



자율주행차는 안전할까요?

WRITTEN BY

Christopher Giordano

Vice President UX/UI Technology

The DISTI Corporation

DISTI[®]

Jim Carroll

Chief Technology Officer

Digica

DiGiCA
Intelligent Digital Solutions



자율주행차는 안전할까요?



자율주행차 (Autonomous Vehicles: AVs)의 UI 기능 안전

언뜻 보기에 자율 주행 차는 사용자 인터페이스 (UI) 기능 안전에 대한 중요한 요구 사항이 없는 것처럼 보일 수 있습니다. 자율 주행 차에는 운전자 컨트롤이 적어 지므로 UI가 감소 할 것입니다. 남아있는 컨트롤은 인포테인먼트와 관련이 있을 수 있으며 기능 안전을 준수 할 필요가 없습니다.

그러나 기존의 운전자 제어가 덜 중요해질 수 있지만 자율 주행 차량의 특성으로 인해 차량 제어에 대한 새로운 추가 요구 사항이 발생합니다.

AV의 표준화는 하룻밤 사이에 발생하지 않을 것이며, 이는 기존의 운전자 제어 차량 (수동차량) 이 AV로 전환하는 기간 동안 이 두 종류의 차량은 도로를 함께 사용해야 합니다. 이 전환 기간 동안 AV에는 안전 관련 고려 사항이 있는 기존의 운전자 제어 수준이 어느 정도 필요합니다. 이 문서에서는 이러한 UI가 무엇 인지와 기능적 안전에 대한 요구를 수용하기 위한 소프트웨어 아키텍처 및 생성 프로세스를 개발할 때의 과제와 잠재적 솔루션에 대해 설명합니다.

AVs에서는 어떤 UI가 필요할까요?

어떤 UI가 디자인의 어떤 부분에 들어가는지에 대해서는 최근 논쟁 중이며 이는 UI와 인터페이스 하는 하위 구성 요소에 따라 달라집니다. 기능 안전 ASIL (Automotive Safety Integrity Level) 등급 UI의 경우, 통상적으로 인스트루먼트 클러스터의 핵심 구성 요소에 UI의 강조점을 두고 있습니다.

클러스터는 본질적으로 차량의 움직임, 동작, 오류 및 동작을 제어하는 통합 차량 부품에 연결됩니다. 이러한 시스템은 '치명적인' 장애 환경을 조성하여 인명 손실을 초래할 수 있습니다.

최근까지 다양한 도메인에는 특히 기능 안전과 관련하여 매우 다른 UI 요구 사항이 있었습니다. AV의 빠른 정착과 함께 ASIL 등급의 UI는 AV의 다양한 구성 요소에서 요구 사항과 사용 사례를 만드는 사람들에게 완전히 새로운 의미를 부여합니다.





인포테인먼트

예를 들어, 인포테인먼트 시스템은 통상적으로 고장이 발생하더라도 이동중인 차량에 안전 위협을 주지 않는 통합되지 않은 시스템의 일부입니다. 서브 시스템 (여러 디스플레이를 관리하는 하나의 SoC)의 통합으로 인해 인포테인먼트는 운전석에서 더 중요한 역할을 하기 시작했습니다.

기능 안전 요구 사항은 클러스터 및 HUD (헤드 업 디스플레이)에서 여러 디스플레이를 실행할 수 있는 성능과 기능을 갖춘 단일 헤드 유닛 시스템으로 전환되고 있습니다. 하드웨어 관점에서는 효율적이지만 소프트웨어 개발자의 관점에서는 이 문서에서 자세히 논의할 특별한 과제를 제시합니다. 하드웨어 및 디스플레이 기술도 이제 대량 판매용 자동차에서 HUD를 사용할 수 있는 지점에 도달했습니다.

새로운 시대의 UI 및 소프트웨어 아키텍처 의사 결정 도입: ASIL 등급이 요구되는 데이터의 경우 앞 유리에 표시 할 데이터와 등급은 무엇일까요? 클러스터와 HUD가 인포테인먼트 시스템에서 더 많은 역할을 하기 시작함에 따라 기능적으로 안전한 아키텍처와 회사의 적절한 안전 문화가 브랜드의 성공과 안전에 가장 중요한 요소가 되었습니다.

원격 제어 (Remote Control)

원격 차량 제어도 ASIL 등급 UI의 유력한 후보이지만 UI가 명확하거나 일관적이지 않습니다. 원격 기능에 대해 생각할 수 있는 방법에는 여러 가지가 있습니다.

- » 단순한 저속 자동 주차에서 차량 외부의 운전자는 일반적으로 원격 요청 키 또는 연결된 모바일 스마트 폰에서 정보를 호출합니다.
- » 차선 유지 지원, 긴급 자동 제동 또는 어댑티브 크루즈 컨트롤과 같은 교통 및 자동 제어권 전환 이벤트 하에 운전하는 차량의 운전자.

첫 번째 경우에서 정의한 것처럼 원격 기능은 일반적으로 차량에 대한 "fire and forget" 명령입니다. 그러나 UI는 여전히 명확하고 일관적이며 신뢰할 수 있어야 합니다. 차량의 움직임을 제어하여 일정 수준의 ASIL 등급을 부여합니다. 하지만 잠재적으로 UI 자체는 없습니다.

두 번째 경우에서 정의한 것처럼 원격 기능은 차량이 움직일 수 있도록 약간의 움직임 조정에 관한 것입니다. 이 상황의 운전자는 이러한 활동을 중단 할 수 없거나 시간이 없을 수 있습니다. 여기에서 안전하고 신뢰할 수 있는 아키텍처도 중요합니다.



비상제어정지장치 (Emergency Override)

의심 할 여지없이 AV에서 UI가 매우 중요한 부분은 '제어권 전환 이벤트'(Takeover event)'라고도 하는 비상제어정지장치 (Emergency Override)의 개념입니다.

120Kph의 고속도로에서 차량이 움직이는 것을 상상해보십시오. 갑자기 외부 이벤트로 인해 차량이 결정을 내리는 데 필요한 정보가 없는 새로운 상태가 됩니다. 이 시점에서 차량은 제어권전환 (Takeover event) 호출하여 운전자가 잠재적으로 생명에 중요한 결정을 내리기 위해 모든 중요한 시스템의 상태를 알 수 있는 시간을 제공합니다.

이 정보는 일반적으로 차량에서 운전자에게 청각적, 시각적으로 전송됩니다. 다중 감각 통합 연구에 따르면 청각 및 시각 데이터의 조합은 개선된 평균 반응 시간을 생성하므로 보다 시기 적절한 인지 반응이 가능합니다¹. 따라서, 인지 전달을 개선하기 위해서 명확하고 간결하며 신뢰할 수 있는 UI가 매우 중요해졌습니다.

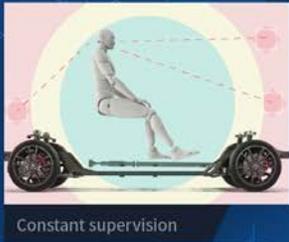
완전한 자율로 직접 이동하고 부분 시스템을 건너 뛰어 제어권전환 이벤트 (Takeover event)의 필요성을 없애자는 데 대한 오랜 논쟁이 있습니다. 이 문서는 다양한 자율성 수준과 그 의미에 대해 다음 섹션 이후에 더 자세히 설명합니다.

1 - Rosemann, S., Wefel, I., Elis, V., & Fahle, M. (2017). Audio-visual interaction in visual motion detection: Synchrony versus Asynchrony. Journal of Optometry, 10(4), 242-251. doi:10.1016/j.optom.2016.12.003

완전자율주행차 도달까지의 긴 여정

AV의 기능이 계속 발전함에 따라 위에 나열된 도메인의 시스템 요구 사항 정의와 마찬가지로 UI 요구사항은 여전히 유효적입니다. 한 가지는 확실합니다. 기능 안전은 시스템의 복잡성이 증가함에 따라 UI 및 소프트웨어 아키텍처에서 지속적으로 증가하는 역할을 합니다. 동시에, 전통적 방식으로 구동되는 차량과 AV와의 격차는 좁혀지고 있습니다. 수동과 부분 자동화로 전환 기간과 궁극적으로 완전 자동화 사이의 전환 기간에는 명확한 정의가 요구됩니다.

자율 주행의 경우 SAE (Society of Automotive Engineers) 2에서 정의한 5 가지 레벨이 있습니다.²



레벨 0 수동운전

인간이 운전



레벨 3 조건부 자동화

차량의 자율주행시스템 (ADS)은 특정 상황에서 운전 작업의 모든 면을 자체적으로 수행 할 수 있습니다. 이러한 상황에서, ADS가 운전자에게 시스템 개입 요청 시 운전자는 언제든지 제어권을 가져올 준비가 되어 있어야 합니다. 다른 모든 이벤트에서는 운전자가 운전 작업을 수행합니다.



레벨 1 운전자 지원

차량의 첨단운전자 보조시스템 (ADAS)은 때때로 운전자의 조향 또는 제동 / 가속을 지원하지만 동시에 둘 다 지원하지는 않습니다.



레벨 4 고등 자동화

차량의 자율주행시스템 (ADS)은 자체적으로 특정 환경에서 모든 운전 작업을 수행하고 운전 환경을 모니터링 (기본적으로 모든 운전을 수행) 할 수 있습니다. 인간은 이러한 상황에서주의를 기울일 필요가 없습니다.



레벨 2 부분 자동화

차량의 첨단운전자 보조시스템 (ADAS)은 특정 상황에서 조향과 제동 / 가속을 동시에 제어 할 수 있습니다. 운전자는 계속해서 완전한 주의를 기울여야 하며 ("운전 환경 모니터링") 나머지 운전 작업을 수행해야 합니다.



레벨 5 완전 자동화

자율 주행 시스템 (ADS)은 모든 상황에서 모든 운전을 할 수 있습니다. 인간 탑승자는 승객 일 뿐이며 운전과 관련 할 필요가 없습니다.

앞서 언급했듯이 많은 사람들은 자율성에 대하여 '쇠뿔도 단김에 빼야 좋다 (rip the band-aid off)'라는 방식이 더 안전한 접근 방식이며 우리는 모든 차량을 레벨 5로 바로 이동시켜야 한다고 믿습니다. 이론적으로는 이것이 이상적 일 수 있지만 비용, 시장 출시 시간, 시스템 테스트 및 이러한 기술에 대한 대중의 편안함 수준과 같은 여러 가지 다른 사항들도 고려해야 합니다.

대중의 AV 채택 의지는 여전히 엇갈린 반응입니다. 우리는 현재 전환기에 있습니다. 우리는 많은 양의 수집된 데이터를 바탕으로 생명에 중요한 운전 결정을 위한 보다 새로운 인공지능시스템을 구현하는 노력을 지속해가고 있습니다. 이 사실은 수월하게 그리고 적어도 지역적으로 레벨 5 자율성에 도달하기 전에 여전히 갈 길이 있음을 강조하는 것입니다. 현재 레벨 5로 견고하게 도약하기에는 너무 많은 요소가 관련되어 있습니다. 주요 질문은 다음과 같습니다.

» 사용 된 기술은 안전합니까?

» 차량을 설계하고 제조하는 OEM 및 Tier 1에 허용수준의 안전 문화가 있습니까?

» 도로상에서 AV는 수동차량과 어떻게 상호 작용합니까?

» 이동 중인 차량 시스템에 대해 어느 정도의 인간 제어가 있어야 하나요? 그렇다면 그것은 무엇입니까?

» 순수 AV의 사고에 대한 책임은 누구에게 있습니까?

» 창의성을 억누르는 수준의 비용 증가 없이, OEM 개발에 요구되는 표준을 정의하는 담당기관은 누구입니까?

AV의 입증된 안전성, 이동성 및 경제적 이점이 있지만, 표준으로서 레벨 5 완성의 단계에 점점 가까워지면서 하나의 요구사항은 명백합니다. 적어도 단기적으로는 AV와 관련된 일정 수준의 인간 제어 또는 개입 기능이 유지 될 것입니다. 이 문제는 우리가 기술을 증명할 때까지 계속 될 것이며 대중은 통제 부족이 훨씬 더 안전한 환경으로 이어질 수 있음을 받아들입니다.

2 - Lynberg, M. (2020, June 15). Automated Vehicles for Safety. Retrieved from <https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety>



AVs에서 UI의 중요한 사항

UI가 멋지게 보이고 브랜드와 일치해야 하는 기본적인 기업 요구 사항 있습니다. 어떤 사람들은 UI의 모양과 느낌이 다른 브랜드와의 주요 차별화라고 주장 할 수 있습니다. 따라서 기본 요구 사항 중 하나는 인터페이스가 '좋아 보인다'는 것인데, 이는 설계자마다 주관적으로 다른 의미를 갖는다는 암시적 이해가 항상 존재합니다.

안전을 위해 UI는 산만함과 상호 작용 시간을 최소화 할 수 있도록 직관적이고 단순해야 합니다. 이러한 UI에 추가된 기능적 안전 수준에 관해서는 OEM 및 공급 업체가 해석할 수 있는 여지가 남아 있습니다. NHTSA, DOT, SAE 및 EC 지침은 있지만 특정 자율성 수준에 따라 준수해야 할 엄격한 FuSa 규칙은 없습니다. 이러한 지침은 변경되며 자율성 수준이 증가할수록 더욱 엄격해져야 합니다.

순수 자율성은 일단 우리가 거기에 도달하게 되면 그 자체의 중요한 레벨을 가질 것입니다. 그러나, 안전 관점에서, 과도기 동안 단기적으로, 자율주행차의 UI에 필수적인 것은 단순성, 성능 성, 안전성의 세 가지입니다.

- » 제어권전환 이벤트 (Takeover events) 동안 인간에 대한 인지 부하 및 중요한 데이터 전송을 최소화하는 단순성.
- » 중요한 콘텐츠가 늦지 않게 표시되도록 하는 런타임 성능.
- » 시스템 안전 설계 전략에서 의도 한대로 중요한 구성 요소를 정확하게 볼 수 있도록 보장하는 기능 안전.

미래로의 운전

UI 관점에서, 위의 모든 사항은 입증된 상용화 된 소프트웨어 도구와 안전 문화를 염두 한 잘 설계된 소프트웨어 아키텍처를 통하여 달성될 수 있습니다. 하드웨어와 소프트웨어는 계속 발전하고 더 효율적이 되어 더 많은 데이터를 더 빨리 처리합니다. 따라서 각각의 시스템을 위한 모든 개발 프로세스에 맞는 단일 사이즈는 없습니다. 오히려 빠르고 효율적인 적응을 통해 변화를 포괄하고 미래의 기술을 입증 할 수 있는 유연성이 모든 OEM 및 Tier 1 공급 업체의 성공을 위한 초석이 됩니다.

예를 들어, 여러 디스플레이를 관리하는 단일 시스템에 대한 최근 추세는 10년 전에는 쉽게 가능하지 않았습니. 하드웨어가 더 작아지고 빨라지고 전력을 덜 사용하지만 여전히 bus를 통해 방대한 양의 데이터를 전송할 수 있게 됨에 따라 각 디스플레이가 자체 SoC를 갖는 대신 한가지 이상을 처리할 수 있는 단일 SoC가 여러 디스플레이를 관리하는 아이디어를 얻을 수 있고 비용적으로도 실행 가능해졌습니다. 하드웨어 아키텍처의 이러한 변화로 인해 기존의 ASIL를 처리하지 않았던 시스템(IVI) 과 처리가 필요한 시스템 (Cluster & HUD)을 혼합하는 소프트웨어 아키텍처의 변화가 필요하게 되었습니다.

이 통합 시스템은 개발자의 관점에서 소프트웨어 아키텍처에 고유 한 문제를 제기했습니다. 이 문제는 기본 아키텍처를 오류 위험에 노출시키지 않으면서 적용할 수 있는 매우 유연하고 상용화 된 도구를 사용하여 완화됩니다.



상용 도구 (Commercial Tooling)

기능 안전을 위해 설계된 도구가 그렇지 않은 도구보다 더 비싸다는 인식이 있습니다. 이러한 도구 채택 접근 방식은 소프트웨어 인증을 위한 자동차 OEM의 노력 비용을 낮게 유지하면서 절감된 초기 엔지니어링 비용으로 효과가 더 커집니다.

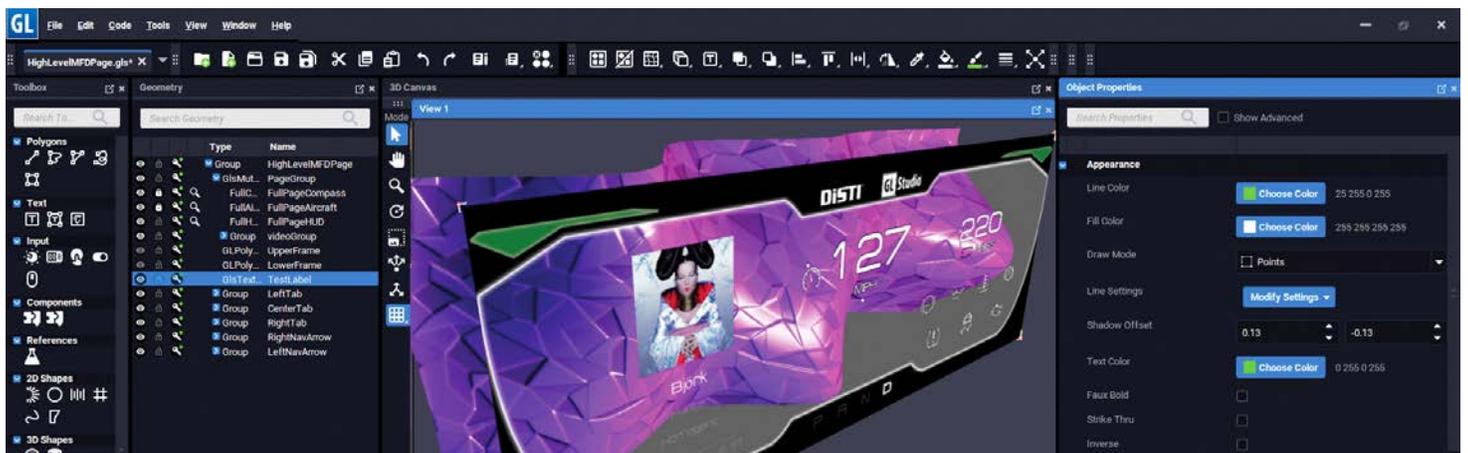
"Aviation Week & Space Technology", 2017년 11월에 발표된 Rockwell Collins 인용

Rockwell Collins의 대답은 자동차 산업 표준의 측면 중 하나인 위험 기반 시스템입니다. 항공전자 및 자동차에 대한 메시지는 분명합니다. 안전에 많은 돈이 쓰입니다. OEM이 어떻게 비용을 절감하면서 동시에 안전 표준을 유지하면서 창의성을 저해하지 않을 수 있을까요? **이미 비용을 부담해 왔으며 기능 안전을 위한 관련 시장에서 입증된 적절한 상용 도구를 사용하십시오.**

"예전에는 새로운 항공 전자 시스템의 경우 1달러당 75센트가 엔지니어링에, 25센트가 글로벌 기관의 인증에 사용되었습니다. 오늘날 그 비율은 크게 변화했습니다."³

자동차 소프트웨어 인증은 지난 10년간 자동차 OEM들의 당면과제였습니다. 인증 노력은 자동차 안전 표준을 개선할 수 있지만, 문제는 혁신과 창의성을 억제하고 개발 비용을 높일 수 있다는 것입니다.

자동차 OEM들이 개발 일정을 단축하려는 시대에 인증 노력은 관료주의를 장려함으로써 반대로 개발 일정을 연장한다는 인식을 주고 있습니다.



반도체 공급 업체는 자동차 부문에서 경쟁 우위를 확보하기 위해 FuSa 호환 도구를 제공하는 것이 중요하다는 것을 인식했습니다. ARM은 안전에 중요한 지원을 제공하는 컴파일러 기능 패키지를 제공합니다. ARM은 많은 SoC에 포함된 프로세서에 실리콘 IP를 제공하므로 반도체 회사는 최소한의 추가 노력으로 이 지원을 채택할 수 있습니다. 파급 효과는 다른 SoC 및 실리콘 IP 공급업체가 이 부문에서 경쟁력을 유지하기 위해 지원을 제공해야 한다는 것입니다.

다른 유사한 영역에서 사용된 도구의 경험을 활용하는 것은 혁신을 촉진하기 위한 리스크가 없는 저비용 접근 방식입니다. 예를 들어, GL Studio UI Toolkit은 20년 이상의 전통을 가지고 있으며 현재 항공기, 우주선 및 헬리콥터에서 사용 중이며 생명에 중요한 의료 기기에 전 세계적으로 사용됩니다. GL Studio는 2015년 4월에 ISO 26262 ASIL D 표준을 인증 받은 최초의 UI 도구이기도 합니다.

그들은 또한 원자력 시설에 대한 NQA-1 사전 인증을 통과했으며, 이는 항공 전자에 대한 FAA의 DO-178C DAL A 인증보다 훨씬 더 엄격한 프로세스입니다.⁴ 이러한 도구의 개발 비용은 30년에 걸쳐서 동일한 고민과 요구 사항을 가진 다양한 산업에서 폭넓은 시장 및 고객에 의해 처리됩니다. 따라서 항공전자등과 같은 시장을 위한 기존 도구 및 경험을 활용하여 보다 엄격한 테스트와 더 나은 성능을 통해 훨씬 더 높은 품질을 생산하는 것은 위험이 낮고 비용이 저렴한 솔루션이 될 수 있습니다.

이 모든 것이 ADAS 시스템과 자동 조종 장치가 등장하는 시대에 훨씬 더 안정적인 소프트웨어 아키텍처를 제공합니다. UI는 이것과 어떤 관련이 있습니까? 이 문서의 앞부분에서 논의된 제어권 전환 (Takeover event) 또는 비상제어정지장치 (emergency override)가 관련이 있습니다.

3 - Statler, K. L. (2017, November 13). The World Needs Seamless Aviation Certification Standards. Aviation Week & Space Technology.
4 - HMI and UI Design Software: Embedded Target Systems. (2019, June 12). Retrieved August 28, 2020, from <https://glstudio.distl.com/features/safetycritical/>

임베디드 소프트웨어

소프트웨어에서 UI 기능 안전에 대한 지원은 SC(Safety-Critical) 애플리케이션을 만들 수 있는 플랫폼과 사용자가 상호 작용하는 애플리케이션의 두 가지 핵심 영역에 있습니다.

플랫폼 소프트웨어는 GPU, 프로토콜 및 코덱을 구현하는 미들웨어, 애플리케이션 계층에 일관된 API를 제공하는 추상화 계층과 같은 기본 하드웨어 구성 요소를 직접 제어하는 장치 드라이버 및 기타 BSP 구성 요소를 비롯한 많은 구성 요소로 구성됩니다. Khronos와 같은 조직은 OpenGL, OpenGL ES 및 Vulkan과 같은 애플리케이션 개발자가 사용할 산업 표준 API를 지정합니다. 이러한 API의 SC (Safety-Critical)로의 변형은 이미 개발 중인 최신 변형과 함께 사용할 수 있습니다. 이러한 API를 지원하기 위해 GPU 공급 업체에서 SC 호환 버전의 드라이버를 제공하고 있습니다.

하드웨어 리소스가 차량 기능 간에 공유되는 경우 소프트웨어 가상화는 일반적으로 기능 간의 분리를 달성하기 위해 구현됩니다. 이러한 환경에서 호스트 및 게스트 드라이버와 하이퍼바이저도 기능 안전을 준수해야 합니다.

동일한 칩에 하드웨어 가상화 및 GPU 분리 기능도 있습니다. 이것은 사용되는 아키텍처에 영향을 미칠 수 있는 관련 문제이지만 하드웨어 가상화는 이 문서의 소프트웨어 아키텍처 범위를 벗어납니다.

왼쪽 그림의 아키텍처에서 알 수 있듯이 소프트웨어 가상화 환경은 복잡 할 수 있습니다. 이러한 접근 방식은 생산 전 결함 해결 프로세스를 복잡하게 하여 전체 개발주기를 연장시킵니다.



SC 규정을 준수하는 도구와 소프트웨어 구성 요소는 빠르게 성숙되고 있습니다. 중기 적으로, 소프트웨어 제품은 완전한 SC 규정을 준수하는 시스템들의 구성을 단순화하여, 하이퍼 바이저가 덜 중요해질 수 있습니다. 이 방법은 출시 시기와 개발비 및 소프트웨어 라이선스 비용을 줄여줍니다. 경우에 따라 SC 도구는 Non-SC에 해당하는 것과 동일한 가격이 발생합니다. 이러한 경우 완전한 SC 준수를 사용하면 추가 비용 없이 전체 SC 준수의 추가 이점과 함께 소프트웨어 아키텍처를 단순화 할 수 있습니다.

차량 UI의 맥락에서, 애플리케이션 소프트웨어는 주로 차량 탑승자가 차량과 상호 작용하는 사용자 인터페이스입니다. 이 맥락에는 클러스터, 환경 제어 및 인포테인먼트와 같은 드라이버 기능이 포함됩니다. 이러한 SC API 및 GL Studio와 같은 기능 안전 UI 도구를 사용하여 애플리케이션을 기능적으로 안전하도록 구축 할 수 있습니다.

코드를 프로그래밍하기 전에 프로젝트의 설계 단계에서 적절한 기능 안전 소프트웨어 구성 요소를 선택하는 것이 중요합니다. 개발주기 동안 기본 기술 선택 항목을 변경하면 상당한 기술적 부채, 추가 통합 작업이 발생하고 결과적으로 소프트웨어의 기능 안전성이 저하 될 수 있습니다.



자동차안전 무결성 수준 (Automotive Safety Integrity Level : ASIL)

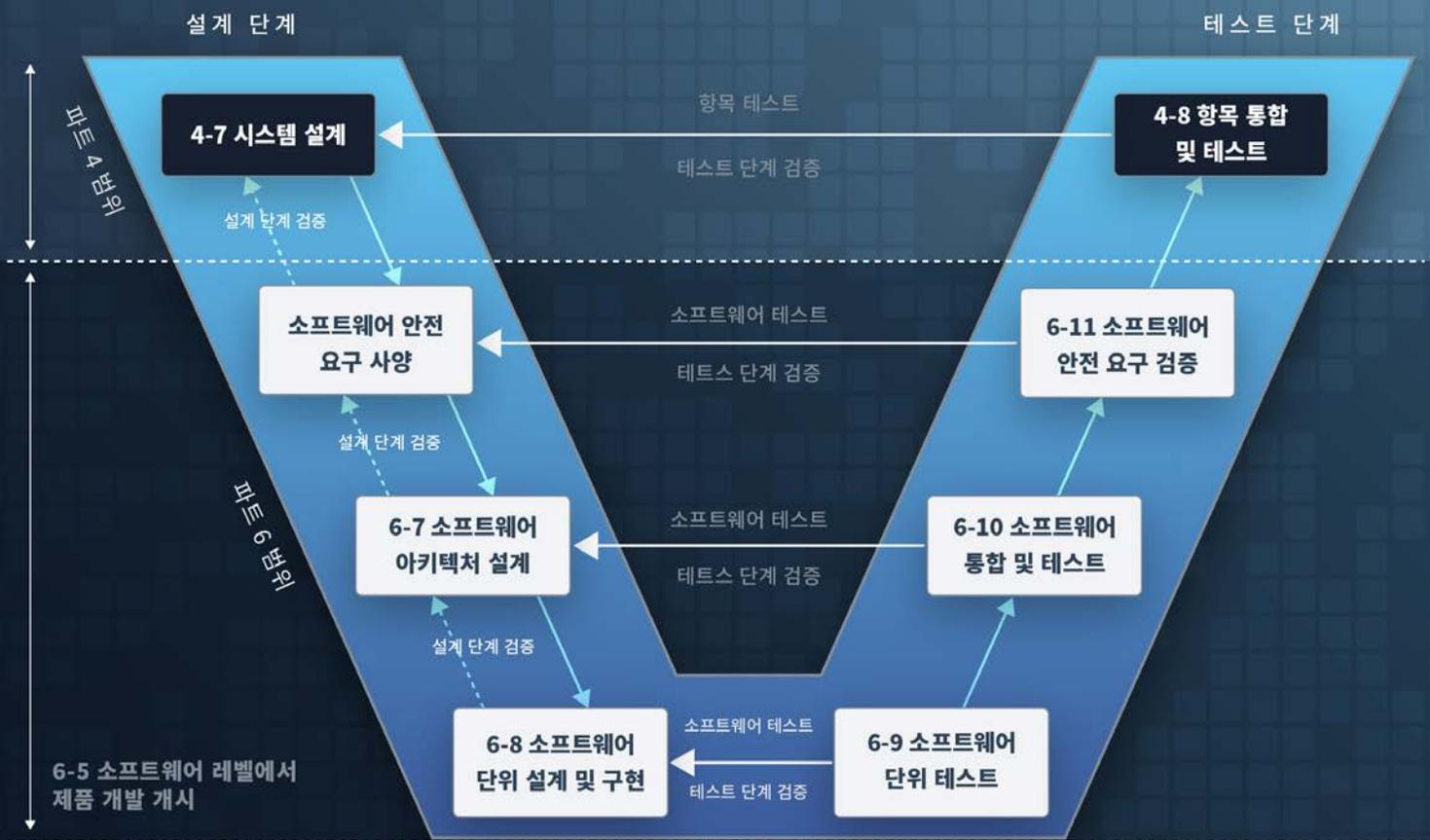
기능 안전의 핵심은 위험 평가입니다. 최종 차량 사용자의 위험을 최소화하기 위해 차량 개발 중에 엄격한 프로세스가 적용됩니다. ASIL (Automotive Safety Integrity Level)로 알려진 위험 분류 체계는 이러한 위험을 평가하는 데 자주 사용됩니다. ASIL A (최저)부터 ASIL D (최고)까지 네 가지 위험 레벨이 있습니다. ASIL D로 평가 된 위험은 생명을 위협하는 위험을 나타내며 이러한 개발에는 가장 엄격한 개발 프로세스가 적용됩니다.

일반적으로 ASIL 작업 시 적용되는 원칙은 산업 표준과 규제 프레임 워크를 준수하면서 추가 비용이 발생하지 않는 최고

레벨에서 작업하는 것입니다. 이러한 접근 방식은 제품 신뢰성을 향상시킵니다; 또한 요구사항 변경 시 유연성을 허용합니다. 변화. 안전이 중요한 환경에서 소프트웨어를 개발하는 데 드는 비용을 감안할 때 수명을 연장하고 장기 비용을 줄이는 것이 중요합니다.

ASIL은 광범위한 표준 인 ISO26262의 하위 집합으로, 자동차 산업의 제품 개발 프로세스에 대한 지침도 제공합니다. 차량 UI 개발에서 이러한 조치를 준수하면 기능 안전이 보장됩니다.

ISO26262의 시스템 및 임베디드 소프트웨어 V-사이클



이 프로세스는 ASIL에서 식별 된 위험이 적절히 설계, 구현 및 테스트 된 솔루션과 함께 문서화되도록 설계되었습니다. "V- 모델"프로세스는 다음과 같은 몇 가지 주요 기능을 강조하는 표준에 의해 설명됩니다.

- » 문서화 된 요구 사항, 설계 및 구현에 따른 추적관리를 포함한 엔지니어링 프로세스.
- » 테스트 주기에서 피어 (peer) 프로세스에 의한 각 개발 단계의 산출물에 대한 구조적 검증.
- » ASIL이 식별한 위험을 다루는 시스템의 안전 요구사항에 특히 초점을 맞춘다.

V 모델 (및 그 변형)은 잘 이해되어 있으며 일반적으로 임베디드 소프트웨어 개발 프로젝트에 사용됩니다. 사용법은 인포테인먼트 및 드라이버 애플리케이션, UI 라이브러리, 그래픽 드라이버를 포함한 모든 자동차 애플리케이션 스택 구성 요소를 만드는 데 동일하게 적용됩니다.

모든 UI 소프트웨어 구성 요소는 SC를 준수하기 위해 이러한 표준에 따라 개발되어야 합니다.



GPU

최신 UI는 놀라울 정도로 복잡하며, 구축된 모든 컨텍스트에서 최신 운영 체제의 구석구석까지 퍼져 있습니다. UI 경험은 사용자가 차량에 탑승하는 순간부터 OEM의 스플래시 화면에서 메뉴 시스템을 거쳐 내비게이션 애플리케이션에 이르기까지 사용자 경험의 최전선에 있습니다. 사용자 경험의 복잡성과 중요성 때문에 성능이 매우 중요합니다. 모든 최신 UI는 GPU가 포함 된 하드웨어에 구축되므로 그래픽 기능이 애플리케이션 프로세서에서 오프로드 될 수 있습니다.

GPU의 복잡성으로 인해 SoC 회사는 종종 타사의 GPU 설계에 라이선스를 부여하여 장치에 포함합니다. Imagination Technologies와 Arm은 "라이선스가 가능한" GPU 설계의 선도적인 공급 업체입니다. 다른 SoC 회사는 애플리케이션 프로세서와 잘 통합 된 GPU를 설계합니다. 이 접근 방식은 Nvidia, Qualcomm 및 Intel과 같은 회사에서 사용합니다.

업계는 소프트웨어 통합을 단순화하기 위해 그래픽 프로그래밍을 위한 표준 API를 정의하기 위해 Khronos와 같은 기관을 구성했습니다. OpenGL은 거의 30년 전에 데스크톱 PC 용으로 처음 지정되었습니다. 기본 하드웨어가 진화하면서 많은 수정을 거쳤고 임베디드 시스템에서 사용하기 위해 OpenGL ES와 같은 여러 관련 API를 생성했습니다.

기타 고려 사항

차량 OEM은 차량의 BOM(Bill Of Materials) 비용을 줄이기 위해 지속적으로 노력하고 있습니다. 이 접근 방식은 일부 SoC를 제거하는 것을 의미 할 수 있으며, 이는 차량 기능간에 물리적 프로세서를 공유 함을 의미합니다. 문서의 앞부분에서 논의했듯이 이 기능과 관련이 있는 GPU 가상화 기능도 있습니다. SC 기능과 Non-SC 기능이 단일 프로세서에 공존하는 경우 해당 프로세서에서 호스팅되는 모든 기능에서 단일 SC UI 소프트웨어 스택을 사용하는 것이 기술적으로 더 간단합니다.

역사적으로 차량 내 시스템은 표준 SoC SKU를 기반으로 했습니다. SoC는 매우 복잡한 장치이므로 비교적 비쌉니다. 특정 애플리케이션의 경우 FPGA 또는 MCU와 같은 저가 실리콘에 시스템을 구축하는 동시에 고품질 사용자 인터페이스 및 SC 준수에 대한 중요한 요구 사항을 충족 할 수 있습니다. 여러 ISV가 소프트웨어 렌더러를 제공합니다. 이는 GPU의 필요성과 운영 체제간에 공유하는 복잡성을 제거합니다. GPU는 SoC에 포함 된 가장 큰 프로세서이므로 이를 제거하면 시스템 공간도 줄어듭니다.

CoreAVI와 같은 회사의 GPU 드라이버 제작을 통하여 두 API의 변형인 OpenGL SC (Safety-Critical)도 사용 가능합니다. Microsoft는 경쟁 API인 Direct3D를 지정하였습니다. Windows는 보통 인포테인먼트 시스템에서 사용되지만 대부분의 인포테인먼트 시스템은 Linux 기반이며 Khronos API를 사용합니다. Apple 또한 경쟁 API인 Metal을 지정합니다. 현재 Direct3D 또는 Metal의 SC 변형은 없습니다. Metal은 아직 상용 자동차 시스템에 사용되지 않았습니다.

GPU는 더 일반적인 계산에도 사용되므로 Khronos는 계산처리용 (compute) API인 OpenCL을 또한 지정하였습니다. 그들은 Vulkan이라는 API의 대안을 개발했습니다. 이 방법은 그래픽, 임베디드 및 계산처리용 (compute) API를 통합합니다. Vulkan은 적절한 시기에 오래된 "형제자매 (siblings)"를 대체 할 수 있지만 이것은 확실하지 않습니다. - 프로그래밍 모델은 이전 모델보다 훨씬 더 복잡합니다. 그래픽 및 계산(compute)을 단일 API로 통합하려면, 이전에 GPU 드라이버에 캡슐화 된 로직의 많은 부분이 이제 애플리케이션 개발자 또는 UI 도구 및 라이브러리 공급업체에 의해 구현되어야 합니다. 이로 인해 도구 공급업체의 기능 차별화가 더 커지지만 지금까지 하드웨어 종속성에서 끌어냈던 구성요소들에 하드웨어와 관련된 상당한 복잡성이 추가됩니다. 2019년에 Khronos는 Vulkan API의 SC 변형 작업을 시작했습니다.

OpenGL ES와 같은 API를 도입 할 당시 소프트웨어 렌더러는 자동차 SC 환경에 포함하기에 충분한 성능을 발휘하지 못했습니다. 오늘날 그들의 성능은 훨씬 향상되었습니다. GPU를 포함한 FPGA의 가능성은 아키텍처 결정에 또다른 차원을 더해 주었습니다.

MCU는 일반적으로 사용자 인터페이스를 포함하지 않는 "더 작은" 애플리케이션에 사용됩니다. 그들은 이미 제동 시스템과 같은 SC 구성 요소를 위한 ECU의 기초로 자동차 컨텍스트에서 많이 사용됩니다. FPGA와 마찬가지로 MCU 또한 소프트웨어 렌더러가 더 넓은 범위의 차내 애플리케이션에서 사용할 수 있을 만큼 충분히 잘 작동 할 정도로 처리 능력이 향상되었습니다. MCU는 SoC 또는 FPGA보다 전력 요구량이 훨씬 낮습니다. 이것은 또한 비용을 줄일 수 있습니다.

하드웨어 선택을 통해 비용을 절감 할 수 있지만 소프트웨어 라이선스 및 개발과 같은 관련 비용에 미치는 영향을 고려하는 것이 중요합니다.

결론

완전한 레벨 5 자율 주행 차량의 채택에 더 가까워짐에 따라 이것이 안전하게 달성되어야 한다는 것은 분명합니다. 이 문서에 정의 된 여러 가지 이유로 인해 AV연속체의 앞으로 나아갈 길이 명확해지고 있습니다. 우리가 그것을 위해 계속해서 노력하고 있지만 완전한 레벨 5 자동화는 바로 앞의 미래에 있지 않다는 것이 분명합니다. 레벨 5로 직접 신속하고 광범위하게 이동하는 데는 너무 많은 요소가 포함됩니다. 순수한 자율성을 통해 얻을 수 있는 많은 이점이 있지만 자동차

산업은 인프라 문제, 사회 경제적 문제, 인간의 감정, 그리고 가장 중요한 것은 이 문서에서 다룬 기술적 과제를 극복해야 합니다.

우리는 계속 발전해야 하지만 이 문서는 대부분 안전하고 경제적으로 실행 가능한 방식으로 진행해야 하는 필요성을 다룹니다. 모델 기반 설계에 대한 잘 알려진 연구에서 언급했듯이 소프트웨어 프로젝트의 "통합 단계에서 결함을 식별하고 수정하는 데 개발 비용의 80 %가 소비됩니다".⁵

왜 모델기반의 설계가 필요한가?

기회 상실

6 개월 혹은 그 이상 늦게 시장에 출시되면 5 년 ROI의 33 %가 손실이 남습니다. (출처: McKinsey Group)

비용

- 통합기간 동안 결함의 원인 파악 및 수정에 개발 비용의 80%가 사용됩니다.
- 빈약한 요구사항으로 인해 IT 개발 예산의 40% 이상이 소비 됩니다.

전통적인 QA 테스트

- 테스트에 25% - 30% 소요 (출처: IBM Study)
- 열악한 업스트림 품질 수율로 인한 재작업
- 일정 압박으로 인한 상황 악화



Software quality optimization: balancing business transformation and risk, Michael Lundblad, program manager, Rational software, IBM Software Group, Moshe Cohen, offering manager, Rational software, IBM Software Group

초기 비용 때문에 일반적으로 잘 받아 들여지는 원칙은 아니지만 핵심은 적절한 하드웨어 및 소프트웨어 아키텍처와 워크 플로우에 선행 투자하는 것입니다. 궁극적으로 미래의 소프트웨어 설계와 새로운 기술 및 아키텍처를 수용하는 데 필요한 유연성은 비용보다 우선합니다. 위의 연구를 고려할 때, 프로젝트 초기에 적절하게 구현 된 경우 프로젝트 전체 기간에 걸쳐서 상당한

노력과 비용을 절약할 수 있습니다. 임베디드 그래픽 소프트웨어 개발의 주요 목표는 안전이어야 합니다. 그뿐 아니라 적절한 선행 아키텍처 설계를 통하여 비용과 Look and Feel의 균형을 이루는 것도 중요하며, 이는 창의력을 저해하거나 예산을 초과하지 않는 안전한 소프트웨어를 이미 지원하는 기존 COTS 도구를 활용하여 더 수월해 집니다.

저자 정보



Christopher Giordano
Vice President UX/UI Technology
The DiSTI Corporation



Chris는 1997년부터 US Navy 와 University of Central Florida 에서 UI 및 HMI 소프트웨어 개발에 주력해 왔습니다. Chris는 1999년부터 DiSTI사에서 60 개 이상의 다양한 프로그램에서 수석 엔지니어 또는 프로그램 매니저로 근무했으며 DiSTI사의 모든 UI 개발 도구에 대한 제품 매니저 역할을 해왔습니다. Chris는 Boeing, Hyundai, Jaguar Land Rover, Lockheed, NASA, Nissan Motors, Northrop Grumman 및 The Space Ship Company를 위한 DiSTI사의 HMI / UI 프로그램을 관리했으며 현재 DiSTI사의 UX / UI 기술 부사장입니다. 그는 10 년 이상 DiSTI사의 UX / UI 비즈니스를 성공적으로 관리하여 HMI / UI 및 기능 안전 전문가로서 글로벌 리더십 위치를 구축했습니다. Chris는 UNCW에서 재무 학사 학위를, UCF에서 컴퓨터 공학 학사 학위를 취득하여 우등으로 졸업했습니다



Jim Carroll
Chief Technology Officer
Digica



Jim Carroll은 임베디드 시스템에 경험이 많은 기술 이사입니다. Jim 은 25 년 이상 소프트웨어 엔지니어, 기술 설계자 및 CTO 직책으로 수백만 명의 사용자에게 제공되는 대규모 소프트웨어 솔루션을 설계하고 구축했습니다. Jim의 업무 대부분은 반도체 분야에서 이루어졌으며 Intel, Arm 및 Imagination Technologies를 포함한 회사와 협력했습니다. 또한 소프트웨어 표준 그룹 인 Khronos의 기고자이기도 합니다.

