QUALITIES OF COMMUNICATIONS PROTOCOLS AND THEIR BEARING ON ENGINEERING SUCCESSFUL SOFTWARE SYSTEMS

DR. STEPHEN CLYDE

UTAH STATE UNIVERSITY

SOFTNET, ICSEA 2016

ROME, ITALY, AUGUST 22-25, 2016
2016 TECHNOLOGY TRENDS

• 8 out of the top 10 technology trends\(^1\) directly involve or require network communications:
  • Device Mesh
  • Ambient User Experience
  • Internet of Things
  • Information of Everything
  • Autonomous Agents and Things
  • Adaptive Security Architecture
  • Adaptive System Architecture
  • Mesh App and Service Architecture

• It almost goes without saying that communications are being integrated into all kinds of applications, e.g.
  • Smart home appliances
  • Smart medical devices
  • Personal and business applications of all kinds

---

A FEW RANDOM THOUGHTS ABOUT COMMUNICATION

Very basic communication: The Chicken and Egg

• Chicken squawks to broadcast that it’s laying an egg
• It communicate something about its ”state”
• However, the chicken does care if others hear her or about the state of the listeners
A FEW RANDOM THOUGHTS ABOUT COMMUNICATION

A slightly more sophisticated example of communication:

The Honey-bee Wiggle Dance

- Upon finding pollen, a scout bee returns to the hive and performs a figure-8 dance, where the orientation, length, and intensity of the wiggles communicate the location of pollen
- The dance communicates state, along with instructions for locating pollen
- The scout bee doesn’t wait for a response from other bees, but competes to be heard
A FEW RANDOM THOUGHTS ABOUT COMMUNICATION

An even more sophisticated example of communication:
**Zazoo’s Head Pointing**

- Zazoo finds me, makes a sort of sneezing sound, points his head in the direction he wants me to go and waits for me to follow.
- He will repeat the sneezing sound and head pointing, if I don’t respond
- He communications involve an exchange of state or information, with some synchronization and reliability
- It may be primitive, but there’s no doubt that we have a “conversation”
CONVERSATIONS AND PROTOCOLS

• A **conversation** is an series of exchanges (messages) among entities (processes) for the purpose of accomplishing some task
• The messages, their sequence, format, semantics, validation, and the rules that govern the each processes expected behavior comprise a **communication protocol**
  • i.e., a protocol prescribes what conversations are allowed
• A set of related communications protocols is a **protocol suite**
LAYERS OF COMMUNICATION PROTOCOLS

System 1
- Application
- Transport
- Network
- Link
- Physical

Application-level Communication Protocols
- TCP / UDP
- IP

System 2
- Application
- Transport
- Network
- Link
- Physical
APPLICATION-LEVEL COMMUNICATION PROTOCOLS (ACP’S)

• Define how the components of application software system communicate
• Can be built on top of or composed other of ACP’s
• Are often unique to and custom built for an application
• Although there are many standardization efforts, certain market and technological forces exist that work against standardization
  • Competition
  • Ongoing need for incremental improvements in a product
  • Advances in technology
  • New user requirements
  • Changes in user expectations
  • Lack of awareness of appropriate standards by developers
THE PROBLEM

• New ACP’s are being written at an accelerating rate
• The quality of ACP’s seems to vary widely, e.g.,
  • Consider common file transfer tools (FTP, SFTP, SCP, file transfer portion of in RDP, etc.) They can different considerably in transfer speeds. Why?
  • Consider the Simple Mail Transfer Protocol (SMTP). Its core documentation is over 30 pages long. Simple?
  • Protocols for exchanging health information (e.g., HL7) are often hundreds of pages. Why and how does this software engineering?
• The need for good software engineers is still growing
• And, the need for training them in implementing effective network communications parallels that growth
MOVING TOWARDS A SOLUTION

• To improve the engineering of applications that have communication requirements, we need
  • More productivity in the development process
  • Ways of estimated the characteristics of the communications a system will have
  • Ways of measuring the actual characteristics of a system once it is operational
  • In other word, higher quality in communication
• But, what is “higher quality” in the context of communications?
• How can we discuss, plan for, implement, and measure such quality?
• How can we teach new developers about quality in communications?
QUALITY MODELS

• Over the last 40 years, there have been many quality models proposed for software (code, software products) and operational procedures (systems in use)
  • Here are a few -- some are very general; others are specialized for certain qualities or types of software

• Over 30 quality models examined to date
• None have addressed issues unique to ACP’s
QUALITIES ➔ FACTORS ➔ ATTRIBUTES ➔ METRICS

Meta-model for Sant’Anna’s Reuse and Maintainability Quality Model (2003)
- e.g., Reuse
- e.g., Flexibility
- e.g., Coupling
- e.g., Depth of Inheritance Tree

- Two qualities models:
  - Software Product
  - System in Use
- Externally observable
- Internal to software or operations
QUALITY MODELS FOR ACP’S

• We adapt the meta-model of ISO 25010
  • Characteristics ➔ general communication qualities
  • Sub-characteristics ➔ externally observable qualities of an ACP
  • Attribute ➔ concrete properties or conditions of the ACP that impact the sub-characteristics and that can be evaluated via metrics
  • Metrics ➔ ways to measure the attributes

• Two quality models for ACP’s
  • Idealistic Quality Model for ACP’s
  • Requirements-oriented Quality Model for ACP’s
IDEALISTIC QUALITY MODEL FOR ACP’S

• Includes characteristics that all ACP’s should try to achieve to some degree, limited only by development resources (developers) and time

• Note that the characteristics may overlap or conflict
  • Achieving one may make it easier or harder to achieve another

• Characteristics:
  • Functional Suitability
  • Operability
  • Security
  • Performance efficiency
  • Maintainability
  • Extensibility
  • Simplicity
FUNCTIONAL SUITABILITY

The degree to which the ACP meets stated and implied needs

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appropriately Accommodating</td>
<td>The degree to which the ACP allows expected conversations</td>
<td>• Involves appropriate participants (e.g., processes)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Authenticates participants</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Includes exchanges of necessary state information at the right time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Uses appropriate messages that trigger desired behaviors in receiving processes</td>
</tr>
<tr>
<td>Appropriately Restrictive</td>
<td>The degree to which the ACP prohibits invalid or undesirable conversations</td>
<td>• Excludes inappropriate or unauthorized participants</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Defines behaviors for illegal messages or message sequences</td>
</tr>
<tr>
<td>Compliance</td>
<td>The degree to which it uses existing standards</td>
<td>• Builds on or integrates existing standards (message format, encoding/decoding, establishing connections, authentication, etc.) were possible</td>
</tr>
</tbody>
</table>
# OPERABILITY

The degree to which the ACP can be understood, learned, operate

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| Recognizable       | The degree to which a developer can recognize whether the ACP is appropriate for their needs | • Has a clear, complete, and update documentation  
• Uses standard or common terms  
• Uses design patterns |
| Learnable          | The degree to which developers can learn how to implement the ACP in an application | • Has a clear, complete, and update documentation  
• Uses standard message formats, encoding, etc.  
• Uses standard or common terms |
| Easy to Operate    | The ease of which network administrators can ensure that the conversations following this ACP can succeed | • Number and type of networks involved  
• Whether participants has dynamic addresses  
• Requires ports  
• Requires tunneling or VPN connections  
• Requires certificates and how they are distributed |
SECURITY

The degree to which the ACP protects against accidental or deliberate misuse, unauthorized access, or destruction of the communicating processes, resources they manage, or network

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authenticity</td>
<td>The degree to which the ACP guarantees that participants are who/what they claimed to be</td>
<td>• Type of authentication (e.g., certificates)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Verification (e.g., certificate authority)</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>The degree to which the ACP guarantees that private data are not accessible to unauthorized parties</td>
<td>• Type of encryption of data</td>
</tr>
<tr>
<td>Integrity</td>
<td>The degree to which the ACP guarantees that messages are not corrupted or tampered with</td>
<td>• Type of encryption</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Type of error detection and correction</td>
</tr>
<tr>
<td>Non-repudiation</td>
<td>The degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later</td>
<td>• Uses digital signatures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Type of synchronization</td>
</tr>
<tr>
<td>Accountability</td>
<td>The degree to which conversations can be traced to communicating processes</td>
<td>• Type of addressing or identification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Type of authentication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Message provenance</td>
</tr>
</tbody>
</table>
PERFORMANCE EFFICIENCY

The degree to which ACP prescribes conversation that have appropriate performance, in terms of response times, turnaround times, and total throughput

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response times</td>
<td>The degree to which the ACP supports conversations where communicating process quickly become aware of other state and that the conversation is proceeding</td>
<td>• Message-to-Message time</td>
</tr>
<tr>
<td>Turnaround times</td>
<td>The degree to which the ACP allows individual conversations to complete in an appropriate amount of time</td>
<td>• Conversation start-end time</td>
</tr>
<tr>
<td>Total throughput</td>
<td>The degree to which the ACP supports an appropriate number of conversations within a fixed about of time</td>
<td>• Completed Conversations per until of time</td>
</tr>
</tbody>
</table>
### MAINTAINABILITY

The degree to which ACP can be modified

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| Modularity         | The degree to which a ACP is broken up into loosing coupling phases or sub-protocol | • Message sequence complexity  
• Presence and number of sub-protocols |
| Cohesion           | The degree to which individual messages or sub-protocols are focused on a single purpose | • Unity of purpose for individual messages  
• Unity of purpose for sub-protocols |
| Reuse              | The degree to which an ACP can be re-used for purposes other than the original one | • Number of uses in other applications |
| Testability        | The degree to which an ACP can be tested                                     | • Support for mockable process  
• Support for mockable messages  
• Deterministic |
EXTENSIBILITY

The degree to which ACP can be extended

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| Open message structure        | The degree to which the messages (their content and structure) can be extended | • Type of messages  
• Message encoding  
• Use of standards |
| Open message sequence         | The amount of flexibility in the acceptable message sequences               | • Stateless operations  
• Unconstrained message sequences |
SIMPLICITY

The degree to which the ACP contains no accidental complexity

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal message sequences</td>
<td>Whether any message in the ACP can be eliminated or combined with another without compromising the functionality, reliability, synchronization</td>
<td>• Structure of the message sequences</td>
</tr>
<tr>
<td>Minimal message content</td>
<td>Whether any content of any message in the ACP be eliminated or combined with another message without compromising, reliability, or synchronization</td>
<td>• Message payloads</td>
</tr>
</tbody>
</table>
REQUIREMENTS-ORIENTED QUALITY MODEL FOR ACP’S

• The degree to which the characteristics in this model SHOULD be obtained depends on the application requirements, not on resources or time.
  • With this model, a “good” ACP is when “The ACP supports the appropriate degree of _______ for this application”, where the blank is the characteristics

• Characteristics:
  • Reliability
  • Synchronicity
  • Longevity
  • Adaptability for scalable distribution
  • Resource utilization
RELIABILITY

The degree to which a communicating process can obtain an assurance that the intended receivers of some message correctly received and reacted to that message.

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| Detectable         | The degree to which a participant in a conversation can detect successfully completed or failed conversations | • Type of message identity  
• Requires messages to be sent after operations, particularly state-changing operations  
• Requires acknowledgements shortly after message receipts  
• Support heart-beat or probe messages |
| Fault Tolerant     | The degree to which ACP can allow conversation to recover from network, server, or process failures | • Allows for detection of lost messages and continuation of the conversation  
• Allows for detection of corrupted or altered messages and the continuation of the conversation  
• Allows for detection and resolution of duplicate messages  
• Allows for detection and resolution of out-of-order messages |
**SYNCHRONICITY**

The amount of inter-process coordination needed in a distributed system relative to execution of actions in a distributed system

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| Informed Operations      | The degree to which a communicating process that need perform an action receives necessary or useful information from other processes | • Number of conversation-related operations that require information from other processes  
                          |                                                                             | • Number of those operations are proceeded by messages with the necessary or useful information                                           |
| Minimize Waiting         | The degree to which a communicating process can proceed with operations related to the conversation without waiting for information from other processes | • Number of blocked-waiting-for-reply states  
                          |                                                                             | • Expected turnaround times the message sequences that cause a process to enter and leave a blocked-waiting-for-reply states |
LONGEVITY

The degree to which an ACP can support long-running conversations caused by long-running operations

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification</td>
<td>The degree to which a process can let other process know that it is beginning, working on, or ending a long-running process</td>
<td>• Whether the conversation triggers long-running operations and others need to be aware of their progress or completion</td>
</tr>
</tbody>
</table>
| Awareness          | The degree to which a process can track the progress of a long-running operation in another process | • Whether the protocol allows long-running operations to be divided into sequential steps or parallel tasks  
                         • Uses state messages to inform others of progress                                      |
ADAPTABILITY FOR SCALABLE DISTRIBUTION

The degree to which an ACP can support scalability distribution (increase in resource and/or number of processes)

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Scalability</td>
<td>The degree to which the ACP allows for an increase in the number of resources</td>
<td>• Type of resource name resolution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supports location transparency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supports migration transparency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supports replication transparency</td>
</tr>
<tr>
<td>Process Scalability</td>
<td>The degree to which the ACP allows for an increase in the number of processes (resource users)</td>
<td>• Type of process name resolution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supports location transparency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supports migration transparency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supports replication transparency</td>
</tr>
<tr>
<td>Concurrent Conversation</td>
<td>The degree to which the ACP allows for an increase in the number of concurrent conversations</td>
<td>• Supports location transparency</td>
</tr>
<tr>
<td>Scalability</td>
<td></td>
<td>• Supports migration transparency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supports replication transparency</td>
</tr>
</tbody>
</table>
The degree to which an ACP can utilize available computation, data, and network resources to maximize overall system throughput and/or reduce response times.

<table>
<thead>
<tr>
<th>Sub-characteristic</th>
<th>Description</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| Load Balancing           | The degree to which the ACP can support the balancing of work loads across devices and adhere to prescribed constraints on the utilization of individual computable resources | • Exchanges device load information  
• Allows for estimation of device loads  
• Supports location transparency  
• Supports migration transparency  
• Supports replication transparency |
| Bandwidth Balancing      | The degree to which the ACP can support the reshaping of network traffic to avoid congestion and adhere to prescribed constraints on the individual or link utilization | • Supports “quality of service” controls  
• Supports location transparency  
• Supports migration transparency  
• Supports replication transparency |
WORK IN PROGRESS

• Continue to refine the characteristics, sub-characteristics, and attributes of each model
• Adopt, adapt, or develop metrics for the attributes
• Use the models to evaluate existing ACP’s
• Study the relationships between characteristics
• Iterate the above
IMPACT ON SOFTWARE ENGINEERING

• Without any notion of quality for ACP’s, developers will be guessing or costly trail-n-error approach to creating any application that requires network communication

• With a quality model for ACP’s, developers would
  • Have a basis for formalizing static metrics, which use ACP designs as input
  • Use those metrics could predicate an ACP’s attributes, and in turn its sub-characteristics and characteristics
  • Be able to answer questions, like
    • Are the requirement-oriented communication characteristics a good fit
    • Are the idealistic communication characteristics achievable with the available resources and time
IMPACT ON SOFTWARE ENGINEERING

(continued)

• Be able to build design tools that help developers decide the degree to which characteristics should and aid in the design of ACP’s
• Capture communication-design expertise in design patterns
• Furthermore, educators could improve curricula to better prepare student for building state-of-the-art software applications
• And, technology professional would have a basis for evaluating, comparing, and recommending software applications
FUTURE WORK

• Formally document
  • Idealistic Quality Model for ACP’s
  • Requirements-oriented Quality Model ACP’s

• Develop tools for
  • Guiding developers in choosing appropriate target degree of support for each characteristic
  • Capturing ACP designs in a structured form
  • Computing measurement based on the metrics
  • Estimating the attributes, sub-characteristics, characteristics of a protocol or entire protocol suite

• Design and conduct empirical studies to validate the models and metrics