

SURESOFT
Intelligence for Safe World



VISTA

커넥티비티 시험 도구



SURESOFT

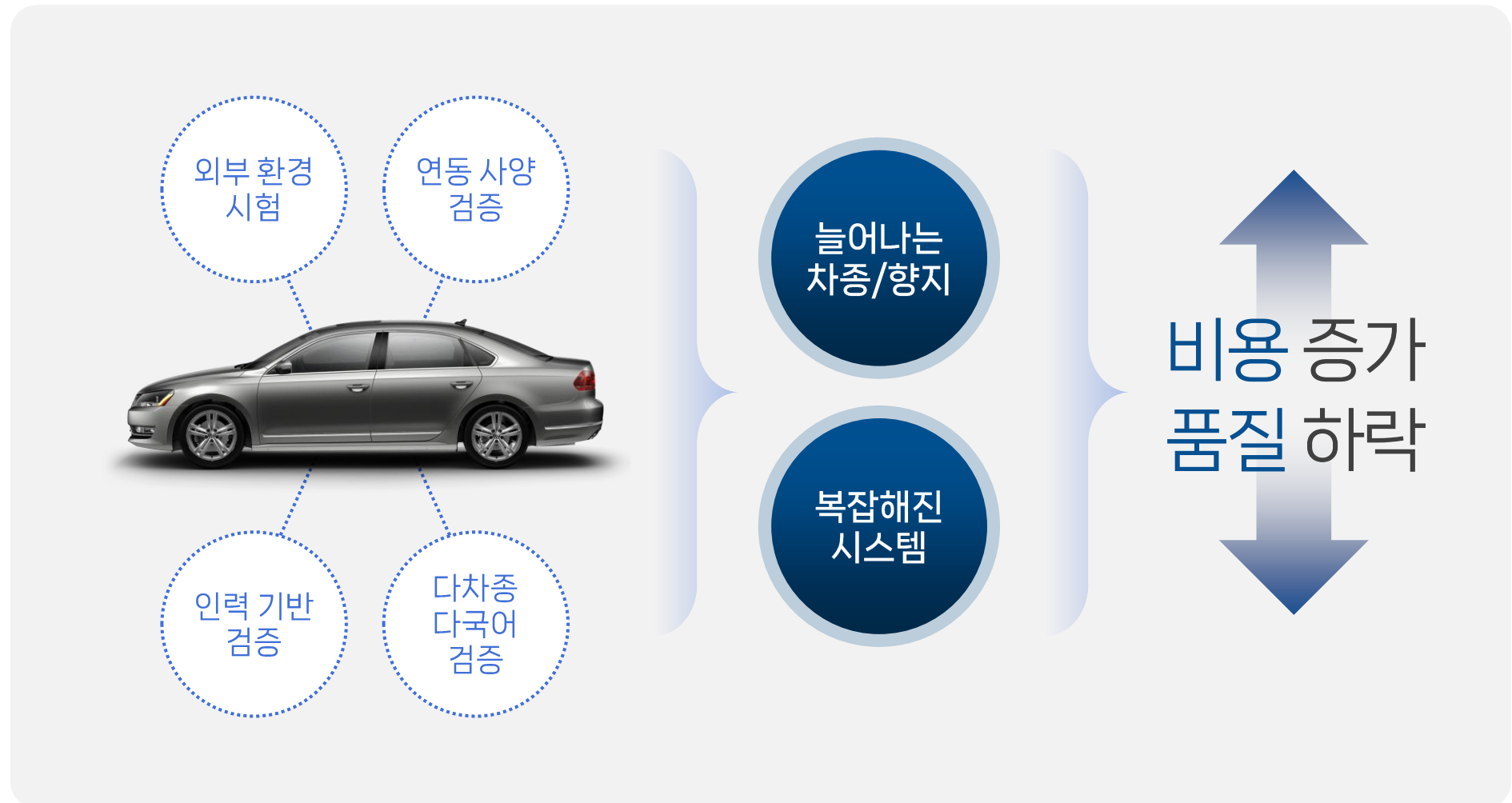
TABLE OF CONTENTS

Why VISTA	3p
Needs	4p
Solution	5p
Features	7p
Specification	23p
References	24p



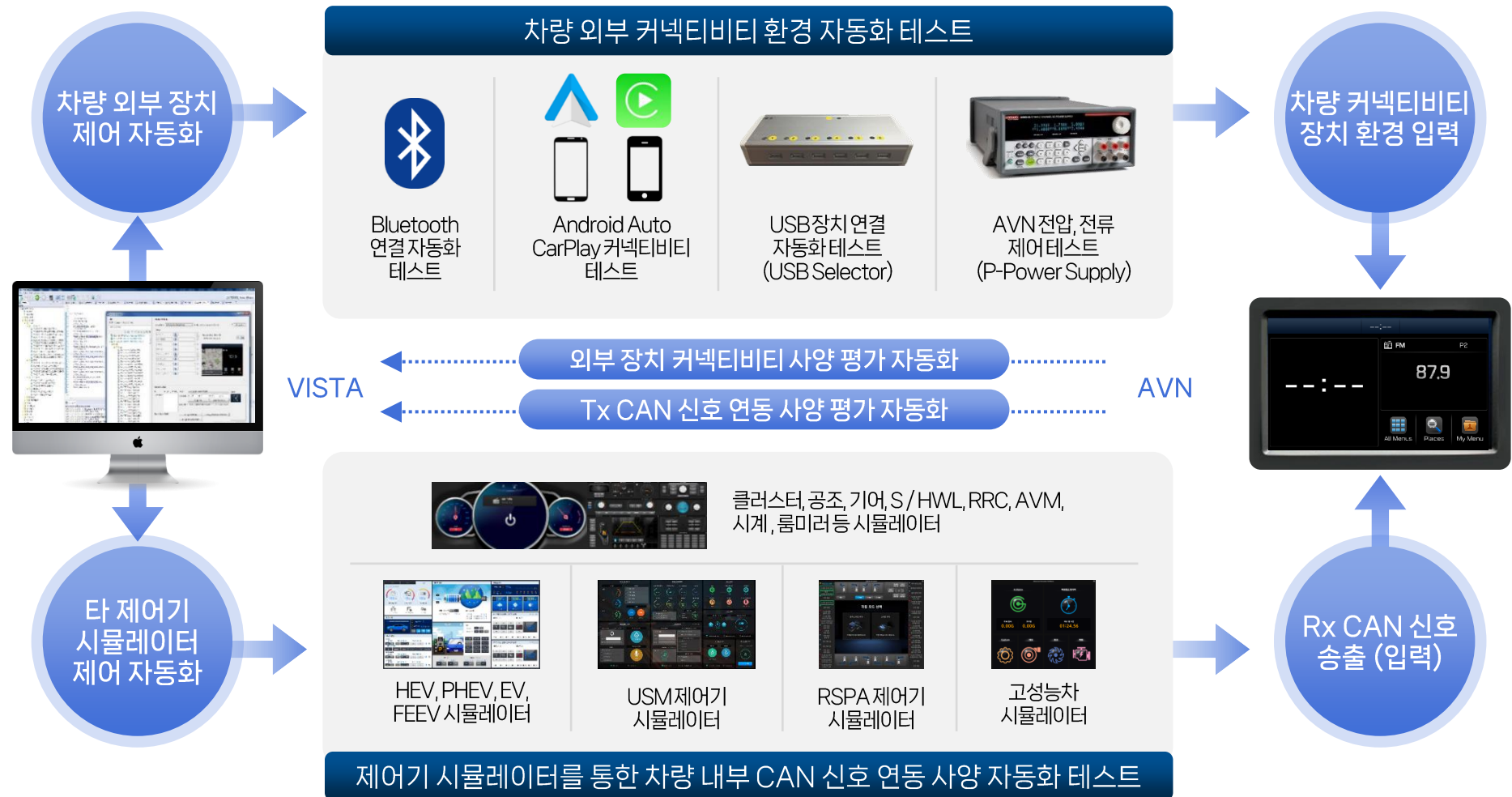
차량 인포테인먼트 시스템의 신뢰성 및 기능성 자동 평가 필요

- 성능 평가, 고장에 대한 복구성 등 신뢰성 평가 필요
- 인포테인먼트의 다양한 기능 평가를 위해 많은 인력 필요 (AVN별 3~4,000여 개의 테스트 케이스)



차량 인포테인먼트 시스템 테스트의 자동화

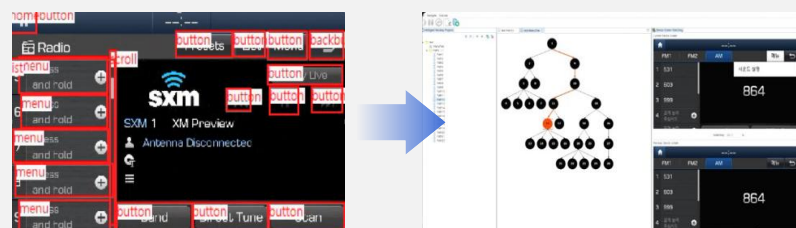
- 차량 외부 커넥티비티 자동화 테스트
- 차량 내부 네트워크 연동 사양 자동화 테스트



단말 메뉴 트리 탐색 및 테스트케이스 자동 생성

- AVN의 메뉴 트리 자동 탐색
- 탐색된 전체 화면을 순회하는 TC스크립트 자동 생성
- 원하는 구간을 실행하는 TC스크립트 자동 생성

01 OCR을 통한 AVN 메뉴 트리 자동 탐색



AVN화면객체 자동 인식 (텍스트기반)

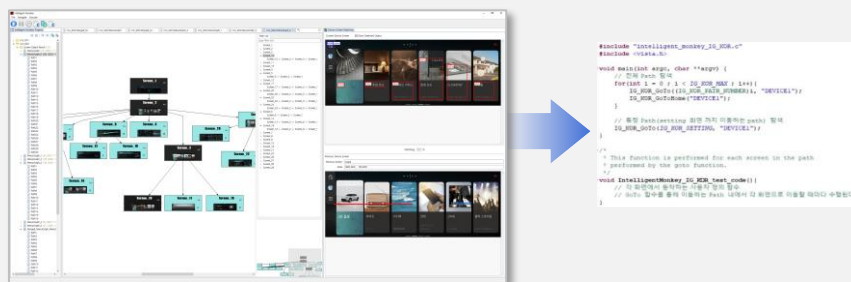
AVN메뉴트리 자동 탐색

02 화면 전환 시간 성능 평가, 오작동 상태 검출



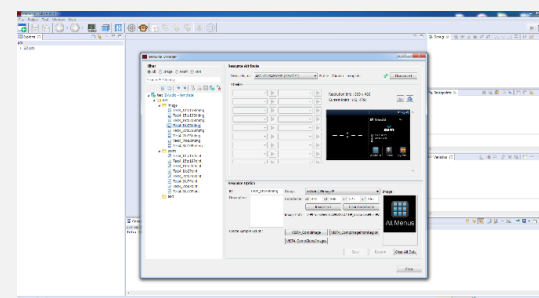
모든 화면 탐색 후
오작동 상태 확인

03 화면 이동 시나리오 검증 스크립트 자동 생성



화면 이동 과정을 검증하는 스크립트 자동 생성

04 리소스원클릭등록



평가를 위한
버튼 이미지,
터치 포인트
리소스
원클릭 등록

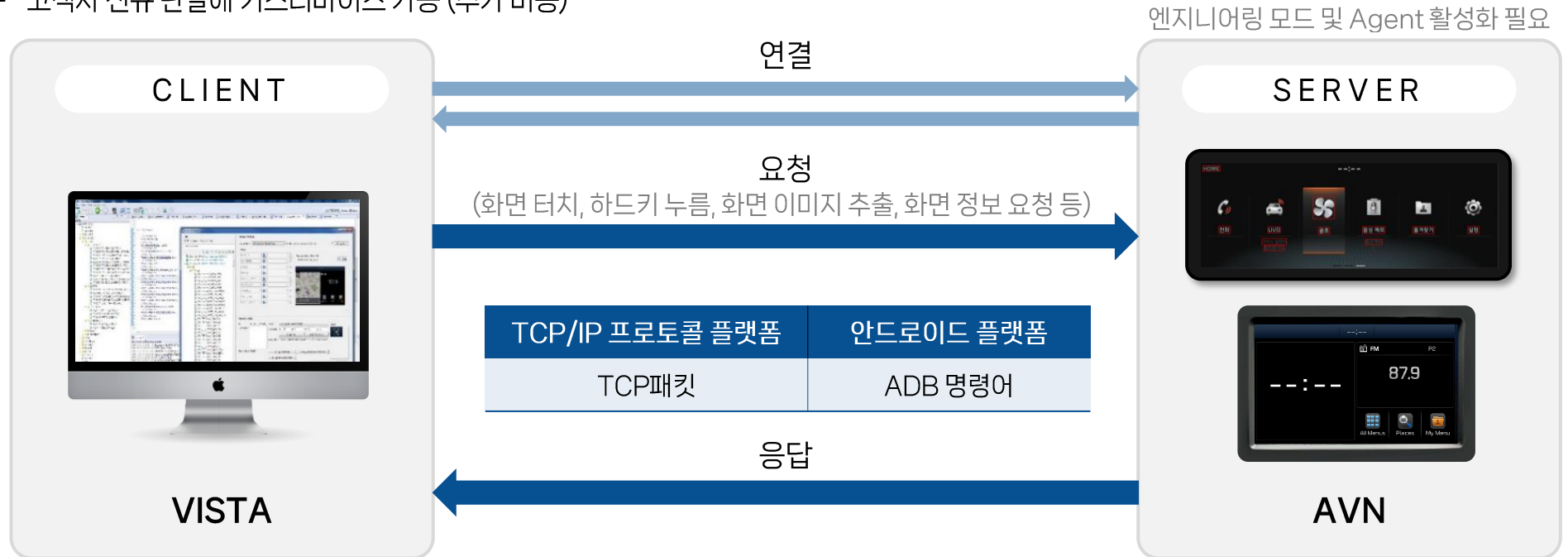
입력 / 판정 자동화 - 개요

- 인포테인먼트 시스템 (AVN) 입력 제어 자동화 및 결과 판정 자동화 테스트



입력 / 판정 자동화 – 연동 프로토콜 지원

- 인포테인먼트 시스템(AVN)과 연동을 위한 자동화 프로토콜 지원
- 고객사 신규 단말에 커스터마이징 가능 (추가 비용)



TCP/IP 프로토콜 방식 패킷 구성 예시

시작 값	패킷 길이	명령어	서브 명령어	응답	요청 데이터	장비 유형	유효성 체크	종료값
0x99	0x04	0x10	0x20	0x00	0x20 0x03	0x01	0x41	0x99

ADB 연동 방식 패킷 구성과 예시

프로그램	단말 IP	명령어	동작 이벤트	동작 유형	동작 요청 데이터	종료값
adb	211.10.20.3	shell sendevent	/dev/input/event0	3 57	250	0

샘플
데이터

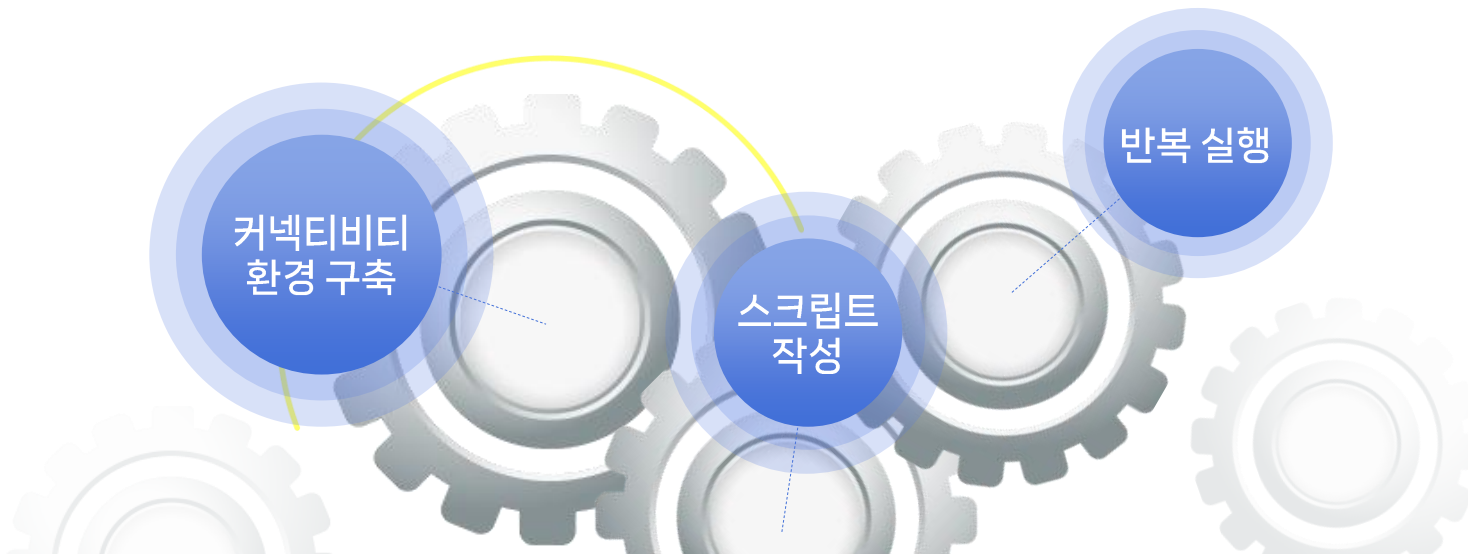
인터페이스 테스트

- 제어기 시뮬레이터를 통한 인터페이스 사양 자동화 테스트



커넥티비티 테스트

- 커넥티비티 테스트 자동화로 검증 업무 효율화



커넥티비티 테스트 스크립트 작성

- Bluetooth테스트
- Call테스트
- Music Play테스트



커넥티비티 테스트 환경 구축

- 커넥티비티 장치를 통해 다양한 조건의 테스트 자동화 환경 구축 및 평가



TcSet(TC 스크립트 세트) 반복 및 일괄 수행

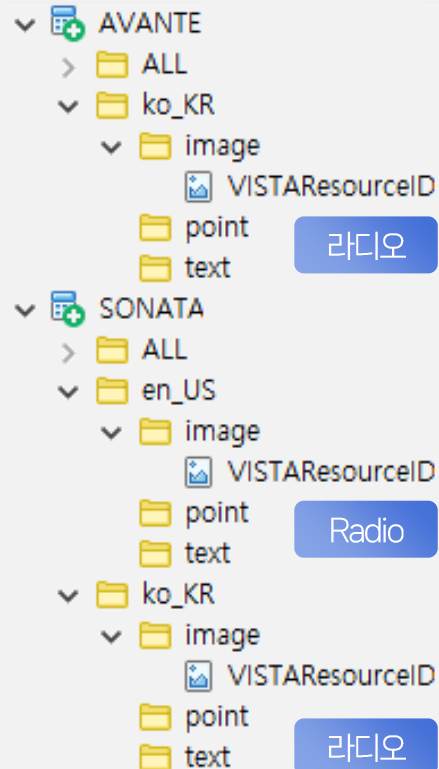
- 반복적인 테스트를 한 번에 수행
- 다차종 조건을 반영한 일괄 수행



다차종 / 다국어 관리

- 다차종 / 다국어 관리로 테스트 케이스(TC) 스크립트 재사용성 극대화
- TC 스크립트에서 커서 위치 리소스 미리 보기를 통해 편리한 유지보수

차종별, 언어별 Resource 관리



Reference

공통 으로 사용되는 TC 스크립트

```
VISTA_stRegionInfo pRegionInfo;  
  
VISTA_eResult result = VISTA_CompImage("VISTAResourceID",0,0,  
    VISTA_eImgCompMode_DEFAULT, VISTA_eRepType_ALL, "DEVICE1");  
if(result != VISTA_eResult_OK){  
    VISTA_GetResCoordinates(pRegionInfo,  
        "VISTAResourceResourceTemplate", VISTA_eResType_IMAGE);  
    VISTA_GetScreenFromRegion("localResourceID",  
        pRegionInfo.nX1, pRegionInfo.nY1, pRegionInfo.nX2,  
        pRegionInfo.nY2,0,VISTA_eImgFmt_PNG, "DEVICE1");  
    VISTA_PrintReport("FALSE",  
        "DEVICE1의 화면에 VISTAResourceID와 일치하는 이미지가 존재하지 않습니다.");  
} else {  
    VISTA_PrintReport("TRUE",  
        "DEVICE1의 화면에 VISTAResourceID와 일치하는 이미지가 존재합니다.");  
}
```

TEST



차종별, 언어별 AVN Test

리소스 교체 기능

- 리소스 교체 기능을 통한 테스트 케이스(TC) 유지보수 시간 단축
: 타겟의 UI가 변경된 경우 리소스를 현재 타겟의 이미지로 일괄 교체

실패한 TC의 Report

No.	Search Name	Search Time	Reference Data	Device Data	Difference Data	Comment	Elapsed Time	Result	Log
1	VEITA_ChildResource	2018-02-12 10:27:26.931					0m 0s 927ms	Fail	
2	VEITA_CompareImage	2018-02-12 10:27:27.874					0m 0s 922ms	Fail	
3	VEITA_CompareImage	2018-02-12 10:27:28.393					0m 0s 922ms	Fail	
4	VEITA_CompareImage	2018-02-12 10:27:29.435					0m 0s 915ms	Fail	
5	VEITA_ChildResource	2018-02-12 10:27:30.265					0m 0s 198ms	Pass	
6	VEITA_CompareImage	2018-02-12 10:27:31.414					0m 0s 955ms	Fail	
7	VEITA_CompareImage	2018-02-12 10:27:32.324					0m 0s 947ms	Fail	

동일한 TC 수행 성공

No.	Search Name	Search Time	Reference Data	Device Data	Difference Data	Comment	Elapsed Time	Result	Log
1	VEITA_ChildResource	2018-02-12 11:01:42.291					0m 0s 955ms	Pass	
2	VEITA_CompareImage	2018-02-12 11:01:43.376					0m 0s 948ms	Pass	
3	VEITA_CompareImage	2018-02-12 11:01:44.340					0m 0s 924ms	Pass	
4	VEITA_CompareImage	2018-02-12 11:01:45.360					0m 0s 924ms	Pass	
5	VEITA_ChildResource	2018-02-12 11:01:46.109					0m 0s 927ms	Pass	
6	VEITA_CompareImage	2018-02-12 11:01:47.769					0m 0s 288ms	Pass	
7	VEITA_CompareImage	2018-02-12 11:01:48.809					0m 0s 972ms	Pass	

Reference Resource 교체

Image Change Confirmation

You can replace the Device Data with the Reference Data. ☒ Check All

Reference Data	Device Data	Change
		☒
		☒
		☒
		☒
		☒
		☒

Tip!
When you select a table,
you can use the keyboard to work comfortably.
Arrow Keys : Switch rows/pages
Alt+a : Click Check All button
Space Bar : Check selected row

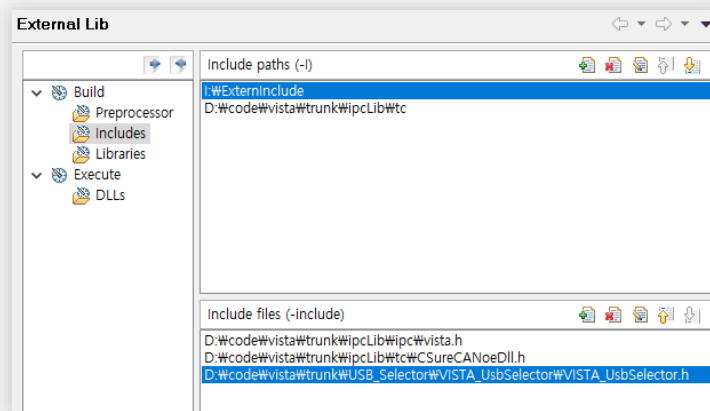
<< < 1 of 2 page > >>

OK Cancel

외부 라이브러리 연동 기능을 활용한 인터페이스 확장

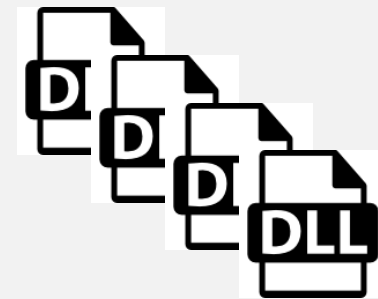
- CAN system의 기존 시뮬레이터 연동 활용 가능 (C API 제공 필요)
- Compile / Link Option 설정 기능을 통한 다수의 외부 라이브러리 참조 용이

VISTA 외부 라이브러리 설정



표준 C++ 컴파일러 기반의 편리한
빌드 설정 기능 제공

외부 라이브러리 (C / C++)



테스트 케이스에 응용할
다수의 라이브러리 참조

Reference

음성 명령 기능 제어 자동화를 통한 단말 동작 검증 용이

- 텍스트 발화(TTS)또는 음원 파일(wav/mp3) 재생을 통한 단말 데이터 입력 및 기능 제어
- 음원 리소스 관리를 통한 다차종/다국어 테스트케이스 작성 시 재사용성 유지
- 단말 화면 및 기능 시험 커버리지 향상



화면 문자열 기반으로 기능 제어와 정합성 검증

- 단말 SW 업데이트의 영향 최소화(이미지의 변경 또는 위치 변경시에도 기존 TC 재사용성 높임)

범주	정상	증상	기존 스크립트 수행 시 다른 기능이 실행됨	예시	[음성 메모]를 수행했지만 [검색/경로]기능이 수행됨
실행 메뉴					
	업데이트 전 수행한 텍스트: 음성 메모 업데이트 후 기대 화면				

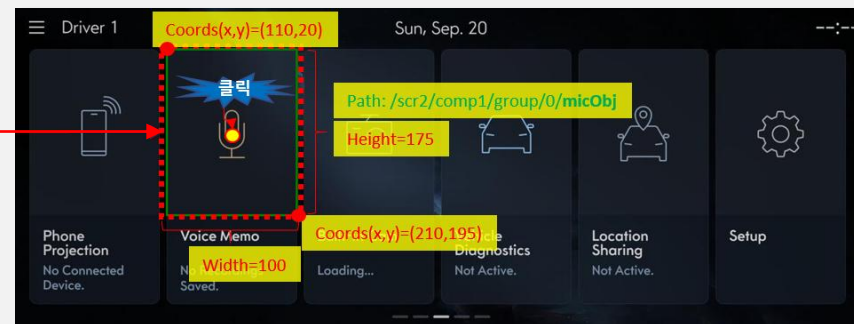
범주	정상	증상	기존 스크립트 수행 시 다른 기능이 실행됨	예시	[음성 메모]를 수행했지만 [검색/경로]기능이 수행됨
VISTA 도구 수행 화면					
	음성 메모 메뉴 위치가 변경되어도 "음성 메모" 문자열 위치를 자동으로 찾고 클릭하여 정상 제어 가능 제어 성공 제어 성공				

화면상의 UI객체 정보를 기반으로 기능 수행 및 정합성 검증

- SW의 UI 업데이트에 영향 받지 않고 기존 TC 재사용
(Android Automotive 플랫폼 지원. 그 외의 UI객체 지원 단말에 대해서는 별도 포팅 필요)

실행 제어 전 이전 화면

Object 정보	
name	micObj
X	110
Y	20
Width	100
Height	175

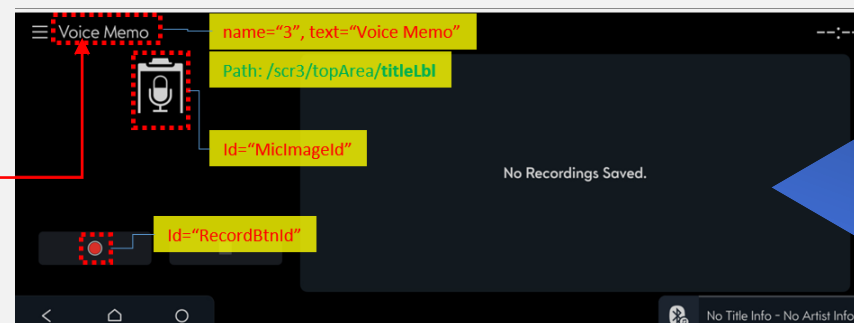


TC Script

제어 {
 VISTA_FetchScreenInfo("Device1", VISTA_eRepType_ALL) { // 단말 현재 화면 객체 정보 취득 }
 VISTA_stElement micObj = VISTA_GetElement("/scr2/comp1/group/0/micObj", "Device1") { // 마이크 객체 취득 }
 VISTA_ClickByElement(micObj, 100, "Device1"); // 해당 Id를 갖는 컴포넌트의 bound상의 중앙 포인트 클릭 제어
 검증 {
 VISTA_stElement titleTextObj = VISTA_GetElement("/scr3/topArea/titleLbl", "Device1") { // 타이틀 텍스트 객체 취득 }
 char * expectedTextList [2] = {"text", "Voice Memo", "visible", "true"}; // key,value의 2개 문자열을 1쌍으로 입력
 if (! VISTA_IsElementToMatchValue(titleTextObj, expectedTextList, VISTA_eComplListMatch_ALL, "Device1")) {
 // 주어진 Id 리스트에 해당하는 컴포넌트의 존재 여부 체크. VISTA_eComplListMatch_ANY는 1개라도 있으면 성공 }
 }
 }

이후 화면에서 동작 검증

Object 정보	
name	titleLbl
text	Voice Memo
X	110
Y	20
Width	100
Height	175



키패드 문자열 입력 자동화

- 키패드 유형별 프로파일 관리 (각 키 리소스는 최초 수동 구축 필요)
- 입력 문자열을 초성/중성/종성으로 분해하여 자동 클릭
- 검색 기능 시험 자동화 가능

키패드 입력 테스트 케이스 예시

```
#include <vista.h>

void main(int argc, char **argv) {
    char* dev = "DEVICE1";

    VISTA_ClickKeypad("서울 강남구 대치동 943-26", "ccnc_keypad.txt", 500, true, true, VISTA_eRepType_DEFAULT, dev);
}
```

띄어쓰기, -(대수)

입력창 초기화

확인버튼 수행

프로그램상에서 키패드 문자열 입력 제어

파라미터로 요청된 문자열을 분해 후 해당 키 자동 입력

검색

확인

검색 결과 (1)

대치동 943-26
서울 강남구 대치동 (대치2동) 943-26

5.0 km

서울시 서초구 부근

50 m

gs25 까치산

```
VISTA_ClickKeypad("gs25 까치산", "ccnc_keypad.txt", 500, true, false, VISTA_eRepType_DEFAULT, dev);
```

영어, 쌍자음 수행

닭과오리

```
VISTA_ClickKeypad("닭과오리", "ccnc_keypad.txt", 500, true, false, VISTA_eRepType_DEFAULT, dev);
```

이중자음, 이중모음 수행

다양한 언어로 TC스크립트 작성 및 유지보수

- 원하는 언어로 편리하게 TC 스크립트 작성 및 관리 (C, Java, Python 지원)
- 객체 지향 TC 프로그래밍
- 재사용성 강화

화면 좌표 클릭 API 예시

C/C++

```
VISTA_eResult VISTA_ClickCoordinates  
(int x, int y, boolean isLongClick, long longClickMs, char* deviceId)
```

python

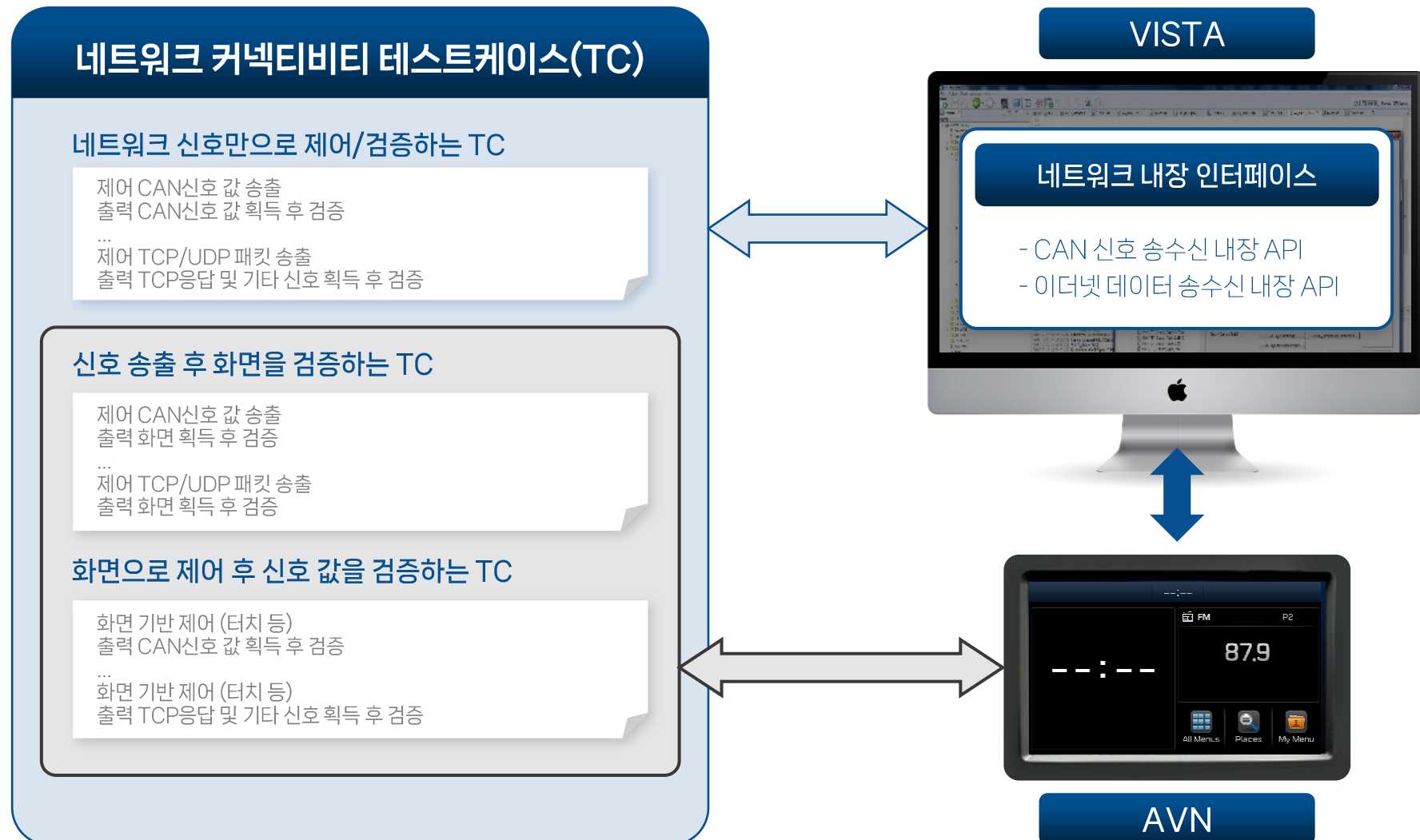
```
class VISTA  
def VISTA_ClickCoordinates  
(x, y, isLongClick, longClickMs, deviceId)
```

Java

```
class VISTA  
VISTA_eResult VISTA_ClickCoordinates  
(int x, int y, boolean isLongClick, boolean longClickMs, String deviceId)
```

CAN 신호와 Ethernet 데이터의 송수신 및 모니터링 기능 내장화

- 내장화되는 PCAN 연동 인터페이스를 통해 간편하게 CAN 시그널 데이터 송수신 (Peak-System PCAN기반)
- 화면 기반 검증 뿐 아니라 눈에 보이지 않는 내부 신호 데이터 검증 사용성 강화
- 이더넷 데이터 기반 검증 지원



단말 직접 제어를 통해 TC 스크립트 작성

- 스크립트 언어에 익숙하지 않아도 손쉽게 TC 생성 가능
- 단말 직접 조작은 Android Automotive OS만 지원, 그 외에는 VISTA 도구를 통한 단말 조작 지원 (화면 캡처 후 제어 방식)
- 단말 제어 후 화면 비교 검증 코드 자동 추가

사용자의 단말 조작



사용자 액션에 대한 동일 액션 동작 API 코드 자동 생성

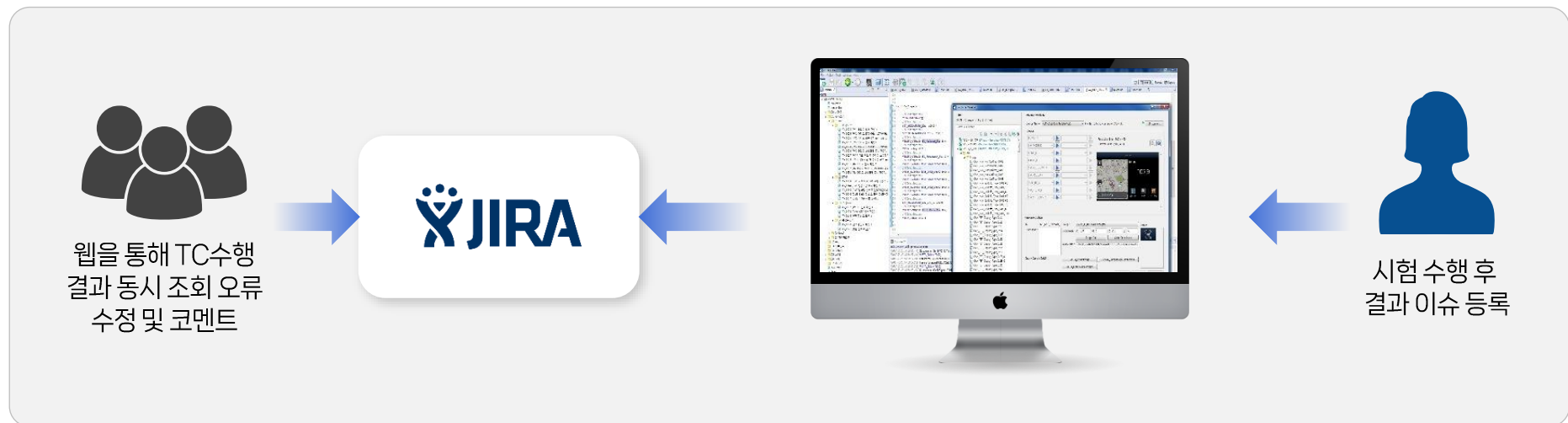
Recoding1TC.java

```
Session testSession = new TestSession("Device1");  
...  
testSession.VISTA_ClickCoordinate (600, 600, 0);  
testSession.VISTA_Delay (1500);  
testSession.VISTA_ClickCoordinate (120, 50, false, 0);  
testSession.VISTA_SwipeTo (500, 70, 800, 70);  
...
```

SW 업데이트 시 기존 기능 정상 동작 자동 확인을 위한
Regression Test 수행 등에 활용

테스트케이스 검증 수행 결과 공유

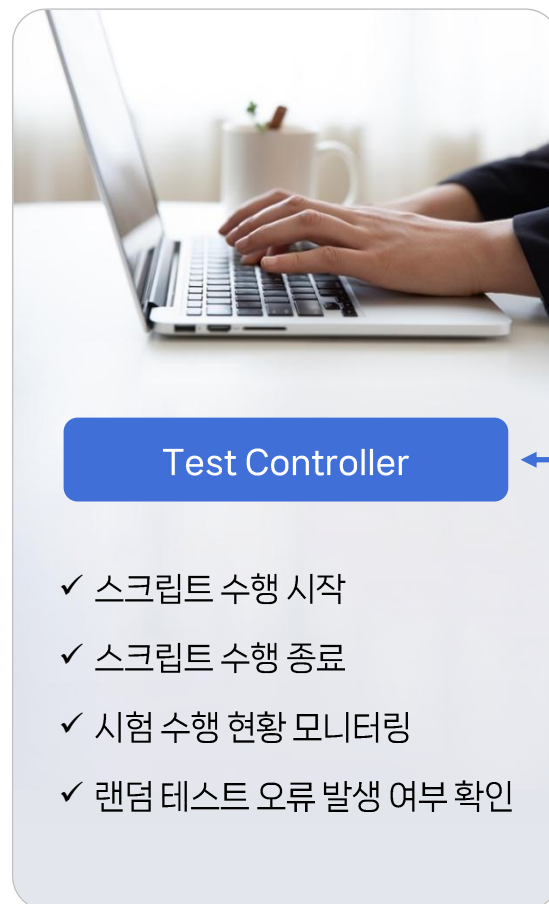
- 이슈 관리 시스템(Jira) 연동을 통해 테스트케이스의 수행 결과 중앙 관리
- 테스트 수행 결과 보고서 자동 업로드



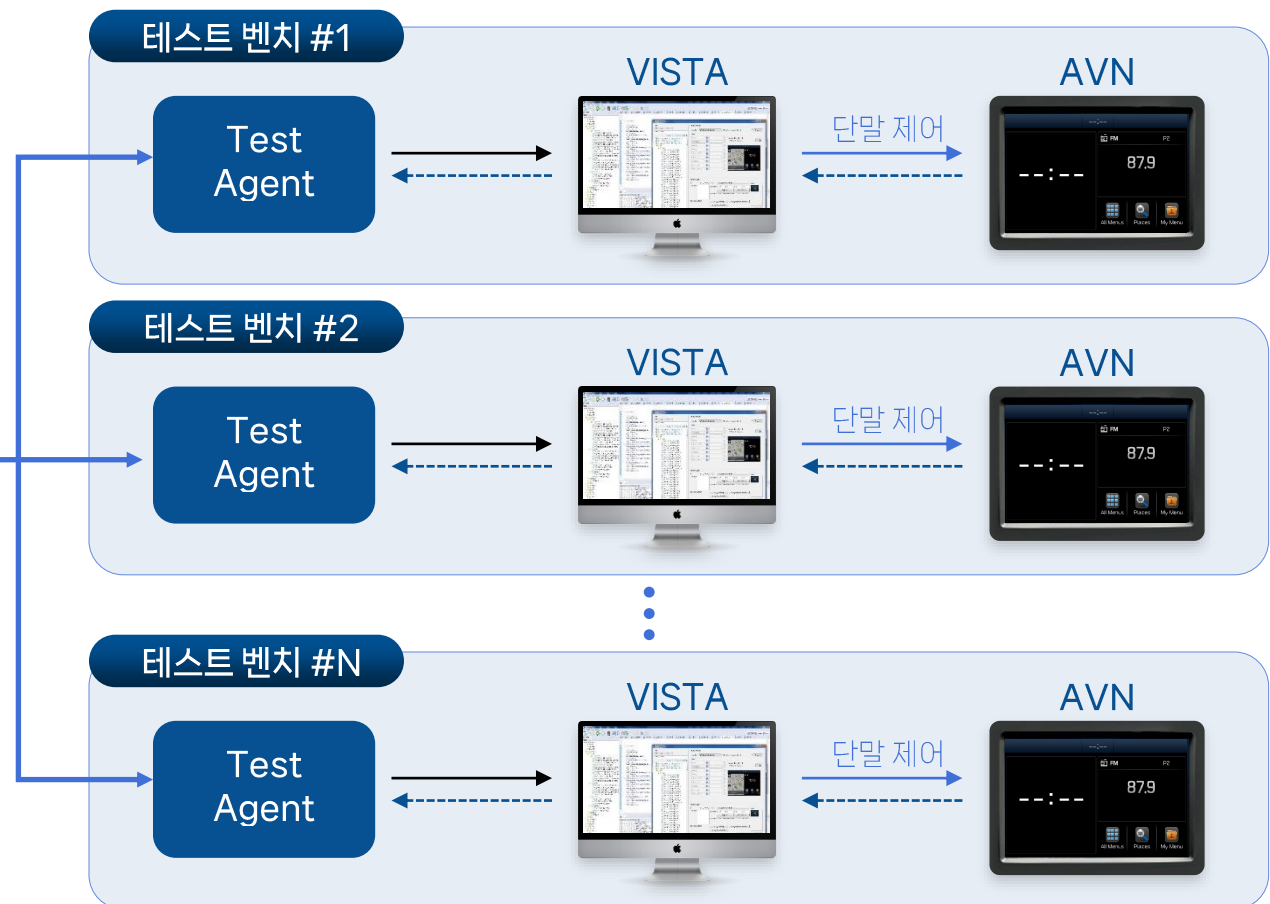
원격 벤치 제어 및 모니터링

- 여러 테스트 벤치의 단말 시험 제어 및 확인
- 사무실 자리에서 원격지의 시험실 내 여러 단말의 시험 제어와 진행 상황 모니터링 및 결과 열람

사무실 연구원 개인 환경 (Local)



원격 환경 (Remote)



I 상세 환경

구분	세부 사항
OS	- Windows 7 x32 / x64 K 버전 이상
실행 환경	- Microsoft Visual C++ 2013 Redistributable
지원 언어	- 영어
자동화 지원 스크립트 언어	- 표준 C, C++ 지원
외부 장치 제어 (확장 가능)	- Vector CANoe 장비 연동 - PEAK CAN 장비 연동 - 안드로이드 스마트폰 (v4.1 이상) - Windows 호환 웹캠 - PPS (Programmable Power Supply) - USB Selector

I 하드웨어 권장 사양

구분	CPU	RAM	HDD
PC	Quad Core 2GHz 이상	2GB 이상	2GB 이상

※ 도구 단독 동작 시 권장 사양으로, 추가 프로그램과 함께 사용할 경우 사양 추가가 필요할 수 있습니다.



미국 자동차 H사 적용 사례 – AVN 시스템 품질 확보 목적

도입 배경

- 신규 차종이 발생할 때마다 반복적인 AVN 시스템 시험의 어려움
- 사양 변경으로 인한 AVN 시스템의 불충분한 테스트

2016년
VISTA를 도입해
시험 자동화
시스템 구축

결과

- VISTA 설치 및 외부 조건 자동화를 위한 장치 공급
- 자동화 테스트 케이스 스크립트 개발
- 자동화 스크립트 개발을 위한 교육

H사 Engineers



슈어소프트테크 Engineers



테스트 시나리오

리뷰

피드백

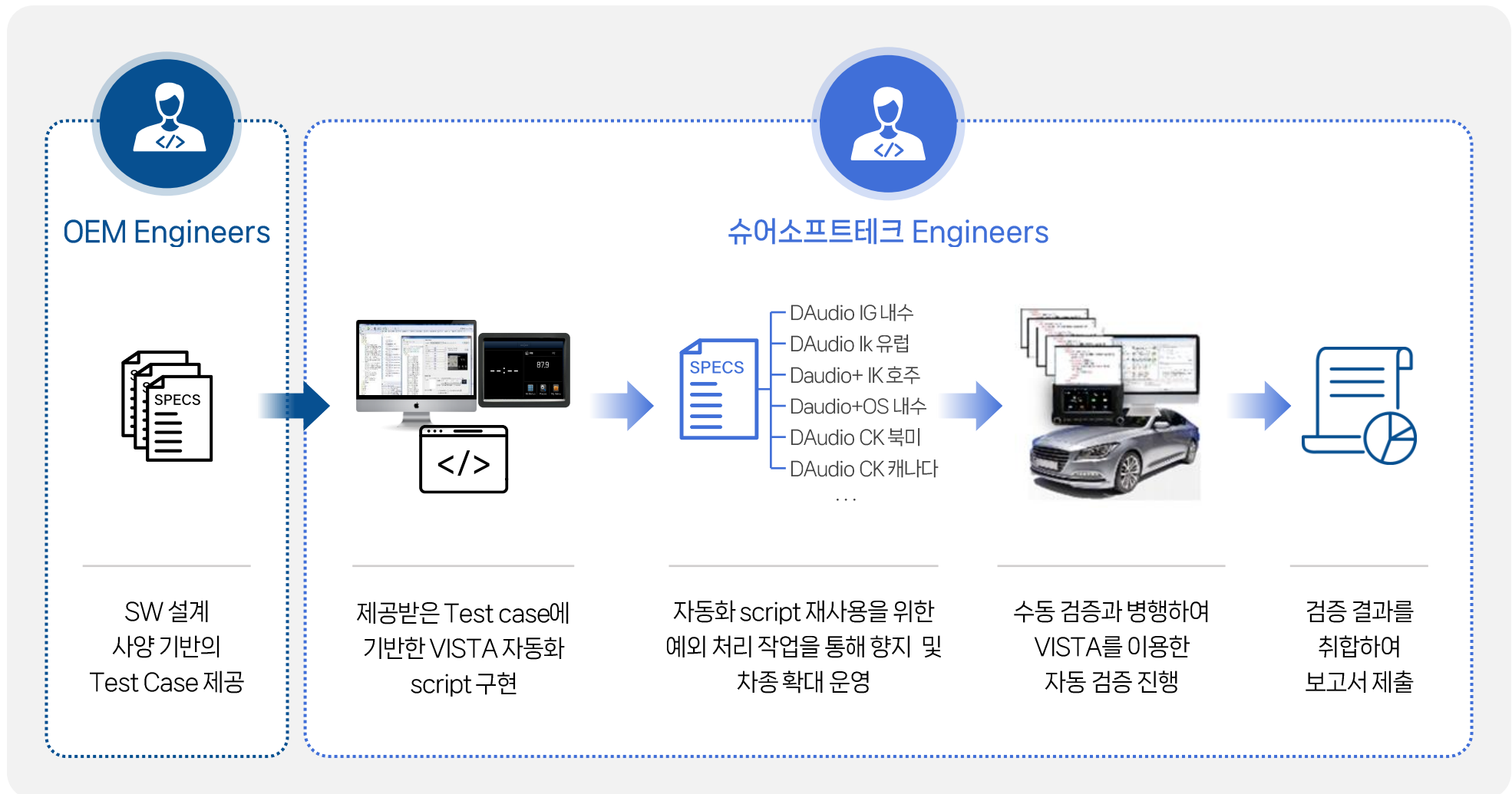
추가적인
테스트
시나리오 개발

자동화 테스트 케이스
스크립트 개발



Example AVN OEM 사례 – 검증 용역 서비스 제공

- VISTA를 도입하여 AVN 수동 검증을 자동검증으로 대체하는 운영 서비스 제공
- VISTA를 이용한 다차종 및 여러 향지의 AVN 자동 검증 운영 서비스 제공



자동차 인포테인먼트(AVNT, 카오디오) 평가 기술 개발 및 검증 사례

- VISTA 를 활용한 검증 자동화 평가 기술 개발



고객사	과제명	년도
Hyundai Motors	Upgrade of the VISTA	2022
Hyundai MOBIS	UI Object Based TC API and Resource Management	2022
HATCI, V-cubed Solutions	Advanced IM GUI Regression Test	2021
HATCI, V-cubed Solutions	IM GUI Regression Test	2020
Hyundai Motors	Development of the VCRM Data integrity Verification Automation Tool	2020
Hyundai Motors	Development of Acceptance Test Automated Evaluation SW	2019
Hyundai MOBIS	AVN SW Verification Automated Script Expansion	2019
Hyundai Motors	Customer-using Scenario-based AVN Verification Technology Development	2017
Hyundai Motors	Multimedia CAN Automation Evaluation Technology for External Signal Display Verification	2016



Thank You