75 Fifth Street NW, Suite 312
Atlanta, GA 30308, USA
voice: +1-404-592-6897
web: www.Intercax.com
email: info@Intercax.com

Dirk Zwemer, Intercax LLC

Technote: Applications of MBE to Electronics

Abstract 1
Introduction 2
Building a SysML Model for an Electronic Product 2
Integrating the SysML Model with Other Engineering Applications 7
Embedding the SysML Model in a Larger System-of-Systems Model 10
The Future of MBE for Electronics Applications 13
About the Author 14

Abstract

We explore the proposition that Model-based engineering (MBE) offers a productive approach to the design of modern electronic products, as it has proved in fields such as defense and aerospace. We begin by describing a SysML model for a multifunctional mobile device (phablet). We show how that SysML model can be connected, building a system graph extending over multiple software tools, including PLM, ALM, CAD and simulation. Finally, we illustrate how the mobile device model can be merged into a larger mobile communications network model in order to specify and evaluate its performance in this environment. We use SysML models in MagicDraw and IBM Rational Rhapsody and InterCAX's MBE solution, Syndeia, to demonstrate these concepts.

Introduction

Over the last fifty years, commercial electronics has shown a number of sustained trends:

- Greater product functionality, generally supported by more software content
- Greater product portability, requiring increased mechanical and thermal engineering attention
- Greater product interconnectivity, with expectations of spontaneous organization into networks like the Internet-of-Things

One result has been a greater stress on reliability, safety and resilience, as these networks take on survival-critical roles in transportation, biomedicine and other domains. Despite this, commercial electronics has been slow to recognize Systems Engineering as a discipline in the same way as defense or aerospace.

We believe the electronics field stands to benefit from modern approaches to Model-Based Engineering (MBE). In this Tech Note, we will try to justify this belief by presenting an example of systems modeling, divided into three parts.

- A SysML model of a typical portable electronic product, a “Phablet” (hybrid of cell phone and tablet), demonstrating how relationships between requirements, functions and structure can be captured;
- A demonstration how the SysML model can be connected to other models in other engineering tools, including PLM, ALM and CAD;
- A demonstration of how the product SysML model fits seamlessly into a SysML model of a larger system-of-systems, the multi-standard (GSM/GPRS/LTE) cellular telecommunications domain.

Building a SysML Model for an Electronic Product

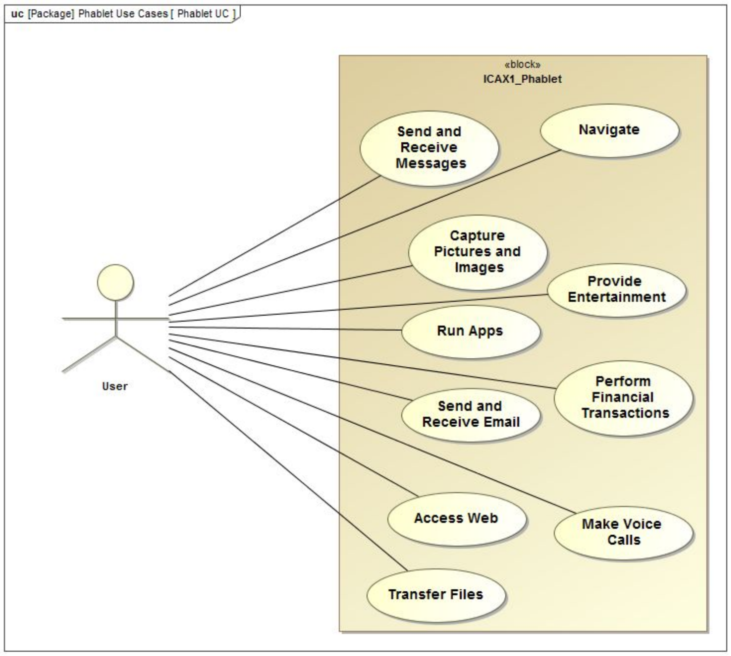

uc [Package] Phablet Use Cases [Phablet UC]
«block»
ICAX1_Phablet
Send and Receive Messages
Navigate
Capture Pictures and Images
Provide Entertainment
Run Apps
Perform Financial Transactions
Send and Receive Email
Access Web
Make Voice Calls
Transfer Files
User

Figure 1 Phablet Use Cases

The design of a consumer electronic product typically begins with a concept of operations, starting with a definition of use cases. In SysML, we can capture and display those use cases in a Use Case diagram, as shown in Figure 1. Our phablet has a wide variety of use cases, like many smart communications devices (on this web page where this tech note is published, the reader can also download the SysML model discussed in both MagicDraw and IBM Rational Rhapsody format; the figures accompanying this text use the MagicDraw versions).

Based on the high-level functional objectives captured in the use cases and a range of other technical, market and regulatory considerations, systems engineers will develop a set of product requirements. Figure 2 shows a partial list of these, each of which would include an id number, a text description, and potentially a range of additional attributes as required. These can be created and managed in the SysML tool or (as we will show in the next part) created in a specialized requirements management tools and transformed into a SysML model.

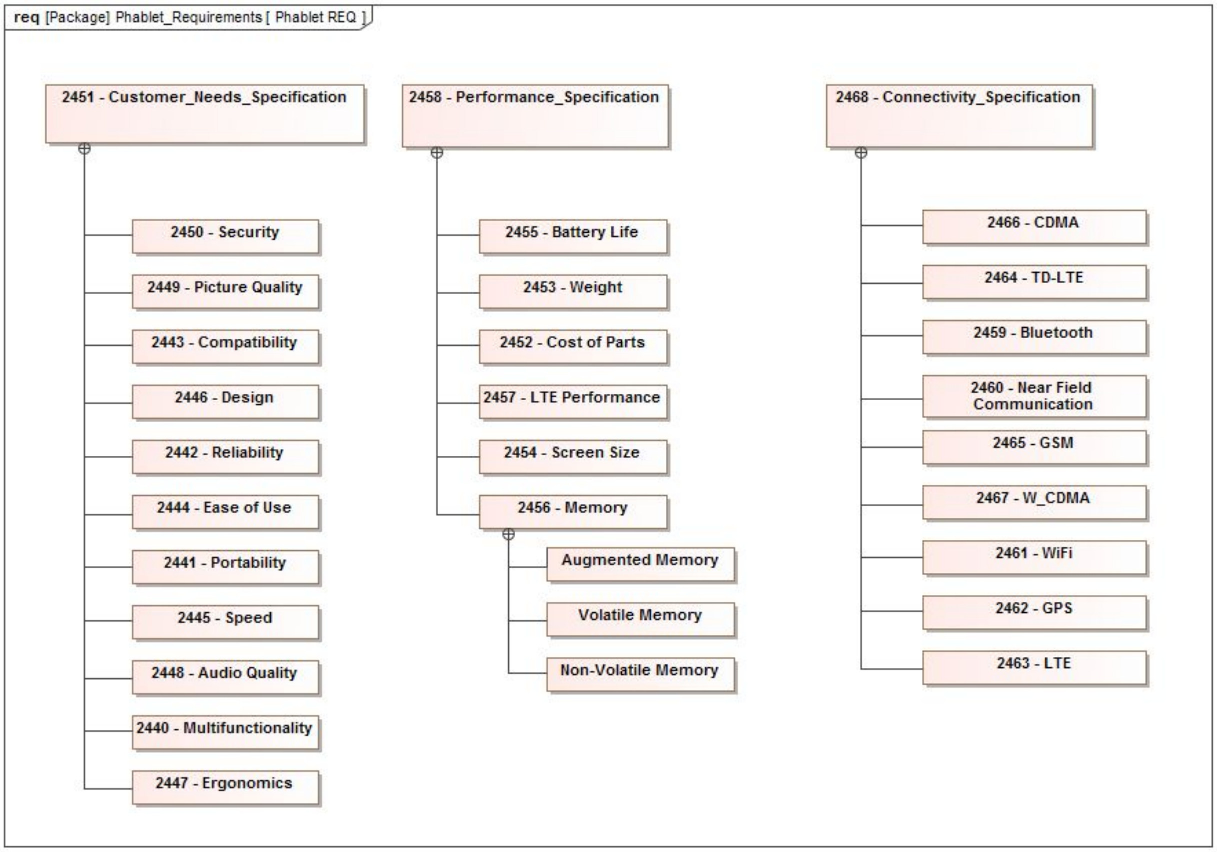


Figure 2 Phablet requirements specifications

This concept-of-operations phase may also include description of desired operational scenarios between user and system, as illustrated in the SysML sequence diagram in Figure 3. This scenario incorporates a number of options, depending on user choice; the user can make voice calls, message, web surf, etc.

The messages from user to phablet are generally Call or Signal messages, triggering internal software services inside the phablet when received. These internal behaviors of the product itself can be captured in a state machine, as shown in Figure 4, where the same calls and signals are shown as triggering transitions between operating states. Inside the state symbols are lists of more detailed behaviors, which can be further decomposed and specified.

As the system behavior description matures, the functional structure can be developed. Figure 5 shows a partial structural decomposition (our example of phablet structure is adapted from an analysis of a real commercial product available at http://www.techinsights.com/teardown/Samsung_SM-Z130H-DS_Sample-Report.pdf).

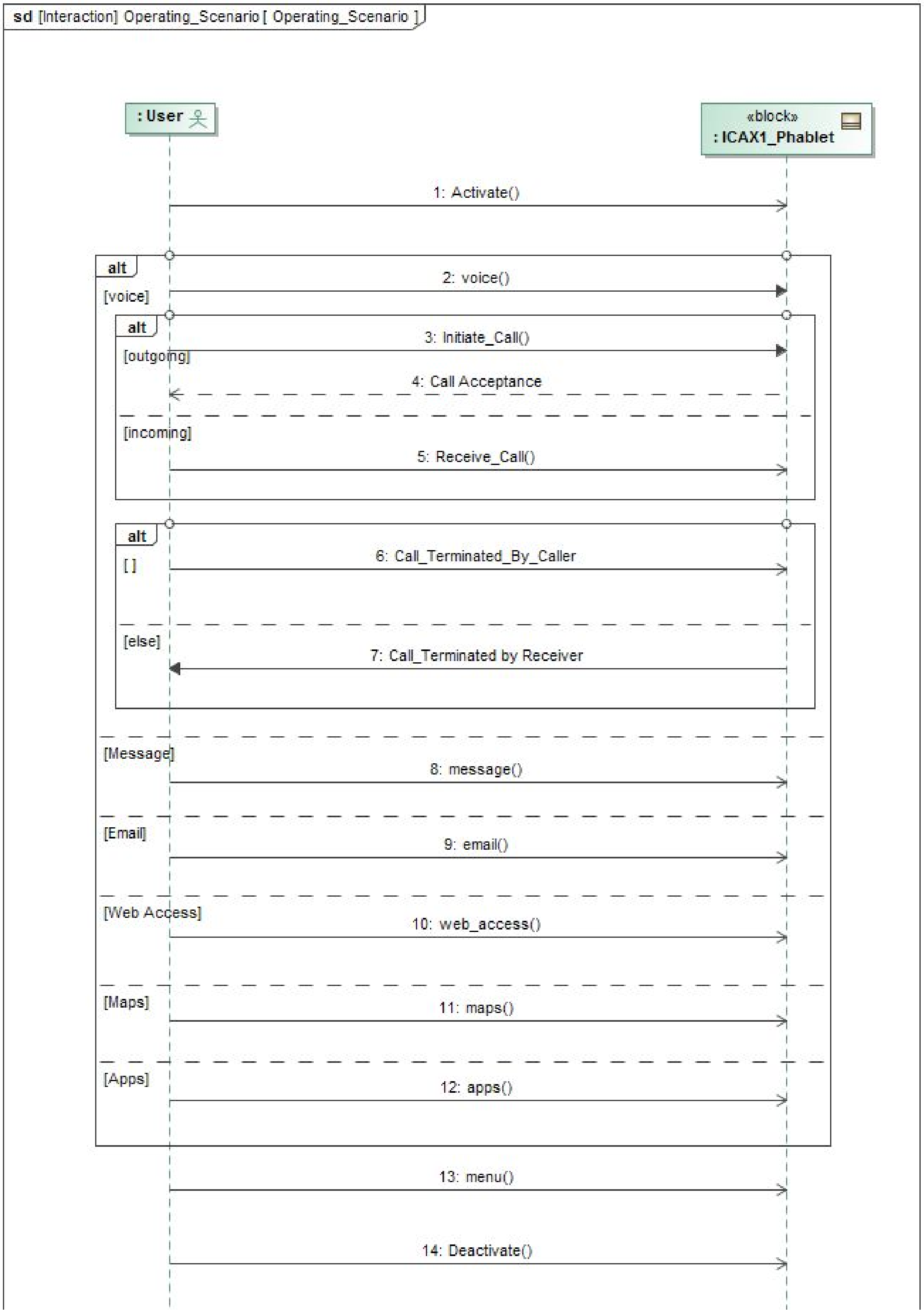


Figure 3 SysML sequence diagram for phablet operating scenario

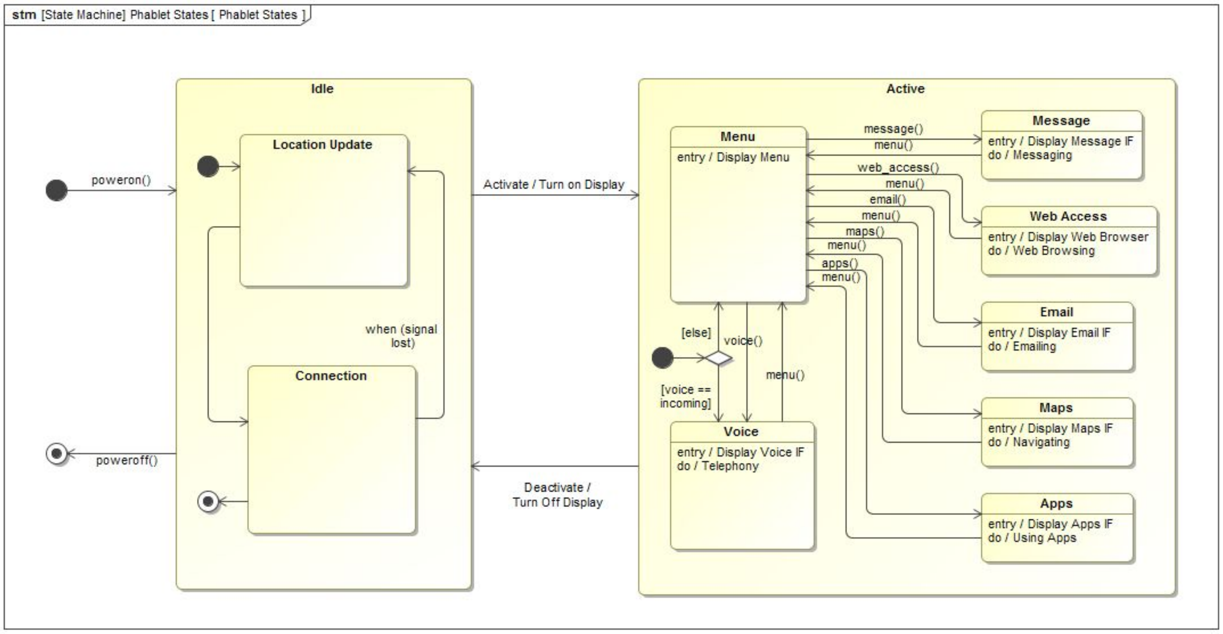
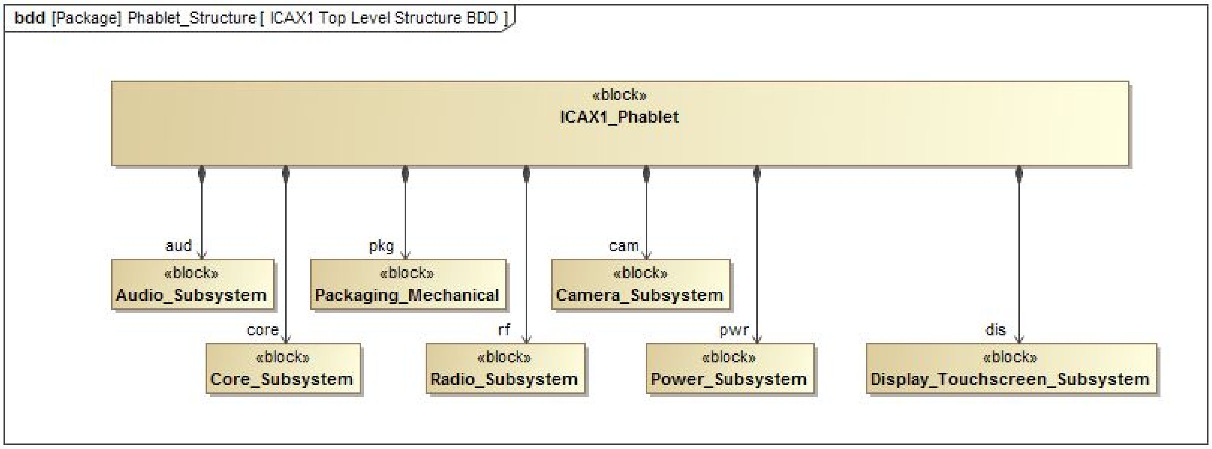


Figure 4 Phablet state machine diagram



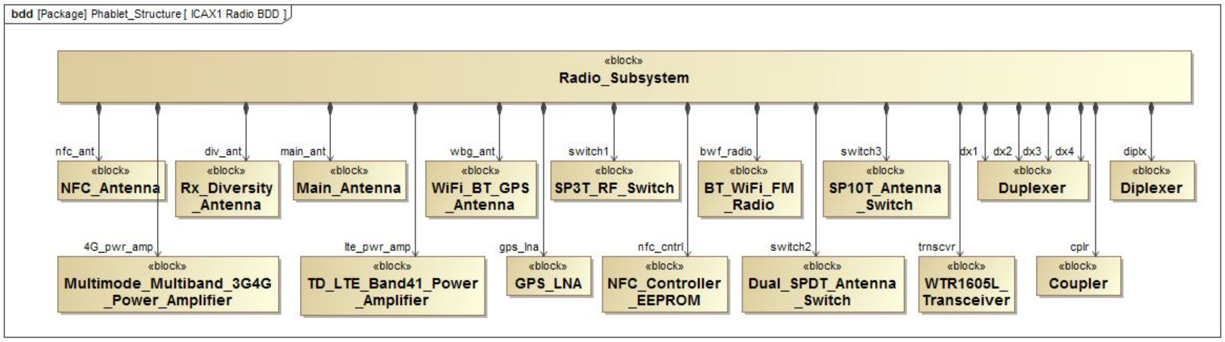


Figure 5 Block definition diagrams showing the top-level structure and a breakdown on one subsystem

A SysML internal block diagram, as shown in Figure 6, shows how the signals are routed between components, which must differentiate between a wide range of frequencies and formats.

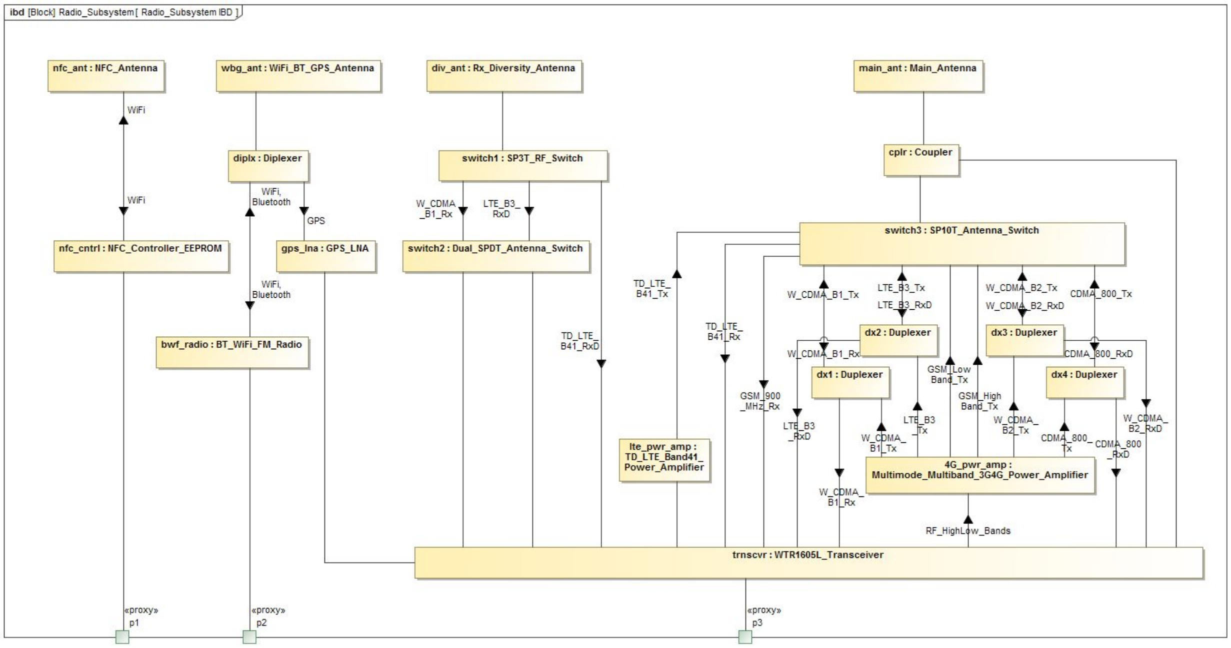
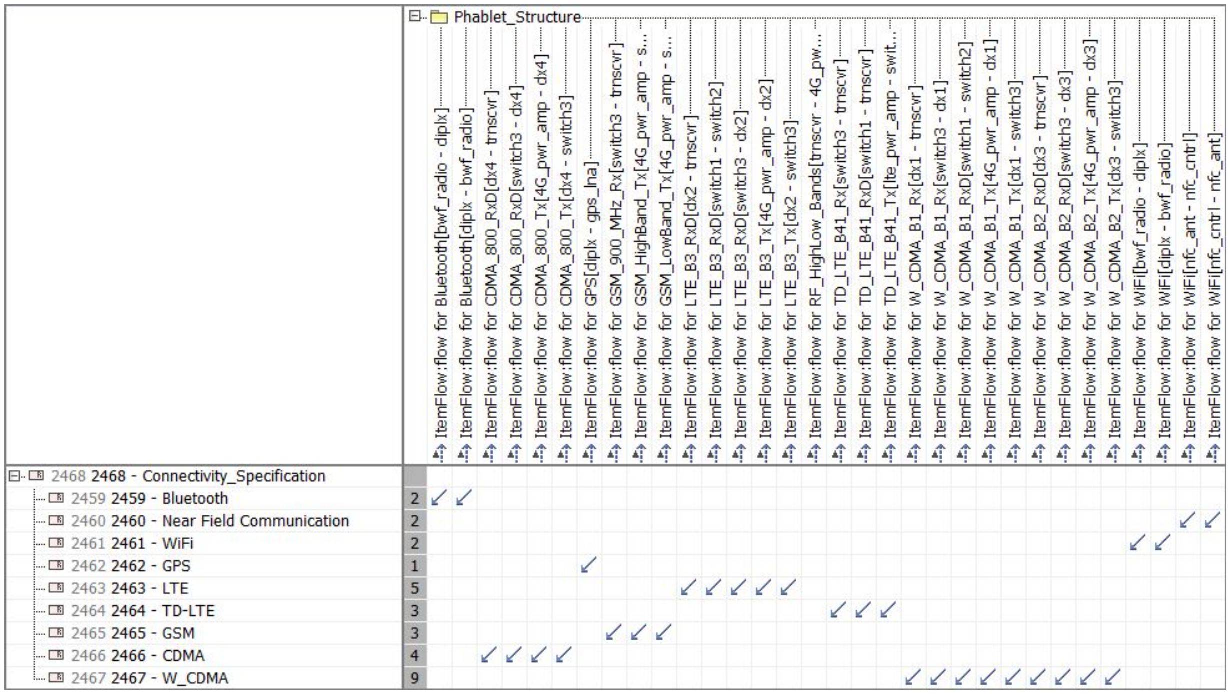


Figure 6 SysML internal block diagram for the phablet radio subsystem, showing the different RF signals and components

Ultimately, we come full circle and connect the structure back to the requirements in Figure 7, which displays <<satisfy>> relations between the Connectivity Specification requirements and the item flows in Figure 6. All connectivity types are represented by at least one item flow in the radio subsystem.



		Phablet_Structure																															
		ItemFlow:flow for Bluetooth[bwf_radio - diplx]	ItemFlow:flow for Bluetooth[diplx - bwf_radio]	ItemFlow:flow for CDMA_800_RxD[dx4 - trnscvr]	ItemFlow:flow for CDMA_800_RxD[switch3 - dx4]	ItemFlow:flow for CDMA_800_Tx[4G_pwr_amp - dx4]	ItemFlow:flow for CDMA_800_Tx[dx4 - switch3]	ItemFlow:flow for GPS[diplx - gps_lna]	ItemFlow:flow for GSM_900_MHz_Rx[switch3 - trnscvr]	ItemFlow:flow for GSM_HighBand_Tx[4G_pwr_amp - s...	ItemFlow:flow for GSM_LowBand_Tx[4G_pwr_amp - s...	ItemFlow:flow for LTE_B3_RxD[dx2 - trnscvr]	ItemFlow:flow for LTE_B3_RxD[switch1 - switch2]	ItemFlow:flow for LTE_B3_RxD[switch3 - dx2]	ItemFlow:flow for LTE_B3_Tx[4G_pwr_amp - dx2]	ItemFlow:flow for LTE_B3_Tx[dx2 - switch3]	ItemFlow:flow for RF_HighLow_Bands[trnscvr - 4G_pw...	ItemFlow:flow for TD_LTE_B41_Rx[switch3 - trnscvr]	ItemFlow:flow for TD_LTE_B41_RxD[switch1 - trnscvr]	ItemFlow:flow for TD_LTE_B41_Tx[lte_pwr_amp - swit...	ItemFlow:flow for W_CDMA_B1_Rx[dx1 - trnscvr]	ItemFlow:flow for W_CDMA_B1_Rx[switch3 - dx1]	ItemFlow:flow for W_CDMA_B1_RxD[switch1 - switch2]	ItemFlow:flow for W_CDMA_B1_Tx[4G_pwr_amp - dx1]	ItemFlow:flow for W_CDMA_B1_Tx[dx1 - switch3]	ItemFlow:flow for W_CDMA_B2_RxD[dx3 - trnscvr]	ItemFlow:flow for W_CDMA_B2_RxD[switch3 - dx3]	ItemFlow:flow for W_CDMA_B2_Tx[4G_pwr_amp - dx3]	ItemFlow:flow for W_CDMA_B2_Tx[dx3 - switch3]	ItemFlow:flow for WiFi[bwf_radio - diplx]	ItemFlow:flow for WiFi[diplx - bwf_radio]	ItemFlow:flow for WiFi[nfc_ant - nfc_cntrl]	ItemFlow:flow for WiFi[nfc_cntrl - nfc_ant]
2468 2468 - Connectivity_Specification																																	
2459 2459 - Bluetooth	2	↙	↙																														
2460 2460 - Near Field Communication	2																															↙	↙
2461 2461 - WiFi	2																													↙	↙		
2462 2462 - GPS	1							↙																									
2463 2463 - LTE	5											↙	↙	↙	↙	↙																	
2464 2464 - TD-LTE	3																	↙	↙	↙													
2465 2465 - GSM	3								↙	↙	↙																						
2466 2466 - CDMA	4			↙	↙	↙	↙																										
2467 2467 - W_CDMA	9																				↙	↙	↙	↙	↙	↙	↙	↙	↙				

Figure 7 Satisfy matrix between connectivity requirement and signal flows in the radio subsystem.

The phablet models provided, and our presentation of them here, are incomplete and are meant to serve as illustrations of the MBSE approach to high-level design of an electronic product. One important area, simulation and analysis, has not been treated, but it would be relatively simple to build roll-up calculations of cost, mass or power consumption into the product structure and be able to keep running tallies of these quantities as the design evolves.

Integrating the SysML Model with Other Engineering Applications

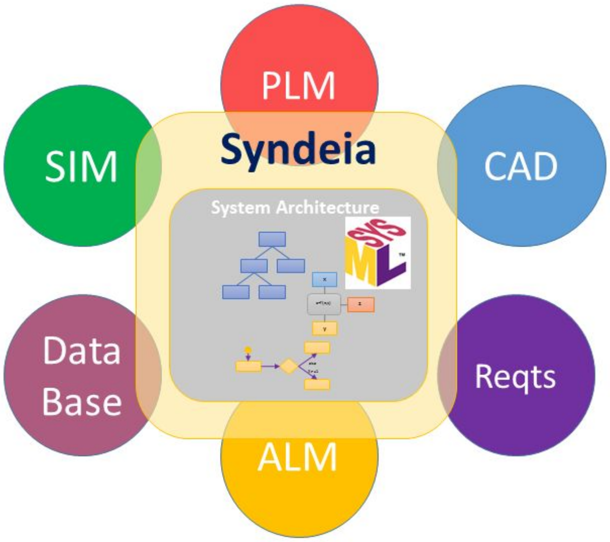


Figure 8 Syndeia architecture

A SysML model can be a valuable first step in the conceptual design of a new product, but it is not intended to replace the detailed design and analysis tools or to store all product data. As the engineering process continues, many additional tools and repositories come into play, including CAD, PLM, ALM, simulation and project management, typically from multiple vendors.

The problem arises that the product models in these separate tools will become inconsistent over time and errors will creep into the design. Powerful dynamic mechanisms are required to mitigate this, leading to the development of the Model-Based Engineering (MBE) concept. Note that MBE does not mean the use of models in engineering; it implies a single product model distributed over multiple tools and repositories.

Syndeia™, from Intercax, is a platform for the practice of MBE. As shown in Figure 8, it uses a system architecture model as central hub and connects SysML elements to elements in PLM, CAD, ALM (application lifecycle management), project management, requirements, simulation and other engineering tools. SysML is an effective medium for building a high-level roadmap of the system because it provides a rich language for connecting structure, behavior, requirements and analysis concepts, concepts that can map to corresponding concepts in more specialized tools. Currently, Syndeia supports two SysML modeling tools, MagicDraw (No Magic Inc.) and IBM Rational Rhapsody, and a variety of other project management system, where status, recent activity, schedule and personnel are tracked.

In each case, the elements remain connected, linking the models in different tools into a single system model, the ultimate objective of MBE. Depending on the nature of the connection (and the capability of the tools involved), these connections can support use cases of practical value to engineering, including

- Compare and synchronize SysML block structure with PLM bill-of-materials or Simulink block structure as they evolve over time
- Update SysML requirements from a master requirements repository
- Open and view CAD models or project management issue trackers directly from the system model
- Import CAD parameters from design files into the system model for requirements verification
- View the latest file versions in system software in GitHub

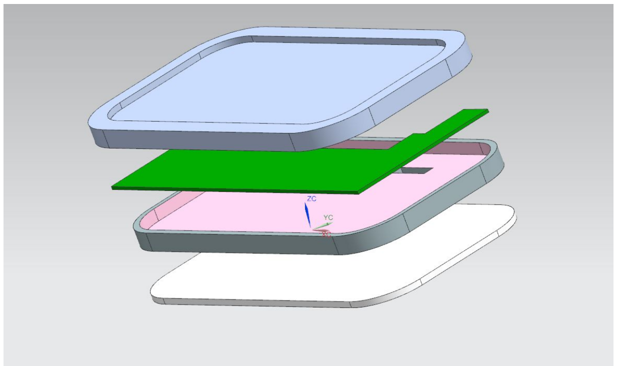
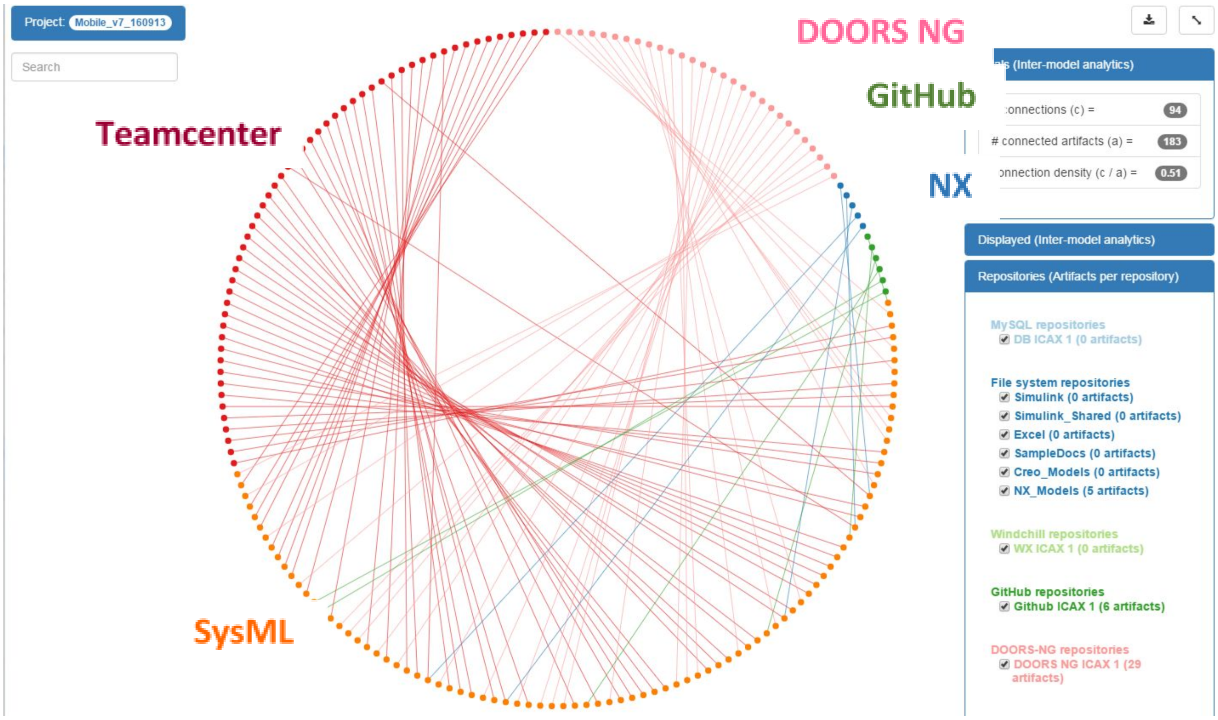


Figure 9 CAD model, main phablet mechanical components

For example, the CAD models of the phablet mechanical packaging components, shown in Figure 9, can be opened directly from the corresponding blocks in the SysML model, once Syndeia has created the connection. If the CAD model is parameterized, the parameter values can be updated between CAD and SysML models when they change, allowing periodic reverification of requirements as the design evolves.

As the number of connections grows, our ability to understand the scope and complexity of the total system model and to identify extended linkages between elements diminishes. We need to be able to view the system model globally and to trace connections locally. Syndeia 3.0 offers the electronics industry both views.

Figure 10 shows a global view of the inter-model connections in a chord plot. The upper right box reports that 183 elements are connected by 94 Syndeia-managed connections. The elements on the periphery of the circle represent SysML (orange), Siemens Teamcenter (green), NX (blue), GitHub (green) and DOORS NG (pink) elements. In our case, system requirements were imported from DOORS at the beginning of the project and the Teamcenter PLM bill-of-materials was generated from the SysML block structure when the functional design was complete. In both cases, changes at one end of the connection may be updated to the other. GitHub and NX connections were created manually between pre-existing elements.



…ls (Inter-model analytics)

…connections (c) = 94

connected artifacts (a) = 183

…nnection density (c / a) = 0.51

Displayed (Inter-model analytics)

Repositories (Artifacts per repository)

MySQL repositories
☑ DB ICAX 1 (0 artifacts)

File system repositories
☑ Simulink (0 artifacts)
☑ Simulink_Shared (0 artifacts)
☑ Excel (0 artifacts)
☑ SampleDocs (0 artifacts)
☑ Creo_Models (0 artifacts)
☑ NX_Models (5 artifacts)

Windchill repositories
☑ WX ICAX 1 (0 artifacts)

GitHub repositories
☑ Github ICAX 1 (6 artifacts)

DOORS-NG repositories
☑ DOORS NG ICAX 1 (29 artifacts)

Figure 10 Chord plot of Inter-model Connections – Global Visualization

The nodes and the lines between them can be interactively highlighted and identified in the light blue box on the upper right, and the entire plot can be searched and filtered. Using this, the system engineer can see at a glance the types of elements being connected and the density of connections in the current state of the model.

While the global view in Figure 10 gives the big picture, many use cases involve tracing extended linkages over both inter-model and intra-model chains. Alternate graphing techniques can make this task simpler. One example is shown in Figure 11, a tree plot initiated from a specific model element in the SysML model or the Syndeia dashboard. Launched from a SysML block, Core Subsystem, it immediately displays six nearest neighbors, a Teamcenter item (shown in red) connected by inter-model connection and five SysML blocks (shown in orange) acting as parts of the Core Subsystem, connected by intra-model connections. One of these last nodes, the Multichip Module, can be further expanded by clicking on it, showing an additional five next-nearest neighbors and the process can be continued selectively until the chain is terminated. In the resulting plot, the extended linkage between the DOORS requirement at the far right, the SysML function it impacts is traced over three connections across two tools. Consider the value of these approaches in expediting design reviews, with explicit connections between system elements and the ability to open those elements in their native tools from a central high-level roadmap of the system.

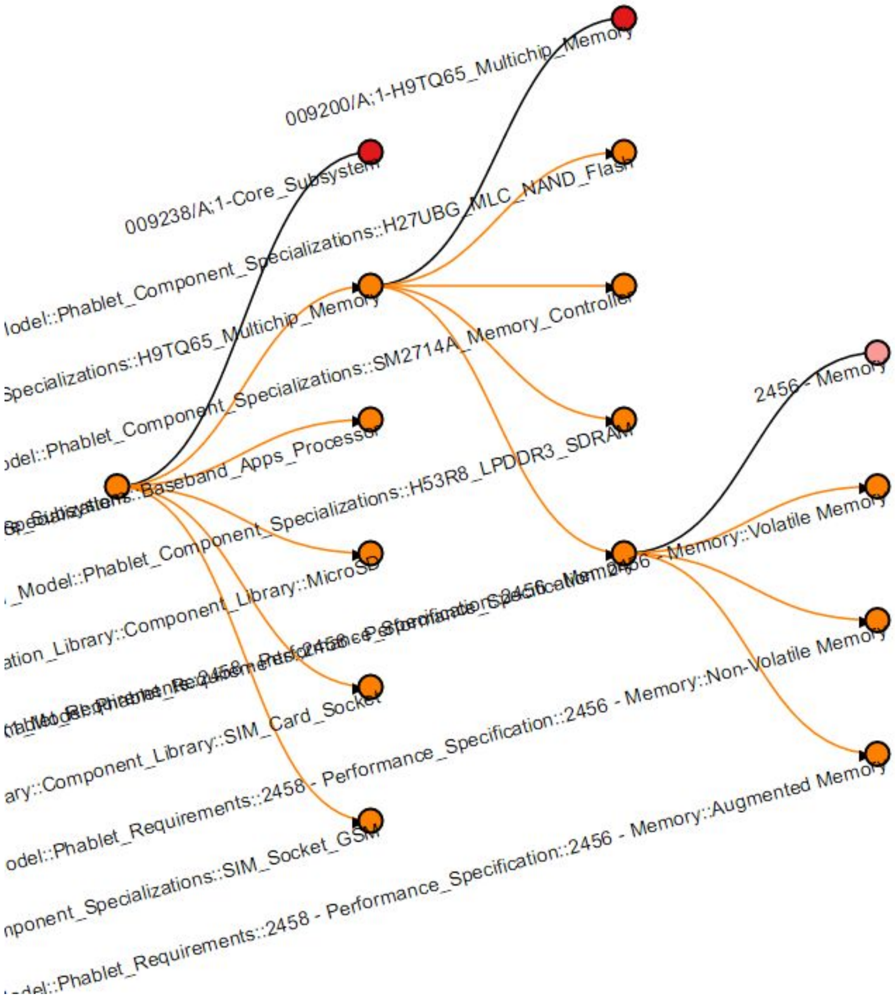

009200/A;1-H9TQ65_Multichip_Memory
009238/A;1-Core_Subsystem
lodel::Phablet_Component_Specializations::H27UBG_MLC_NAND_Flash
Specializations::H9TQ65_Multichip_Memory
odel::Phablet_Component_Specializations::SM2714A_Memory_Controller
Baseband_Apps_Processor
_Model::Phablet_Component_Specializations::H53R8_LPDDR3_SDRAM
2456 - Memory
Memory::Volatile Memory
ation_Library::Component_Library::MicroSD
ary::Component_Library::SIM_Card_Socket
odel::Phablet_Requirements::2458 - Performance_Specification::2456 - Memory::Non-Volatile Memory
nponent_Specializations::SIM_Socket_GSM
odel::Phablet_Requirements::2458 - Performance_Specification::2456 - Memory::Augmented Memory

Figure 11 Tree plot of Inter-model and Intra-model Connections demonstrating traceability across graph

Analysis is a critical part of MBE and should be incorporated into the system model as early as possible

- To evaluate possible system variants
- To explore system risk
- To verify requirements on an ongoing basis

One MBE approach supported by Syndeia allows parts of the system architecture model, for example, the radio subsystem shown in Figure 6, to be transformed into a Simulink block structure, part of which is shown in Figure 12. The block structure, ports and connectors are carried over into Simulink (The Mathworks, Inc.), where the Simulink analyst adds MATLAB code and internal structure as needed to create an executable simulation. Syndeia allows the top-level architecture of the two models to be compared and, as needed, updated, as the SysML and Simulink models evolve.

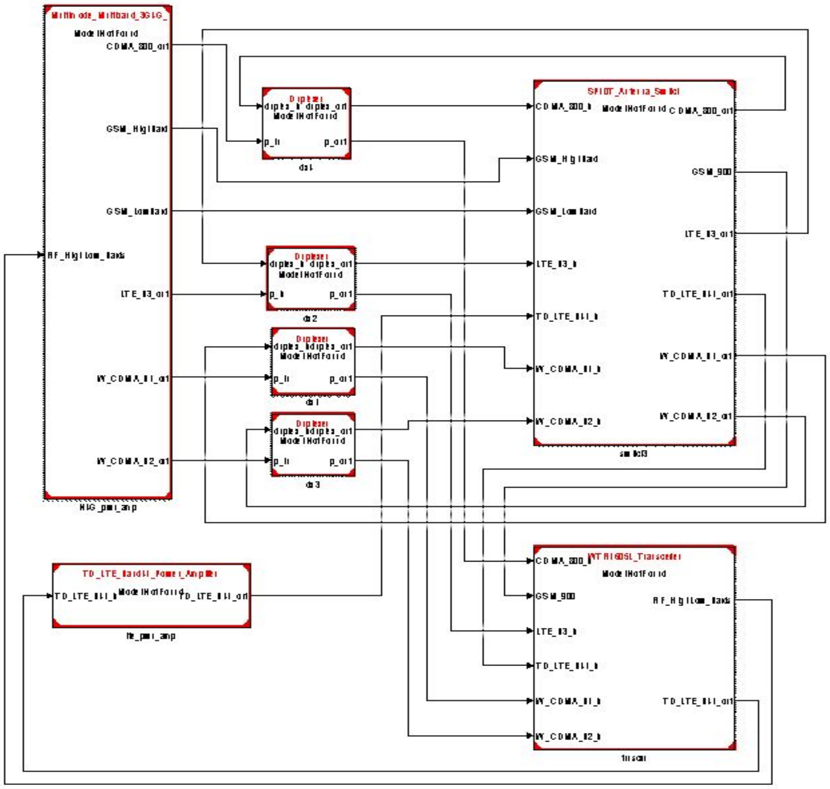


Figure 12 Simulink model of a portion of the radio subsystem model, created from the SysML model by Syndeia

Embedding the SysML Model in a Larger System-of-Systems Model

While the phablet has some off-line capabilities, e.g. playing stored videos, most of its functionality requires connection with cellular, wi-fi or other external networks. A significant benefit of the object-oriented general purpose SysML modeling language is that models can be merged easily in order to evaluate systems-of-systems.

As an example, we have provided an initial model of the cellular telephone domain and included our complete phablet model as a specialization of the Mobile Station block. As our phablet is designed to send and receive in many countries, it is compatible with a number of different frequencies and cellular communications standards (e.g. GSM, GPRS and LTE, corresponding roughly to 1G, 2G and 3G networks). Our SysML model, shown at the top level in Figure 13, incorporates all three of these standards (the author is indebted to the book *From GSM to LTE*, M. Sauter, Wiley, 2011 for its lucid explanation of these standards).

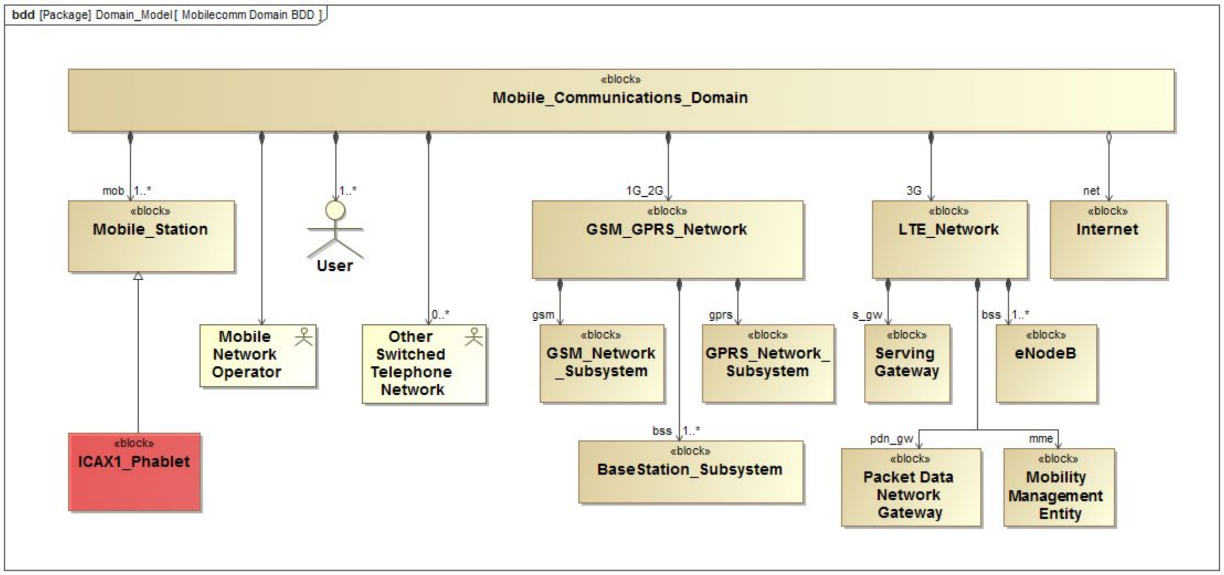


Figure 13 Using the phablet model (red) inside a larger Mobile Communication system model

Capturing the full description of these standards is beyond the scope of this effort, but we can illustrate some possible approachs.

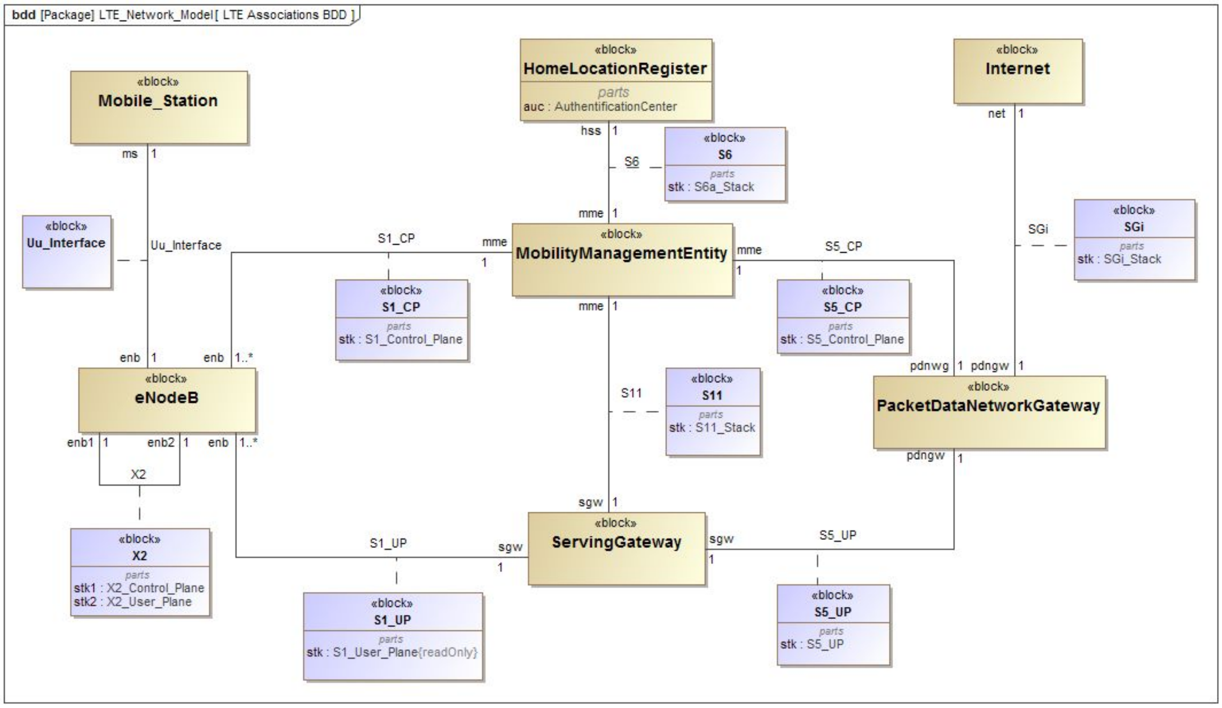


Figure 14 Regional Energy Model block definition diagram, Phase 2

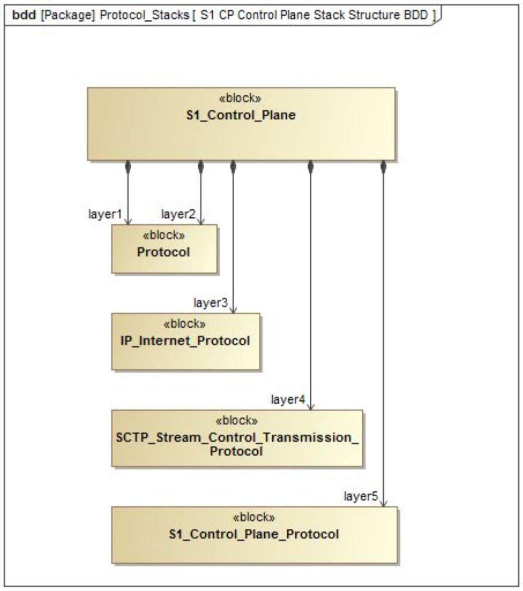

bdd [Package] Protocol_Stacks [S1 CP Control Plane Stack Structure BDD]
«block»
S1_Control_Plane
layer1
layer2
«block»
Protocol
layer3
«block»
IP_Internet_Protocol
layer4
«block»
SCTP_Stream_Control_Transmission_Protocol
layer5
«block»
S1_Control_Plane_Protocol

Figure 15 Example of protocol stack in SysML

First, we identify the major component types in a network. In Figure 14 for an LTE cellular network, these include the eNodeB (cell tower and associated electronics), Mobility Management Entity, Serving Gateway, Packet Data Network Gateway, and Home Location Register, as well as the Mobile Station and Internet. Second, we define a set of interface types using SysML Association Blocks. Third, we decompose each association block as a protocol stack. Figure 15 shows such a stack for the S1 CP interface between eNodeB and Mobility Management Entity. While the task of specifying the many interfaces may be daunting, once completed, they can be placed in a SysML model library (see the Telecom Library in the downloadable models), and used to build specific network configurations in SysML internal block diagrams, e.g. Figure 16.

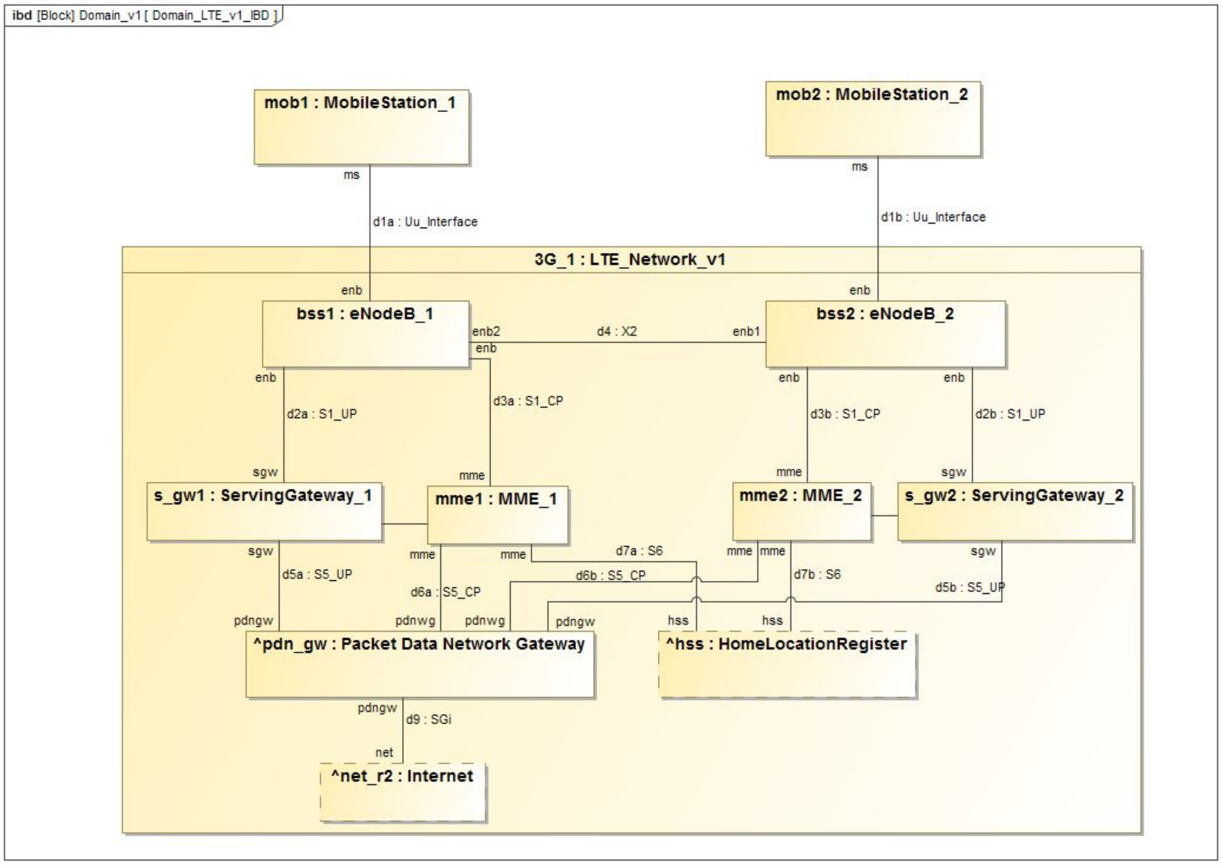


Figure 16 SysML IBD showing interconnection between two mobile stations via an LTE cellular network

While this approach is effective at describing the “static” interfaces, the dynamic communications can best be described with SysML behavior diagrams. Figure 17 uses a SysML sequence diagram to describe the process for establishing a signaling connection between mobile station and GSM network. Note that this is the same formalism used to describe interactions between the phablet and user in Figure 3. By incorporating the phablet within the larger domain model, we can now build sequence diagrams exploring the phablet’s response to scenarios with both user and network messages.

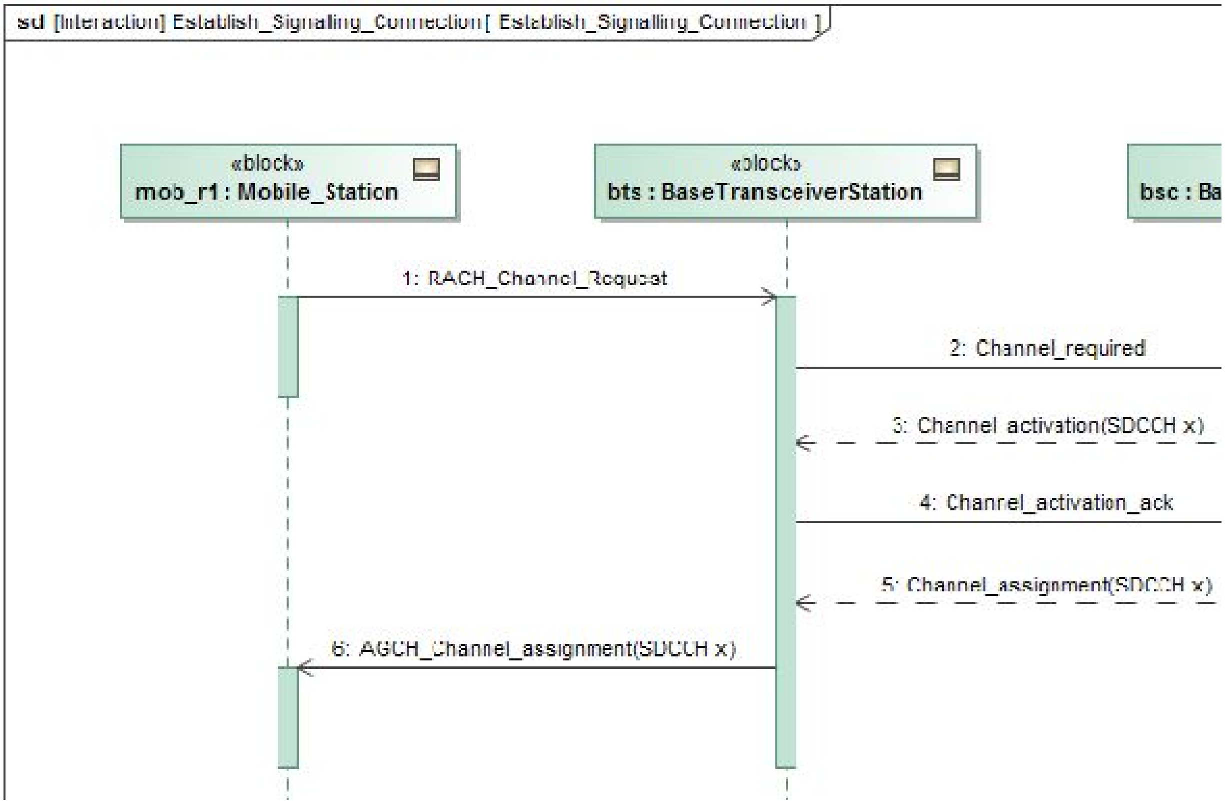

sd [Interaction] Establish_Signalling_Connection [Establish_Signalling_Connection]
«block»
mob_r1 : Mobile_Station
«block»
bts : BaseTransceiverStation
bsc : Ba
1: RACH_Channel_Request
2: Channel_required
3: Channel_activation(SDCCH x)
4: Channel_activation_ack
5: Channel_assignment(SDCCH x)
6: AGCH_Channel_assignment(SDCCH x)

Figure 17 SysML sequence diagram illustrating communication dialog between mobile station and network

The Future of MBE for Electronics Applications

Our discussion began with the observation that most future electronics products will take advantage of greater interconnectivity to increase their functionality. This observation has two important consequences for electronic product design.

- More participants representing more disciplines will be engaged in the development presenting a need for new communication and collaboration approaches. High-level abstractions such as SysML models will facilitate communication, including team members like human factors and user interface designers that may not come from traditional engineering backgrounds.
- With the range of possibilities for user and network interaction expanding geometrically, there is a need for faster, more systematic generation and evaluation of operational scenarios, which the system-of-systems (SoS) models can provide. Using the same high-level modeling languages for both product and SoS simplifies this process.

As electronic products have evolved, so has the design process. In all fields, engineering software is transitioning from independent, stand-alone tools to web-based enterprise applications. Electronic companies prefer a multi-vendor tool environment, using a combination of legacy and best-in-class tools. At the same time, they want product data available to team members anywhere in the development process, on a variety of platforms; laptop, tablet, even smartphone. MBE approaches that supply the ability to search, access and edit distributed data structures support those objectives.

InterCAX is committed to helping organizations in all fields that want to adopt best practices in MBE. For further information, visit us at www.intercax.com or contact us at info@intercax.com.

About the Author

Dr. Dirk Zwemer (dirk.zwemer@intercax.com) is President of Intercax LLC (Atlanta, GA), a supplier of MBE engineering software platforms like Syndeia and ParaMagic. He is an active teacher and consultant in the field and holds Level 4 Model Builder-Advanced certification as an OMG System Modeling Professional.

During twenty years in the commercial electronics field, he worked on the design, manufacturing and testing of printed circuit boards, advanced chip packaging and flat-panel displays. For further information, visit us at www.intercax.com or contact us at info@intercax.com.