Ecosystem-based assurance for plug-and-play medical systems

Oleg Sokolsky

Joint work with Insup Lee, Andrew L. King, Lu Feng, Sanjian Chen, Sam Procter*, John Hatcliff*

* Affiliation: Department of Computer & Information, Kansas State University.
Outline

1. Introduction & Motivation
   • A hypothetical case study
2. Liability in plug & play systems
3. Strawman proposal: plug & play ecosystem
4. Ecosystem-based assurance for plug & play systems
5. Conclusion
Patient Controlled Analgesia

- Clinical standard for pain management
- **Overdose Hazard:** Pump misconfigured, PCA by proxy...

- PCA infusion pump
- Bolus trigger
- Pulse oximeter
Closed-loop Control for Patient Controlled Analgesia

• **Vision for pain management:**
  • External controller monitors patient respiration, then shuts-down pump before overdose happens.
Users buy different components from different vendors and combine them into a new system. **Integration is automated via standardized interoperability protocols.**

Users run software (Apps) to drive overall system behavior and achieve function (usually entertainment or productivity).

Computational Platform (e.g., Personal Computer, iPhone/Android phone...)

Devices register their capabilities/"API" with platform on connection.
Vision: Plug & Play for Medical Systems.

Hospitals/clinics buy different components from different vendors and combine them into a new system. **Integration is automated via standardized interoperability protocols.**

Clinicians run software (Apps) to drive overall system behavior and achieve a **life or safety critical** function.

Devices register their capabilities/"API" with platform on connection.

High Assurance Computational Platform
Closed-loop Control for Patient Controlled Analgesia

- There are new safety hazards to overcome!

Is the interlock adequate?

Does the infusion pump understand controller commands?

Does pulse oximeter send data in the right format?

Safety interlock

Is it connected to the right patient?
What if the Patient is Overdosed?

- A device can be faulty
  - Each device is separately evaluated for safety by regulators
  - Also need to make sure that the device is suitable for interoperability
- Apps and the deployment platform can be faulty
  - Apps should be certified like devices, platform needs high assurance
- Mismatched assumptions can be a cause
  - High assurance standards?

Goal:
- Structure of the safety argument should help us determine, who is to blame
Safety assurance for Plug & Play Medical Systems

Desiderata:
• Need to avoid certification from scratch each time a new app is introduced or a new device comes to market
• The chosen approach should enable compositional reasoning about safety
• The approach needs to be flexible enough to accommodate most apps and devices, yet rigid enough to facilitate development and assessment

Assurance to be supported by an ecosystem
• Clear delineation of responsibilities between stakeholders
• Means of certifying that responsibilities have been fulfilled
• An argument pattern to leverage guarantees from each stakeholder
# A Medical Plug & Play Ecosystem

<table>
<thead>
<tr>
<th>Ecosphere Standards Consortium</th>
<th>Device Vendor</th>
<th>App Vendor</th>
<th>Platform Vendor</th>
<th>Third-party Certifier &amp; Regulatory Authority</th>
<th>Health Delivery Org. (HDO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compliance Passed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compliance Review</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Role: Ecosphere Standards Consortium

Defines:

• **Interoperability standards:**
  • Physical connection: e.g., Ethernet, 802.11 (WiFi), USB, Bluetooth...
  • Connectivity protocols: e.g., TCP/IP, DDS, 11073....
  • Security protocols: Authentication/Authorization/Privacy...

• **Standard device APIs for each device type. (e.g., pump/pulseox...)**
  • How apps interact with devices.
  • Allowed behavior for each API – **Ideally formalized as a model**

• **Standard App runtime environment:**
  • App execution semantics.
  • Platform compliance criteria.

• **App/Platform/Device certification processes and compliance criteria**
# A Medical Plug & Play Ecosystem

<table>
<thead>
<tr>
<th>Ecosphere Standards Consortium</th>
<th>Device Vendor</th>
<th>App Vendor</th>
<th>Platform Vendor</th>
<th>Third-party Certifier &amp; Regulatory Authority</th>
<th>Health Delivery Org. (HDO)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Activity" /></td>
<td><img src="image" alt="Compliance Passed" /></td>
<td><img src="image" alt="Compliance Review" /></td>
<td><img src="image" alt="Compliance Review" /></td>
<td><img src="image" alt="Compliance Review" /></td>
<td><img src="image" alt="Compliance Review" /></td>
</tr>
</tbody>
</table>
Role: Device Vendors

Responsible for designing, manufacturing, and marketing devices.

• Each device:
  • Implements the standard protocols.
  • Implements the standard device APIs for each supported interoperable function.
  • Carries a digital declaration of the APIs it implements.

The Dev. Mfg. **must** provide assurance to the certification authority that their device properly implements the interfaces it specifies. The device will **not** be certified if the vendor’s assurance argument is inadequate.
A Medical Plug & Play Ecosystem

- Ecosphere Standards Consortium
- Device Vendor
- App Vendor
- Platform Vendor
- Third-party Certifier & Regulatory Authority
- Health Delivery Org. (HDO)

Activity
Compliance Passed
Compliance Review
DI Certification
Role: App Vendors

Responsible for designing, implementing, and marketing Apps.

- **Each app:**
  - Is designed for a specific clinical situation.
  - Implements a control/coordination algorithm.
  - Specifies the device interfaces it requires to work properly.

The App vendor acts as a *virtual systems integrator*. They assess the safety of the system assuming their app is used with devices that satisfy the specified interfaces.
A Medical Plug & Play Ecosystem

<table>
<thead>
<tr>
<th>Ecosphere Standards Consortium</th>
<th>Device Vendor</th>
<th>App Vendor</th>
<th>Platform Vendor</th>
<th>Third-party Certifier &amp; Regulatory Authority</th>
<th>Health Delivery Org. (HDO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>Compliance Passed</td>
<td>Compliance Review</td>
<td></td>
<td>DI Certification</td>
<td>App Certification</td>
</tr>
</tbody>
</table>

16
Role: Platform Vendors

Responsible for designing, implementing, and marketing computational platforms.

- The platforms host apps and must:
  - Ensure proper execution of app.
  - Enforce correct composition of app/devices
    - Only devices that satisfy an app’s required APIs.
  - Be rigorously engineered to support life critical functions.
    - Analogy: Integrated Modular Avionics (IMA) infrastructure or High Assurance operating system.
    - E.g., DO-178(b/c) Level A, EAL 6+, etc.

The platform vendor **must** assure their platform will faithfully execute app code and allow only valid system compositions.
A Medical Plug & Play Ecosystem

Ecosphere Standards Consortium

Device Vendor

App Vendor

Platform Vendor

Third-party Certifier & Regulatory Authority

Health Delivery Org. (HDO)

Activity

Compliance Passed

Compliance Review

DI Certification

App Certification

Platform Certification

Bedside System Assembly
A Medical Plug & Play Ecosystem

- **Ecosphere Standards Consortium**
  - Activity
  - Compliance Passed
  - Compliance Review

- **Device Vendor**
  - DI Certification

- **App Vendor**
  - App Certification
  - Platform Certification

- **Platform Vendor**
  - Platform Checks compatibility

- **Third-party Certifier & Regulatory Authority**
  - DI Certification
  - App Certification
  - Platform Certification

- **Health Delivery Org. (HDO)**
  - Bedside System Assembly

**Flow Diagram:**
- From Ecosphere Standards Consortium to Device Vendor
- From Device Vendor to App Vendor
- From App Vendor to Platform Vendor
- From Platform Vendor to Third-party Certifier & Regulatory Authority
- From Third-party Certifier & Regulatory Authority to Health Delivery Org. (HDO)

**Notes:**
- Platform Checks availability
- Platform compatibility
A Medical Plug & Play Ecosystem

Ecosphere Standards Consortium

Device Vendor

App Vendor

Platform Vendor

Third-party Certifier & Regulatory Authority

Health Delivery Org. (HDO)

Activity

Compliance Passed

Compliance Review

Filters out “uncertified” combinations

DI Certification

App Certification

Platform Certification

Bedside System Assembly

Compatibility check

Resource check

operation

PRECISE
App Assurance For System Safety

App developers will need to use model-based-reasoning to provide system-level assurance. Why?

• Each App defines a set of possible systems, each of which is an allowed combination of medical devices and platforms.

• App vendors would not be able to analyze all possible systems directly since.
  • The number of device/platform combinations may be huge.
  • New devices may be admitted after the App is certified.

Suitability of chosen models depends on many factors:

• Regulatory framework assurances & guarantees
  • what behaviors do we expect compliant devices have?

• Intended use of app / operating environment. (higher criticality -> more fidelity)

• Safety property under consideration.
Assume-Guarantee Safety Assurance

**Goal:** guarantee that, given their A, all $P(A) (\| \|_{j=1...n} D_j) \| \| E \models \phi$

The execution of App A on the platform P, denoted by $P(A)$, together with the assembly of medical devices $D_1, \ldots, D_n$ in the environment $E$ satisfies the safety property $\phi$.

**Entities in the assume-guarantee reasoning rule:**

<table>
<thead>
<tr>
<th>Model</th>
<th>Software / Specification</th>
<th>Physical Embodiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td>$A^m$</td>
<td>A</td>
</tr>
<tr>
<td>Interface</td>
<td>$A_I^m (j=1...n)$</td>
<td>$A_I (j=1...n)$</td>
</tr>
<tr>
<td>Devices</td>
<td>-</td>
<td>$D_I (j=1...n)$</td>
</tr>
<tr>
<td>Platform</td>
<td>-</td>
<td>$P_s$</td>
</tr>
<tr>
<td>Environment</td>
<td>$E^m$</td>
<td>E</td>
</tr>
</tbody>
</table>
Assume-Guarantee Reasoning Rule

1. $A^m \cong A$
2. $AI_j^m \cong Ai_j$
3. $E^m \cong E$

Justified informally or by model extraction

<table>
<thead>
<tr>
<th>Model</th>
<th>Software / Specification</th>
<th>Physical Embodiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td>$A^m$</td>
<td>$A$</td>
</tr>
<tr>
<td>Interface</td>
<td>$AI_j^m$ ($j=1\ldots n$)</td>
<td>$AI_j$ ($j=1\ldots n$)</td>
</tr>
<tr>
<td>Devices</td>
<td>-</td>
<td>$DI_j$ ($j=1\ldots n$)</td>
</tr>
<tr>
<td>Platform</td>
<td>-</td>
<td>$Ps$</td>
</tr>
<tr>
<td>Environment</td>
<td>$E^m$</td>
<td></td>
</tr>
</tbody>
</table>

App developers need to assure that models are faithful to the implementation/platform/environment.
Assume-Guarantee Reasoning Rule

1. $A^m \simeq A$
2. $A_{i_j}^m \simeq A_{i_j}$
3. $E^m \simeq E$
4. $A^m \parallel \bigwedge_{j=1}^{n} A_{i_j}^m \parallel E^m \models \phi$

Justified informally or by model extraction

<table>
<thead>
<tr>
<th>Model</th>
<th>Software / Specification</th>
<th>Physical Embodiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td>$A^m$</td>
<td>$A$</td>
</tr>
<tr>
<td>Interface</td>
<td>$A_{i_j}^m$ (j=1…n)</td>
<td>$A_{i_j}$ (j=1…n)</td>
</tr>
<tr>
<td>Devices</td>
<td>-</td>
<td>$D_{i_j}$ (j=1…n)</td>
</tr>
<tr>
<td>Platform</td>
<td>-</td>
<td>$P^s$</td>
</tr>
<tr>
<td>Environment</td>
<td>$E^m$</td>
<td></td>
</tr>
</tbody>
</table>

App developers use model based reasoning to check that the models satisfy the property.
Assume-Guarantee Reasoning Rule

1. $A^m \simeq A$
2. $A_{1j}^m \simeq A_{ij}$
3. $E^m \simeq E$
4. $A^m (\|_{j=1\ldots n} A_{1j}^m) \| E^m \models \phi$
5. $P^s \simeq P$

Justified informally or by model extraction

Enforced by ecosystem

<table>
<thead>
<tr>
<th>Model</th>
<th>Software / Specification</th>
<th>Physical Embodiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td>$A^m$</td>
<td>$A$</td>
</tr>
<tr>
<td>Interface</td>
<td>$A_{1j}^m (j=1\ldots n)$</td>
<td>$A_{ij} (j=1\ldots n)$</td>
</tr>
<tr>
<td>Devices</td>
<td>-</td>
<td>$D_{ij} (j=1\ldots n)$</td>
</tr>
<tr>
<td>Platform</td>
<td>-</td>
<td>$P^s$</td>
</tr>
<tr>
<td>Environment</td>
<td>$E^m$</td>
<td></td>
</tr>
</tbody>
</table>

By 1-3, we can substitute the models for their corresponding artifacts. By 5, we can introduce the platform that “does not get in the way.”
Assume-Guarantee Reasoning Rule

\[ 1-5 \quad A (\big|\big|_{j=1\ldots n} A_{I_j}) \big|\big| P \big|\big| E \models \phi \]

\[ 6 \quad D_{I_j} \simeq D_j \quad \text{Enforced by ecosystem} \]

<table>
<thead>
<tr>
<th>Model</th>
<th>Software / Specification</th>
<th>Physical Embodiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td>( A^m )</td>
<td>( A )</td>
</tr>
<tr>
<td>Interface</td>
<td>( A_{I_j}^m ) (j=1\ldots n)</td>
<td>( A_{I_j} ) (j=1\ldots n)</td>
</tr>
<tr>
<td>Devices</td>
<td>-</td>
<td>( D_{I_j} ) (j=1\ldots n)</td>
</tr>
<tr>
<td>Platform</td>
<td>-</td>
<td>( P^s )</td>
</tr>
<tr>
<td>Environment</td>
<td>( E^m )</td>
<td></td>
</tr>
</tbody>
</table>

**Recall:** All certified devices are compliant with their interfaces: By defn. all \( D_{I_j} \simeq D_j \)
Assume-Guarantee Reasoning Rule

1. \( A \bigwedge_{j=1}^{n} A_l_j \bigwedge P \bigwedge E \equiv \phi \)
2. \( D_l_j \simeq D_j \)
3. \( A_l_j \simeq D_l_j \)

<table>
<thead>
<tr>
<th>Model</th>
<th>Software / Specification</th>
<th>Physical Embodiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td>( A^m )</td>
<td>( A )</td>
</tr>
<tr>
<td>Interface</td>
<td>( A_l_j^m ) (j=1…n)</td>
<td>( A_l_j ) (j=1…n)</td>
</tr>
<tr>
<td>Devices</td>
<td>-</td>
<td>( D_l_j ) (j=1…n)</td>
</tr>
<tr>
<td>Platform</td>
<td>-</td>
<td>( P^s )</td>
</tr>
<tr>
<td>Environment</td>
<td>( E^m )</td>
<td></td>
</tr>
</tbody>
</table>

Recall: Platform enforces compatibility between app device requirements and devices.
Assume Guarantee Reasoning Rule

1-5  \( A (||_{j=1\ldots n} A_l_j) || P || E \models \phi \)
6  \( D_l_j \simeq D_j \)
7  \( A_l_j \simeq D_l_j \)
8  \( A || P \simeq P(A) \)

Enforced by ecosystem

Enforced by platform

<table>
<thead>
<tr>
<th>Model</th>
<th>Software / Specification</th>
<th>Physical Embodiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td>( A^m )</td>
<td>( A )</td>
</tr>
<tr>
<td>Interface</td>
<td>( A_l_j^m (j=1\ldots n) )</td>
<td>( A_l_j (j=1\ldots n) )</td>
</tr>
<tr>
<td>Devices</td>
<td>-</td>
<td>( D_l_j (j=1\ldots n) )</td>
</tr>
<tr>
<td>Platform</td>
<td>-</td>
<td>( P^s )</td>
</tr>
<tr>
<td>Environment</td>
<td>( E^m )</td>
<td></td>
</tr>
</tbody>
</table>

Recall: Platform enforces compatibility between app device requirements and devices and enforces execution semantics.
Assume Guarantee Reasoning Rule

1-5 $A (\| \|_{j=1\ldots n} A_{I_j}) \| P \| E \models \phi$

6 $D I_j \simeq D_j$

7 $A I_j \simeq D I_j$

8 $A \| P \simeq P(A)$

Steps of the reasoning rule can help in accident analysis and establishing liability

The execution of App $A$ on the platform $P$, denoted by $P(A)$, together with the assembly of medical devices $D_1, \ldots, D_n$ in the environment $E$ satisfies the safety property $\phi$. 

<table>
<thead>
<tr>
<th>Model</th>
<th>Software / Specification</th>
<th>Physical Embodiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>App</td>
<td>$A^m$</td>
<td>$A$</td>
</tr>
<tr>
<td>Interface</td>
<td>$A I_j^m (j=1\ldots n)$</td>
<td>$A I_j (j=1\ldots n)$</td>
</tr>
<tr>
<td>Devices</td>
<td>$-$</td>
<td>$D I_j (j=1\ldots n)$</td>
</tr>
<tr>
<td>Platform</td>
<td>$-$</td>
<td>$P^s$</td>
</tr>
<tr>
<td>Environment</td>
<td>$E^m$</td>
<td>$-$</td>
</tr>
</tbody>
</table>
Proposed Assurance Argument Pattern

S: Platform Argument: App the platform based argument

G: Models Adequate: Given the ecosystem certification processes, the models are adequate for $\phi$ and $E$

G: $A^m$ is adq: $A^m$ adequately captures the behavior of $E$ relevant to $\phi$

G: $A^{l,m}$ is adq: $A^{l,m}$ adequately captures the behavior of all $D_n$ that satisfy $A_{i_n}$

Ctxt: Ecosystem: Ecosystem certification processes and compliance criteria

Ctxt: Models:
$A^m \leftrightarrow A$
$E^m \leftrightarrow E$
$A^{l,m}_n \leftrightarrow A_{i_n}$ for all $n$

G: All Sat: There is adequate assurance all instantiations of $A$ satisfies $\phi$ in $E$

G: Model Sat: $A^m | | A^{l,m}_1 | | ... | | A^{l,m}_n | | E^m$ satisfies $\phi$

Justify $\sim$

Model based reasoning
Summary

• Plug & play medical systems is what clinicians want.

• Proposal: Ecosystem for plug & play medical systems:
  • Divides assurance responsibility among stakeholders.
    • Device Mfg. assure API compliance.
    • Platform Mfg. assures faithful app execution.
    • App Mfg. assures based on assumed device/platform compliance.
  • Enables App developer to produce system-level safety argument.

• App assurance argument pattern:
  • Model-based reasoning during design.
  • Forces developer to justify modeling assumptions based on ecosystem guarantees.
Thank You!

Questions?