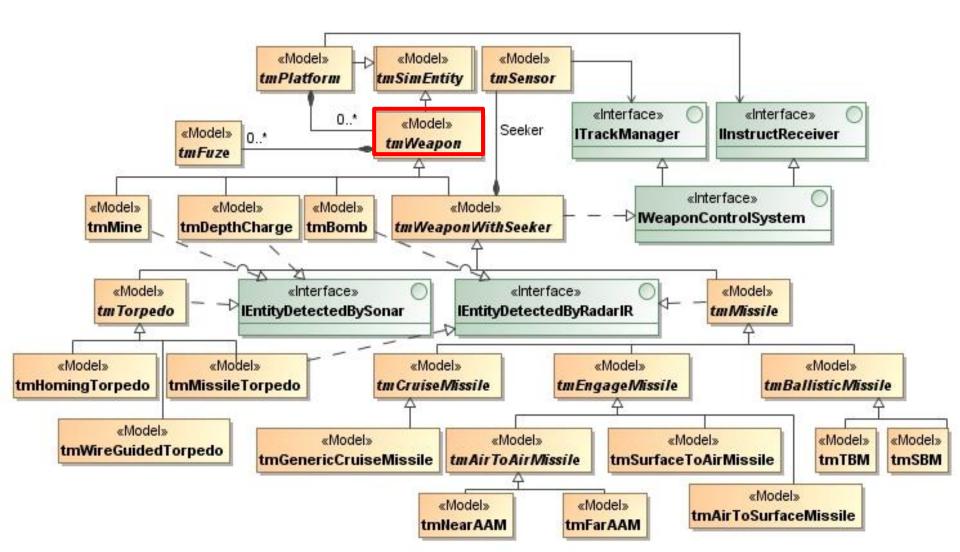


Air fire radar statecharts model

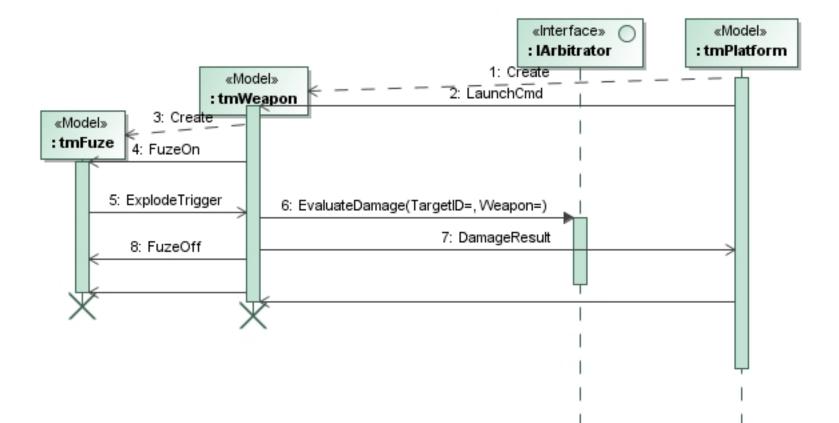


3. Generic model architecture

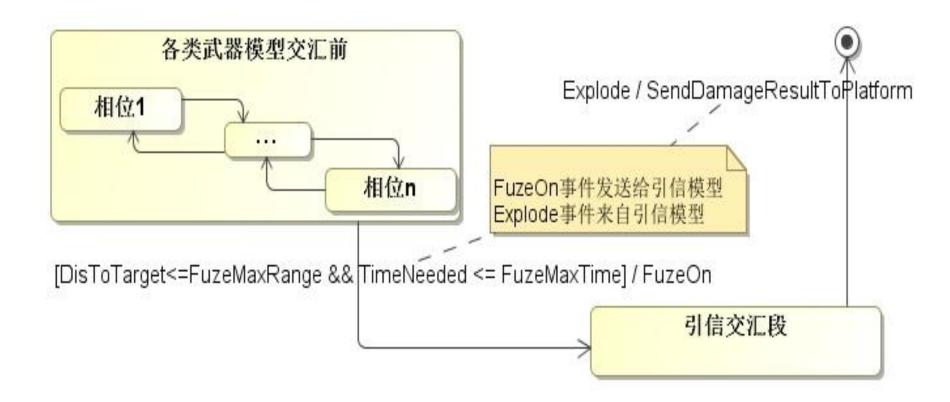
Weapon model architecture



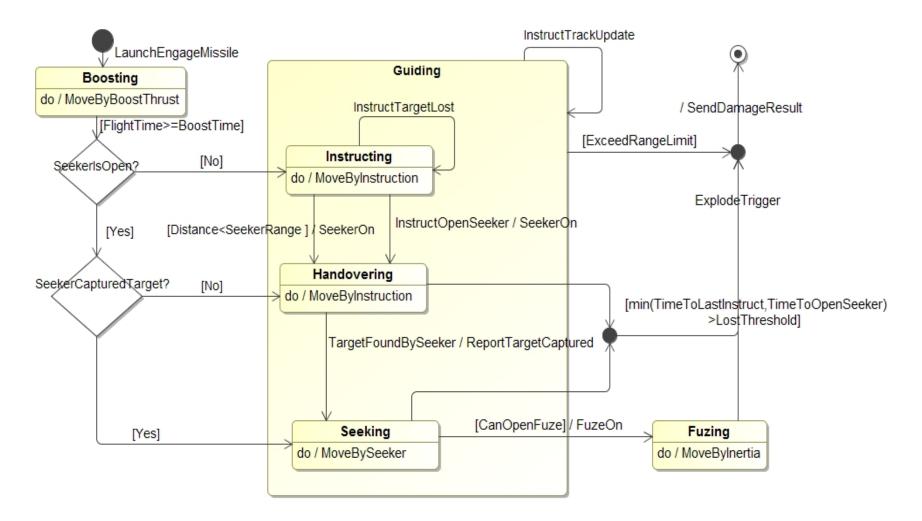
Damage evaluation and report



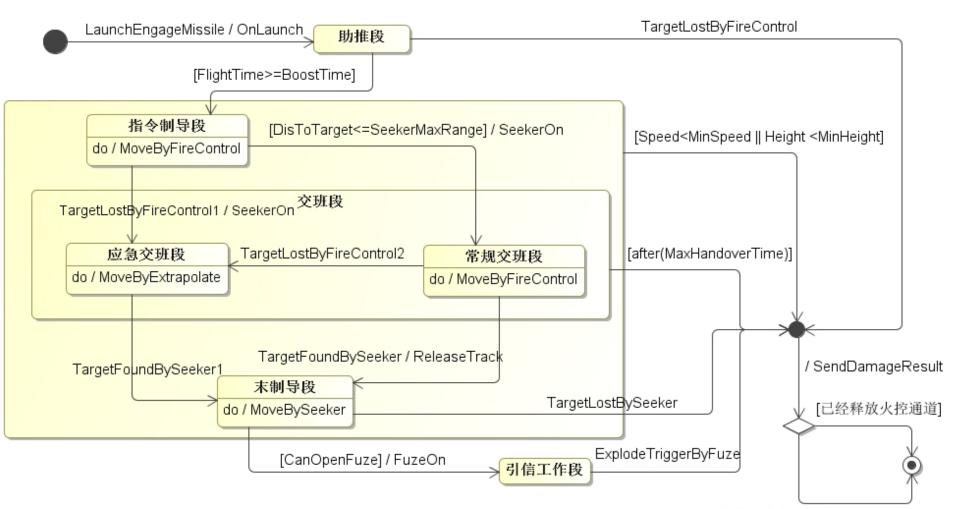
Statecharts-based guided weapon modeling framework



Surface to air missile physical behavior

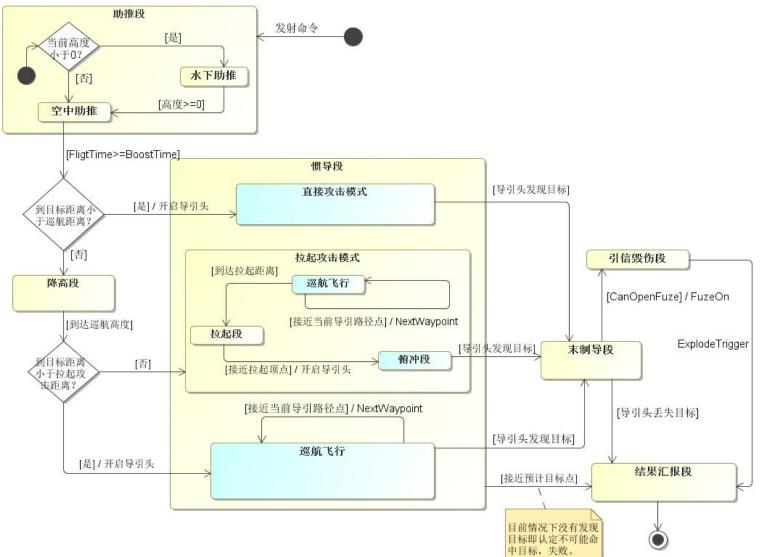


Air to surface missile physical behavior

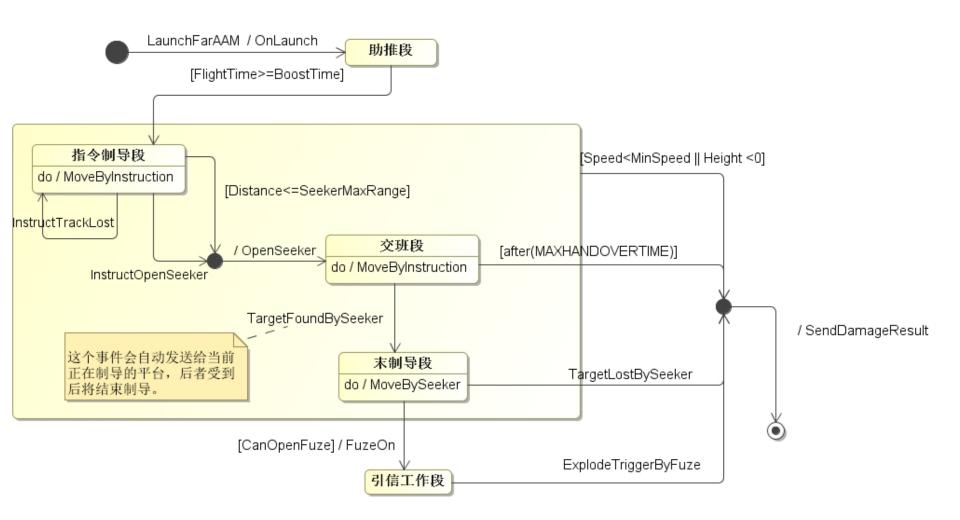


[[]尚未释放火控通道] / ReleaseTrack

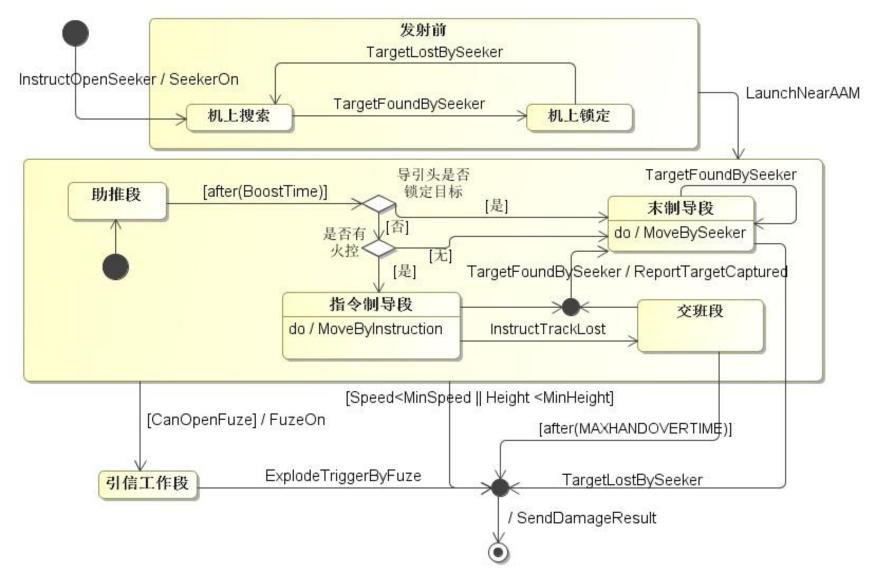
Cruise missile physical behavior



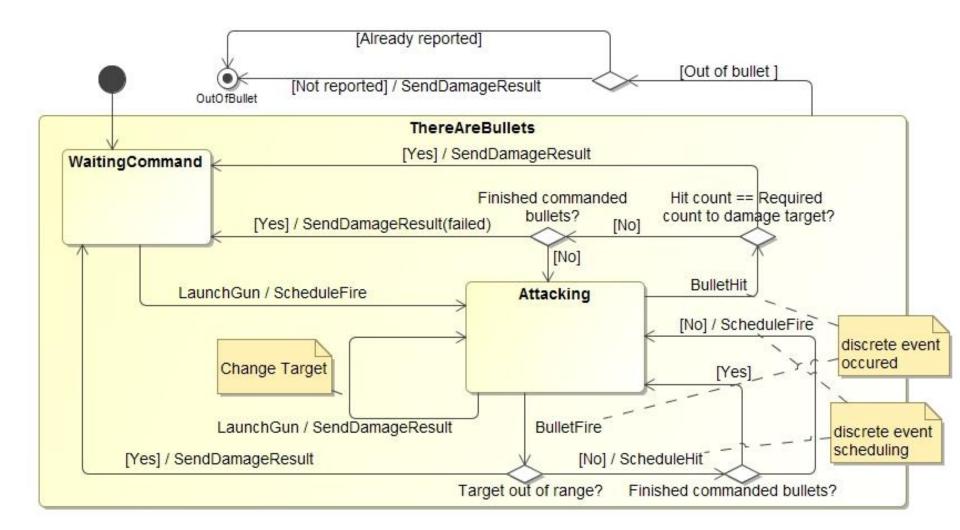
Far air to air missile physical behavior



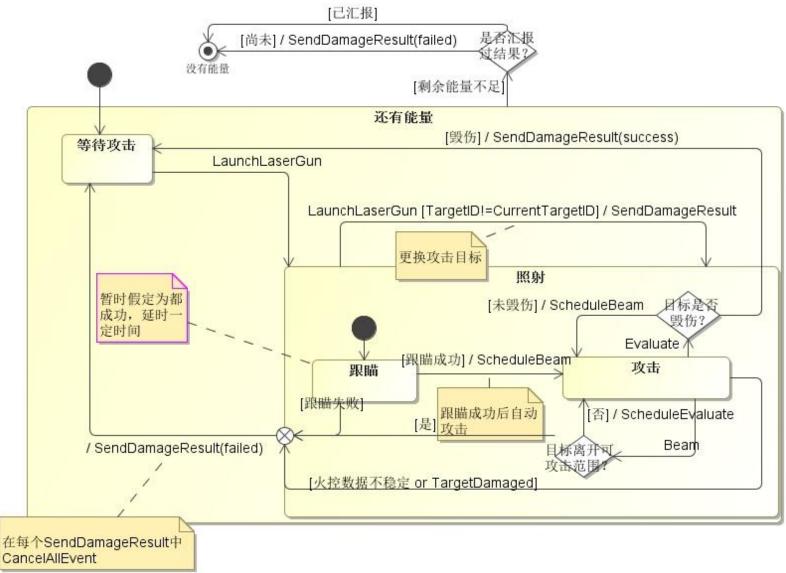
Near air to air missile physical behavior



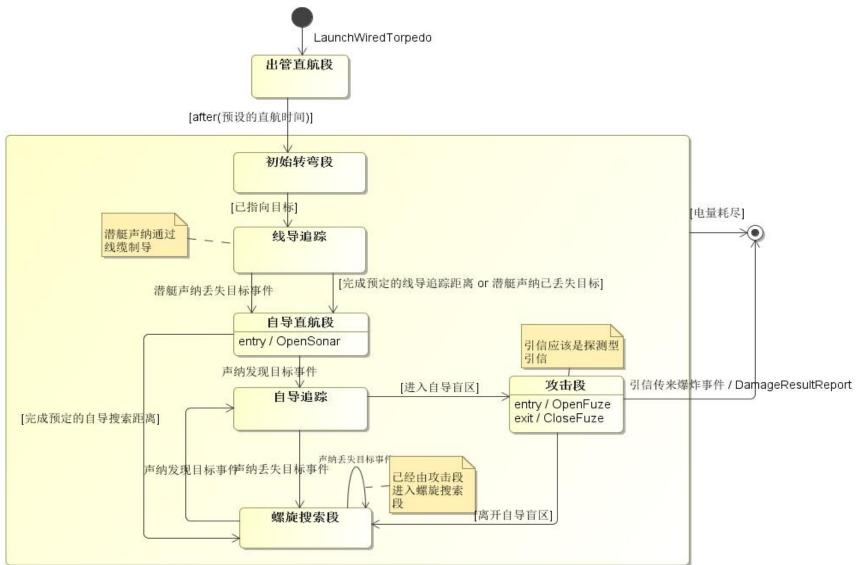
Ballistic gun physical behavior



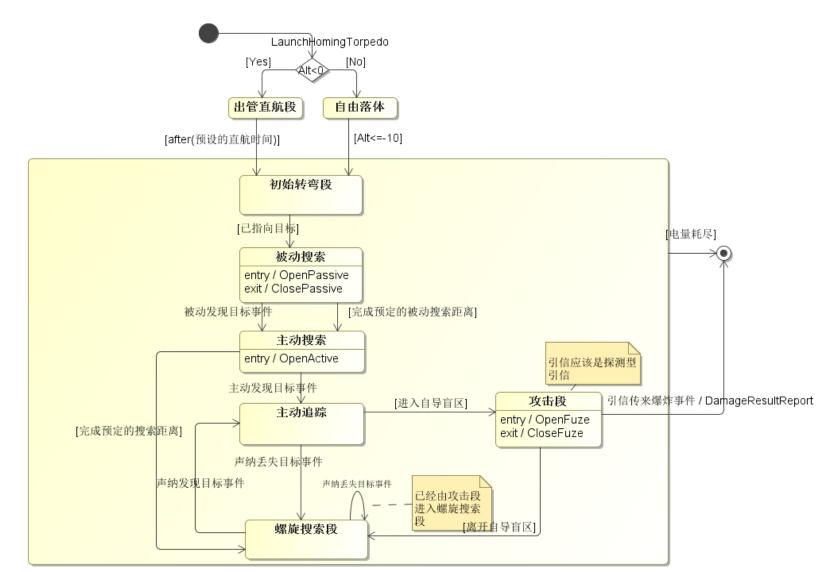
Laser gun physical behavior



Wired torpedo physical behavior

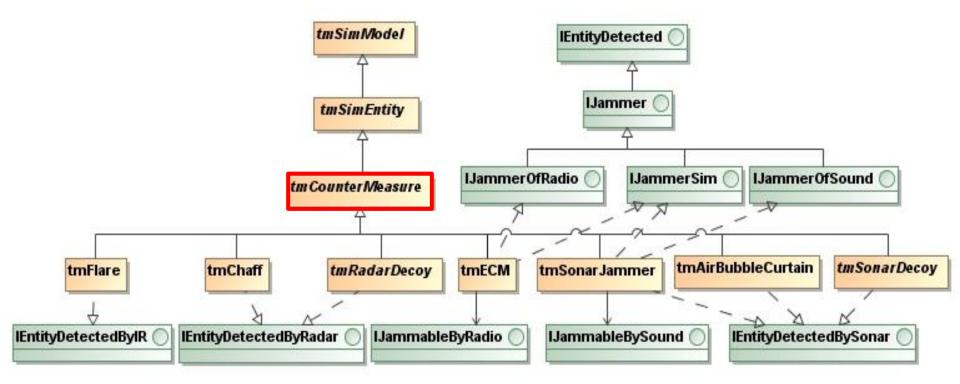


Homing torpedo physical behavior



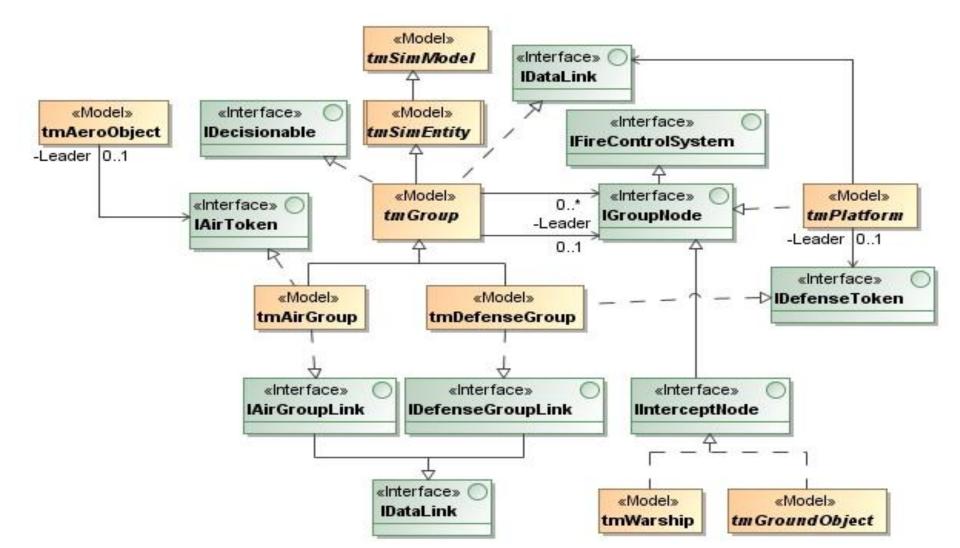
3. Generic model architecture

Countermeasure model architecture



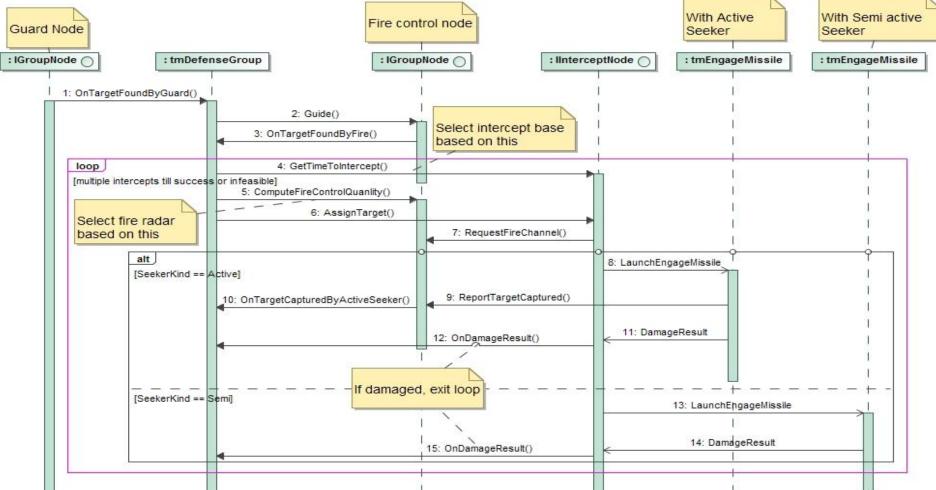
3. Generic model architecture

Group model architecture

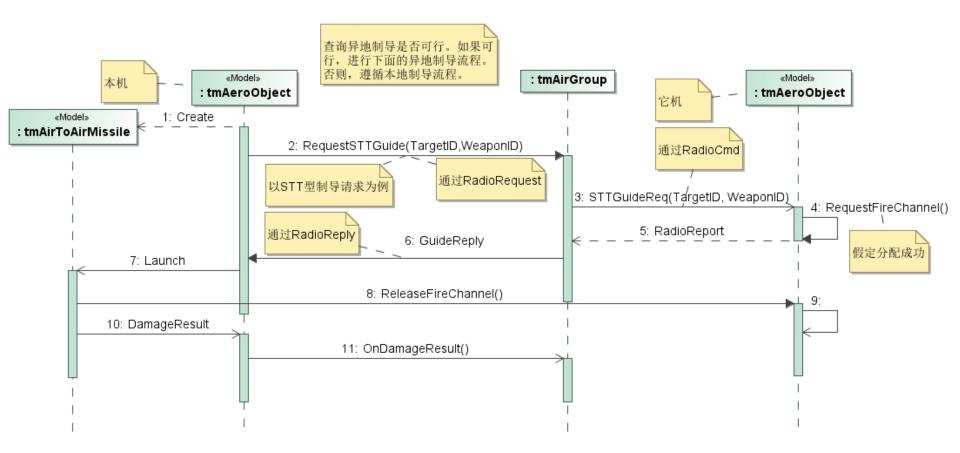


Group model architecture

Sequential diagram-based defense group collaborative behavior



Air group collaborative guiding behavior



SMP based structural model architecture representation

- Represented using simulation model definition language (SMDL) of SMP standard
- Generated from the UML representation partly shown in previous slides by way of the UML profile for SMP.

🚍 🔶 Namespace BaseModel						
🖅 💠 Interface IAirFireControlSystem 🚽						
🖅 💠 Interface IAirFireRadar						
🖅 💠 Interface IArbitrator						
🖅 🔶 Interface ICOP						
🖅 🔶 Interface IDataRecorder						
표 💠 Interface IDecisionable						
Interface IEntityDetected						
Interface IEntityDetectedByIR						
🛓 💠 Interface IEntityDetectedByRadar						
➡ ◆ Interface IEntityDetectedByRadarIR						
Interface IEntityDetectedBySonar						
🖅 💠 Interface IEntityManager 🛓 🔶 Interface IEntitySim						
Interface IEnvironment						
Interface IFireControlSensor						
Interface IInstructReceiver						
🖅 🔶 Interface IJammable						
🗄 🔶 Interface IJammableByRadio						
🗄 🔶 Interface IJammableBySound						
🖅 🔶 Interface IJammer						
🖅 🔶 Interface IJammerOfRadio						
💼 🔶 Interface IJammerOfSound						
💼 💠 Interface IJammerSim						
🖅 💠 Interface IModelManager						
🖅 🔶 Interface IWESS						
🖅 💠 Interface IPrototypeManager						
🖅 🔶 Interface ISchedulable						
🖻 🔶 Interface IScheduleManager						
🖻 🔶 Interface IScriptRunner						
🖮 🔶 Interface ISensorManager						
🖅 💠 Interface ISimModel 🛓 🔶 Interface ITrackManager						
Interface lifackmanager						
Hodel tmWESS						
Model tmSimEntity						
🖅 🔶 Model tmSimModel						
🗄 🔶 Namespace CommonDataType						
∃ ♦ Namespace Group						
🖅 🔶 Namespace Gun						
🖅 🔶 Namespace Interaction						
主 🔶 Namespace Platform						
🗄 🔶 Namespace Sensor						

١

÷

◆

Namespace Weapon

SMP based structural model architecture representation

🛓 🚸 Namespace EventType 표 🔶 Event Type AirFCRadarControl 🗄 🔶 Event Type CloseJammer 🗄 🔶 Event Type DamageResult 🗄 🔶 Event Type ExplodeTrigger 🗄 🔶 Event Type FireBallisticGun 🗄 🔶 Event Type FireLaserGun 표 🔶 Event Type FuzeOff 🗄 🔶 Event Type FuzeOn 🗄 🔶 Event Type LandAero 🗄 🔶 Event Type LandHelo 🗄 🔶 Event Type LaunchAero 🗄 🔶 Event Type LaunchAirBubbleCurtain 🗄 🔶 Event Type LaunchAirToSurfaceMissile 🗄 🔶 Event Type LaunchBomb 😟 🔶 Event Type LaunchChaff 🗄 🔶 Event Type LaunchDepthCharge 😟 🔶 Event Type LaunchECM 표 🔶 Event Type LaunchFarAAM 🗄 🔶 Event Type LaunchFlare 🗄 🔶 Event Type LaunchGenericCruiseMissile 🗄 🔶 Event Type LaunchHelo 🗄 🔶 Event Type LaunchHomingTorpedo 🗄 🔶 Event Type LaunchMine 🗄 🔶 Event Type LaunchMissileTorpedo 🗄 🔶 Event Type LaunchNearAAM 🗄 🔶 Event Type LaunchRadarDecoy 표 🔶 Event Type LaunchSBM 🗄 🔶 Event Type LaunchSonarDecoy 🛓 🔶 Event Type LaunchSonarJammer 🗄 🔶 Event Type LaunchSonobuoy 🗄 🔶 Event Type LaunchSurfaceToAirMissile 표 🔶 Event Type LaunchTBM 표 🔶 Event Type LaunchUAV 🗄 🔶 Event Type LaunchUUV 🗄 🔶 Event Type LaunchWiredTorpedo 🗄 🔶 Event Type LockedByRadar 🗄 🔶 Event Type MissileLaunchedWarning 🗄 🔶 Event Type OpenJammer 표 🔶 Event Type RadioCmd 🗄 🔶 Event Type RadioReply 표 🚸 Event Type RadioReport 🗄 🔶 Event Type RadioRequest 🗄 🔶 Event Type ReleaseByRadar 🗄 🔶 Event Type SensorOff 🗄 🔶 Event Type SensorOn 표 🔶 Event Type VisualTargetFound

🖃 🔶 Namespace Platform 🗄 🔶 Interface IAirbase 🗄 🔶 Model tmPlatform 🖃 🚸 Namespace AirObject 🗄 🔶 Model tmAeroObject 🗄 🔶 Model tmAirObject 🗄 🔶 Model tmHeloObject 🗄 🔶 Model tmUAV Ē 🚸 Namespace GroundObject 표 🔶 Model tmAirDefenseBase 표 🚸 Model tmAirfield 🗄 🔶 Model tmGroundObject 🗄 🧇 Model tmVehicle 🖃 🔶 Namespace SpaceObject 🗄 🔶 Model tmSatellite 표 🔶 Model tmSpaceObject 🖃 🔶 Namespace SubObject 🗄 🔶 Model tmConventionalSubmarine 🗄 🔶 Model tmNuclearSubmarine 🗄 🔶 Model tmSubmarine 표 🔶 Model tmSubObject 💼 🔶 Model tmVVV 🖃 🔶 Namespace SurfaceObject 🗄 🔶 Model tmCarrier 🗄 🔶 Model tmSonobuoy 🗄 🔶 Model tmSurfaceObject 🗄 🧇 Model tmWarship

- 🚍 🔶 Namespace Sensor
 - 🗄 🔶 Model tmSensor
 - 🚊 🔶 Namespace Fuze
 - 🛓 🔶 Model tmContactFuze
 - 🗄 🔶 Model tmFuze
 - 🛓 🔶 Model tmSenseFuze
 - 🗄 🔶 Model tmTimeFuze
 - 🚍 🔶 Namespace Optical
 - 🚊 🔶 Model tmInfraRed
 - 🗄 🔶 Model tmIRST
 - 🗄 🔶 Model tmLaser
 - 😟 🔶 Model tmLaserIlluminator
 - 😟 🔶 Model tmOptical
 - 连 🔶 Model tmSeekerIR
 - 🛓 🔶 Model tmSemiActiveLaserSeeker
 - 🗄 🔶 Model tmVisual
 - 🗄 🔶 Model tmWarningIR
 - 🚍 🔶 Namespace Radar
 - 🗄 🔶 Model tmRadar
 - 🚍 🔶 Namespace ActiveRadar
 - 🕕 🔶 Model tmActiveRadar
 - 🚊 🔶 Model tmActiveRadarSeeker
 - 🚊 🔶 Model tmAirFireRadar
 - 🐵 🔶 Model tmFireControlRadar
 - 🚊 💠 Model tmSurfaceFireRadar
 - 🗄 🔶 Model tmSurveillanceRadar
 - 🖮 💠 Namespace PassiveRadar
 - 😟 🔶 Model tmESM
 - 🗄 🔶 Model tmPassiveRadar
 - 🗄 🔶 Model tmPassiveRadarSeeker
 - 🛓 🔶 Model tmSemiActiveRadarSeeker
 - 🖃 🔶 Namespace Sonar
 - 🚊 🔶 Model tmActiveSonar
 - 표 💠 Model tmActiveSonarSeeker
 - 😟 💠 Model tmActiveTrackSonar
 - 🚊 🔶 Model tmPassiveSonar
 - 🚊 🔶 Model tmPassiveSonarSeeker
 - 🚊 🔶 Model tmPassiveTrackSonar
 - 🕕 🔶 Model tmSonar
 - 🗄 🔶 Model tmSSM

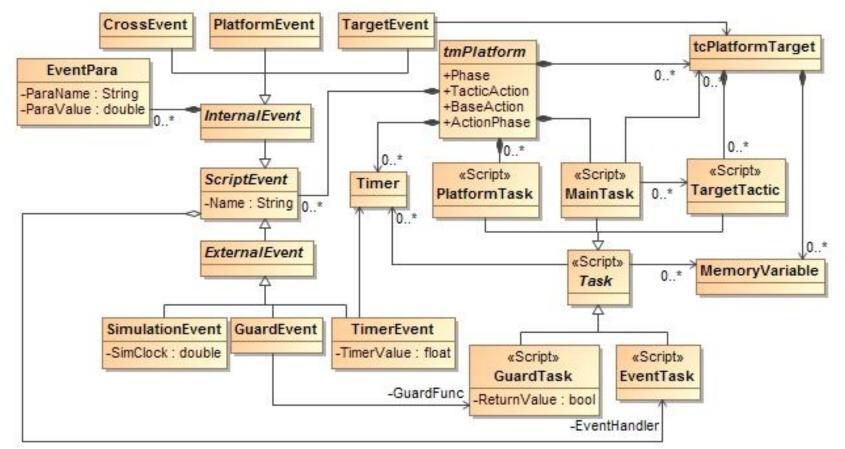
SMP based structural model architecture representation

🖃 🔶 Namespace Weapon 표 🔶 Interface IWeaponControlSystem 🗄 🔶 Model tmBomb 🗄 🔶 Model tmDepthCharge 🗄 🔶 Model tmMine 主 🔶 Model tmWeapon 💠 Model tmWeaponWithSeeker 🚸 Namespace Missile 🗄 🔶 Model tmMissile 😑 🚸 Namespace BallisticMissile 🖮 🚸 Model tmBallisticMissile 🖅 🔶 Model tmSBM 🗄 🔶 Model tmTBM 😑 🔶 Namespace CruiseMissile 💼 🚸 Model tmCruiseMissile 🖅 🚸 Model tmGenericCruiseMissile 🖃 🔶 Namespace EngageMissile 庄 🔶 Model tmAirToAirMissile 💼 🔶 Model tmAirToSurfaceMissile 🗄 🔶 Model tmEngageMissile 표 🔶 Model tmFarAAM 🗄 🔶 Model tmNearAAM 🗄 🔶 Model tmSurfaceToAirMissile 🖃 🔶 Namespace Torpedo 🗄 🔶 Model tmHomingTorpedo 🗄 🧄 🔶 Model tmMissileTorpedo 主 🔶 Model tmTorpedo 庄 🔶 Model tmWireGuidedTorpedo

🖃 🔶 Namespace CounterMeasure 🖮 🚸 Model tmAirBubbleCurtain 🗄 🔶 Model tmChaff 😟 🚸 Model tmCounterMeasure 💼 🔶 Model tmECM 🗄 🔶 Model tmFlare 😟 🚸 Model tmRadarDecov 😟 🔶 Model tmSonarDecov 😟 🚸 Model tmSonarJammer 🖃 🚸 Namespace Group 🗄 🔶 Interface IAirGroupLink 🗄 🔶 Interface IAirToken 🖅 🔶 Interface IDataLink 🗄 🔶 Interface IDefenseGroupLink 💼 🚸 Interface IDefenseToken 표 🔶 Interface IGroupNode 🗄 🔶 Interface IInterceptNode 🖅 🔶 Model tmAirGroup 主 🔶 Model tmDefenseGroup 🗄 🔶 Model tmGroup 🖃 🔶 Namespace Gun 표 🚸 Model tmBallisticGun 🖅 🚸 Model tmGun 🗄 🔶 Model tmLaserGun

Cognitive modeling interface

• Base cognitive behavior metamodel (BCBM)



165 API functions for cognitive behavior modeling

- Parameters query(23)
- Mission & task query(12)
- Situation analysis (16)
- Platform maneuver(8)
- Aircraft close combat maneuver(13)
- Waypoint management(13)
- Fire control(26)

- Sensor control(16)
- Group control(10)
- Simulation control(5)
- State based modeling(8)
- Task based modeling(4)
- Event based modeling(11)

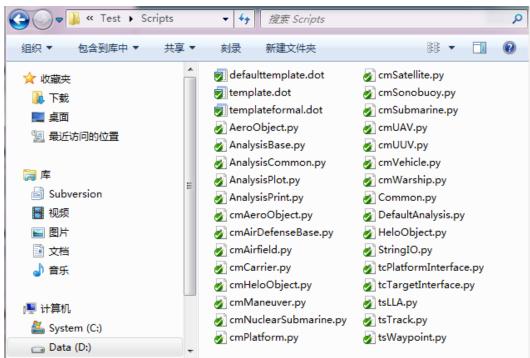
4. a model component library

- One DLL for each concrete model within AMA
- Total 54
 - In platform models
 - 14 weapon models
 - 20 sensor mode
 - ountermeasur
 - 3 group models.

组织 ▼ 包含到库中	•	共享 ▼ 刻录 新建文件夹		
 ▲ ☆ 收藏夹 ↓ 下载 ■ 桌面 	^	⊗ tmActiveRadarSeeker.dll ⊗ tmActiveSonarSeeker.dll ⊗ tmActiveTrackSonar.dll ⊗ tmAeroObject.dll	& tmFarAAM.dll ॐ tmFlare.dll & tmGenericCruiseMissile.dll & tmHeloObject.dll	& tmSide.dll & tmSonarDecoy.dll & tmSonarJammer.dll & tmSonobuoy.dll
5 最近访问的位置 4 篇 库	н	imAirBubbleCurtain.dll	imHomingTorpedo.dll	🚳 tmSSM.dll 🚳 tmSurfaceFireRadar.dll
▷ ➡ Subversion		🚳 tmAirfield.dll 🚳 tmAirFireRadar.dll	⊗ tmLaserGun.dll ⊗ tmMine.dll	⊗ tmSurfaceToAirMissile.dll ⊗ tmSurveillanceIR.dll
▶ 🛃 视频		imAirGroup.dll ⊚tmAirToSurfaceMissile.dll	🚳 tmMissileTorpedo.dll 🚳 tmNearAAM.dll	⊗ tmSurveillanceRadar.dll ⊗ tmTimeFuze.dll
4 🖹 文档		🚳 tmBallisticGun.dll	🚳 tmNuclearSubmarine.dll	🔊 tmVisual.dll
▷ ▶ 我的文档 ▶ ▲ 公用文档		imBomb.dll ⊚ tmCarrier.dll	ঊ tmOCEAN.dll ঊ tmPassiveRadarSeeker.dll	🚳 tmWarningIR.dll 🚳 tmWarship.dll
		⊗ tmChaff.dll ⊗ tmContactFuze.dll		🗟 tmWireGuidedTorpedo.d
▶ 🕌 公用音乐		🚳 tmConventionalSubmarine.dll 🚳 tmDefenseGroup.dll	⊗ tmSatellite.dll ⊚ tmSBM.dll	
▲ 🖳 计算机		🚳 tmDepthCharge.dll 🚳 tmECM.dll	⊗ tmSeekerIR.dll ⊗ tmSemiActiveRadarSeeker.dll	
▷ 🏭 本地磁盘 (C:)	-	🚳 tmESM.dll	🚳 tmSenseFuze.dll	

5. a set of cognitive decision model scripts

- One default script for each combat platform model
- Model-instance separation of cognitive behavior scription
- Each script corresponds to a graphical conceptual mod



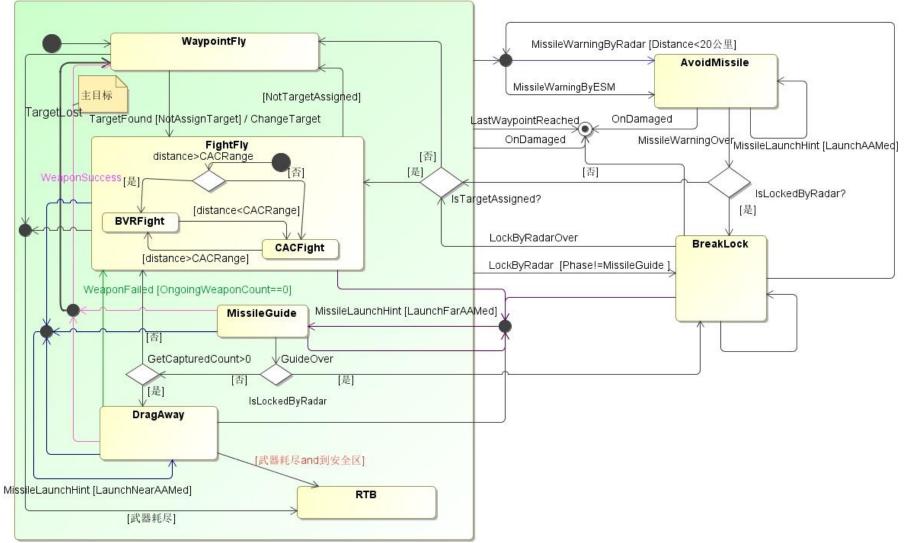
Model-instance separation of cognitive behavior scripts

• Instance specific scripts inherit from the default

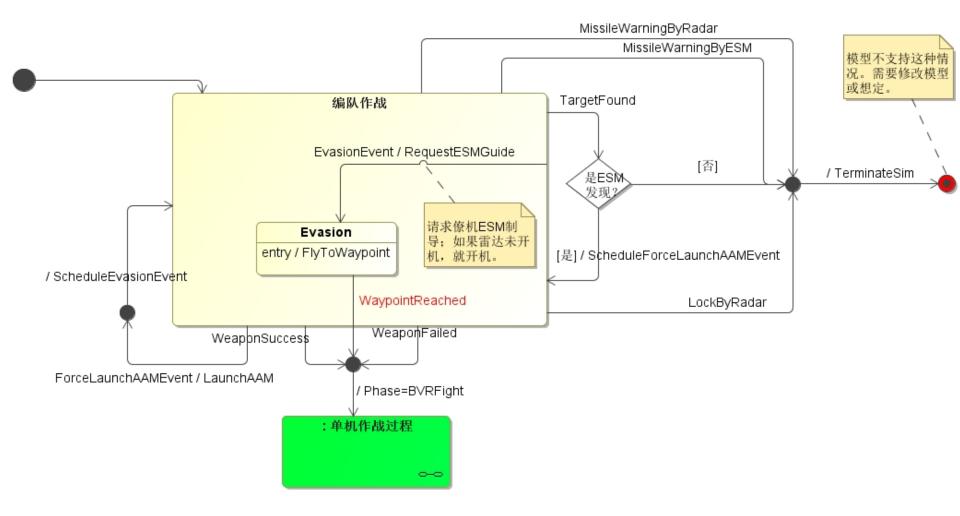
```
cmAeroObject.py
             cmPlatform.py
                         cmHeloObject.py \times
                                        Helo1new.pv
    # -*- coding: cp936 -*-
    from cmPlatform import *
    C DippingSonarWorkingDepth = 30#the actual workign depth of dipping sc
    #statemachine
    P Flying
                              = 1 #fly to next waypoint
                              = 2 #hover and release dipping sonar
    P Hovering Releasing
    P Hovering Waiting
                              = 3 #hover and wait target found event
9
    P Hovering Withdrawing = 4 #hover and withdraw dipping sonar
10
11
12 - class cmHeloObject(cmPlatform):
        #entry point for initialization
13
        def InitDecision(self, PI):
14 -
            PI.SubscribeEvent ("WaypointReached", "WaypointReachedHandler")
15
            PI.SubscribeEvent ("TargetFound", "TargetFoundHandler")
16
17
            PI.AircraftTakeOff()
                                                      cmAeroObject.py
                                                                    cmPlatform.py
                                                                                cmHeloObject.py
                                                                                              Helo1new.py ×
            PI.Phase = P Flying
18
19
                                                         # -*- coding: cp936 -*-
        #entry point for decision
20
                                                         from cmHeloObject import *
        def StepDecision(self, PI):
21 -
                                                      4 Class Helo1new(cmHeloObject):
                                                              def InitDecision(self, PI):
                                                      5 -
                                                                  PI.SubscribeEvent ("WaypointReached", "WaypointReachedHandler")
                                                                  PI.SubscribeEvent("TargetFound", "TargetFoundHandler")
                                                                  PI.Phase = P Flying
                                                     10 - def InitDecision (PlatformInfo):
                                                              Helo1new (PlatformInfo.Name). InitDecision (PlatformInfo)
```

5. a set of cognitive decision model scripts

Each script corresponds to a graphical conceptual model - default aero object



conceptual model of an instance aero object by reuse of the default



欢迎使用WESS效能仿真系统!

Dr. Yonglin Lei(雷永林) 0731-84574538 13874992600 yllei@nudt.edu.cn