Efficient/Optimized Test suites for Automotive Software using MATLAB

Manjushree Sathyanathan
and Asha Vaddadi

www.continental-corporation.com

P ES SE EPC Exhaust Aftertreatment

Software Testing in Global Industry

Need for the Hour

- Bridging Gaps between Developers and Testers in Globally distributed Software Development network

Future Paradigms – Author's Vision

- We need to rethink the way testing is performed in distributed development environments
- We present a possible research agenda (Framework) that would help address these identified issues like function and requirement testing when applied to Model Based Design which would give a improved quality of software

Test-Case Design Methodologies

Black Box

- Equivalence Class Partitioning (ECP)
- Boundary-Value Analysis (BVA)
- Choice Relationship Framework (CRF)
- Predicate Testing
  - Boolean Operator(BOR)
  - Boolean Relational Expression(BRE)

White Box

- Statement Coverage
- Decision Coverage
- Condition Coverage
- Multiple-Condition Decision-Condition Coverage(MCDC)

Solution Blueprint : Combination of Test Framework-CTF

Step1. Applying ECP and BVA

Select test data such that Boundary value occurs at least once in test inputs

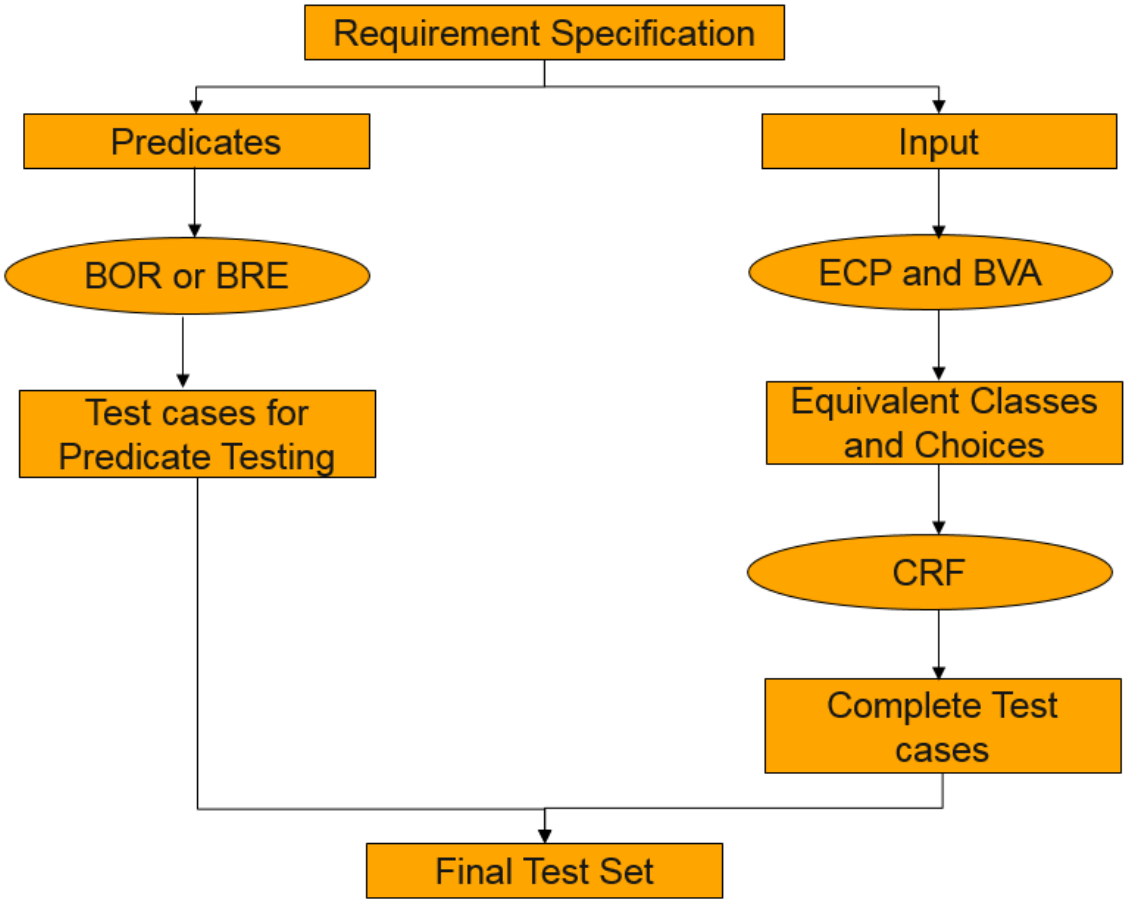
Step2. Applying Choice Relation Framework

Treat each parameter as a category and equivalent classes as its choices

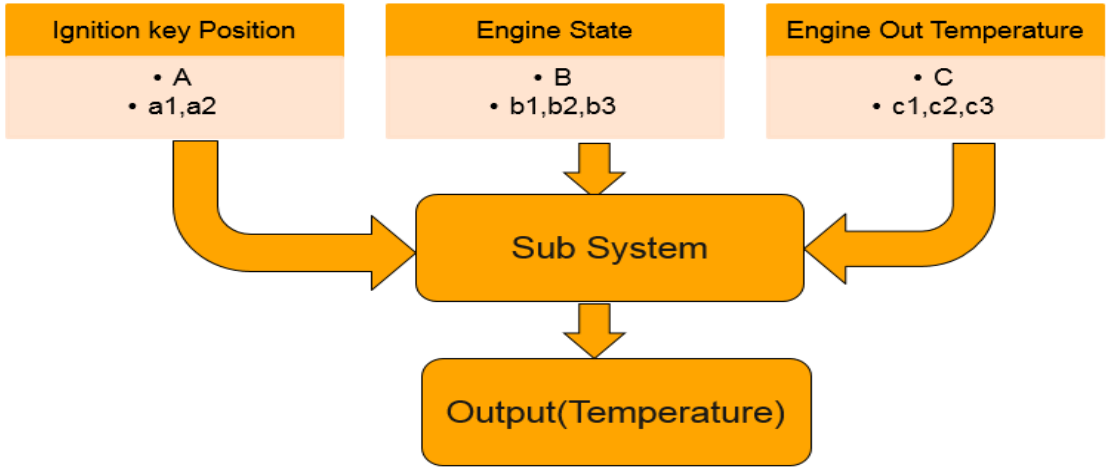
Step3. Union with the results of predicate testing

Union the test cases generated by CRF with test cases generated by BOR or BRE

Process flow of CTF



Case Study : Determination of Upstream Catalyst Temperature



	Variable Name	Meaning	Value Type	Value Range	Category	Choice	
1	IgkPos_flag	Ignition key position	Unsigned Integer	{0…1} (0…1)	A	a1	0
						a2	1
2	EngSt_u8	Engine State	Unsigned Integer	{0,1,2} (STOP, START, RUNNING)	B	b1	STOP
						b2	START
						b3	RUNNING
3	EngTmp_u16	Engine Out Temperature	Unsigned Integer	{0…32767} (-273.15…1500)	C	c1	-273.15
						c2	20… 650
						c3	1500

Choice Relation Table without considering requirements

	a1 (0)	a2 (1)	b1 (STOP)	b2 (START)	b3 (RUNNING)	c1 (-273.15deg)	c2 (20..650deg)	c3 (1500deg)
a1(0)	FE	NE	PE	PE	PE	PE	PE	PE
a2(1)	NE	FE	PE	PE	PE	PE	PE	PE
b1(STOP)	PE	PE	FE	NE	NE	PE	PE	PE
b2(START)	PE	PE	NE	FE	NE	PE	PE	PE
b3(RUNNING)	PE	PE	NE	NE	FE	PE	PE	PE
c1(-273.15deg)	PE	PE	PE	PE	PE	FE	NE	NE
c2(20..650deg)	PE	PE	PE	PE	PE	NE	FE	NE
c3(1500 deg)	PE	PE	PE	PE	PE	NE	NE	FE

21 Test cases generated based on choice relation table without considering requirements

Relations
FE : Fully Embed (Invalid)
PE : Partially Embed
NE : Not Embed (Invalid)

Choice Relation Table considering requirements

	a1 (0)	a2 (1)	b1 (STOP)	b2 (START)	b3 (RUNNING)	c1 (-273.15deg)	c2 (20..650deg)	c3 (1500deg)
a1(0)	FE	NE	NE	NE	NE	NE	NE	NE
a2(1)	NE	FE	PE	PE	PE	PE	PE	PE
b1(STOP)	NE	PE	FE	NE	NE	**PE**	**PE**	**PE**
b2(START)	NE	PE	NE	FE	NE	**PE**	**PE**	**PE**
b3(RUNNING)	NE	PE	NE	NE	FE	**PE**	**PE**	**PE**
c1(-273.15deg)	NE	PE	PE	PE	PE	FE	NE	NE
c2(20..650deg)	NE	PE	PE	PE	PE	NE	FE	NE
c3(1500deg)	NE	PE	PE	PE	PE	NE	NE	FE

Keeping Ignition Key constant at ON(1) position

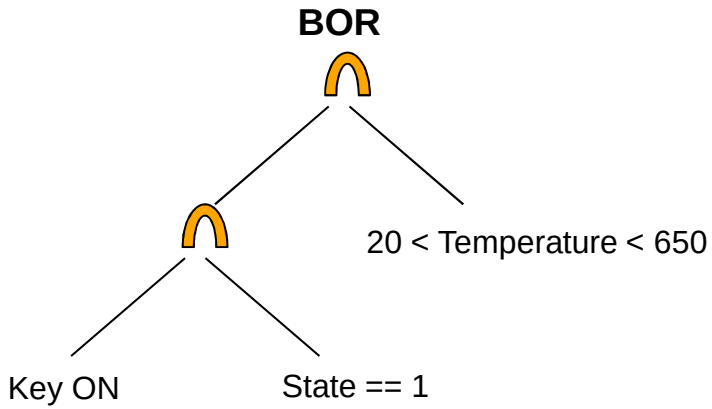
	c1 (-273.15deg)	c2 (20..650deg)	c3 (1500deg)
b1(STOP)	**PE**	**PE**	**PE**
b2(START)	**PE**	**PE**	**PE**
b3(RUNNING)	**PE**	**PE**	**PE**

The number of generated test cases is reduced to 9

	Ignition key	Engine state	Engine out temperature
1	1	STOP	<20
2	1	STOP	20… 650
3	1	STOP	>650
4	1	START	<20
5	1	START	20… 650
6	1	START	>650
7	1	RUNNING	<20
8	1	RUNNING	20… 650
9	1	RUNNING	>650

Predicate testing : Boolean Operator - BOR

S={operators(!,&,|,….)}
For each operator : true or false



	Ignition key	Engine state	Engine out temperature	If/else	TC
1	T	T	T	if	Y
	T	T	F	else	Y
2	T	F	T	else	Y
	T	F	F	else	
3	F	T	T	else	Y
	F	T	F	else	
4	F	F	T	else	
	F	F	F	else	

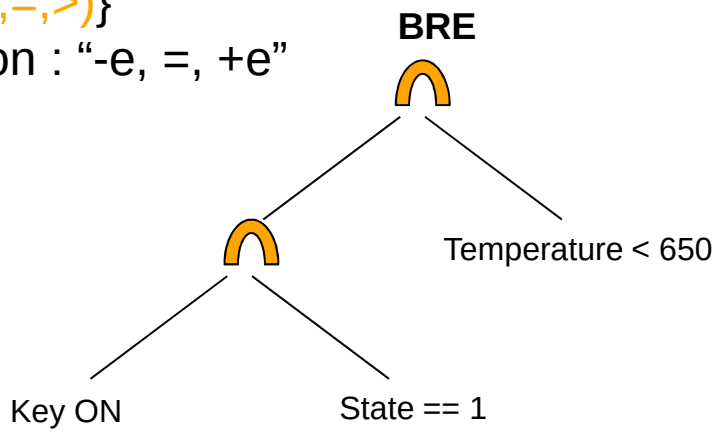
If a predicate contains *n*-Boolean operator, then the maximum size of the BOR adequate test set should be *n*+2

4 test cases generated which completely cover the MC/DC coverage (Minimum number of test cases)

Boolean and Relational Expression - BRE

By keeping Ignition key Constant (ON) predicates are
Pr:((state==1)&&(Temperature<650))

S={Relations(<,=,>)}
For each relation : “-e, =, +e”

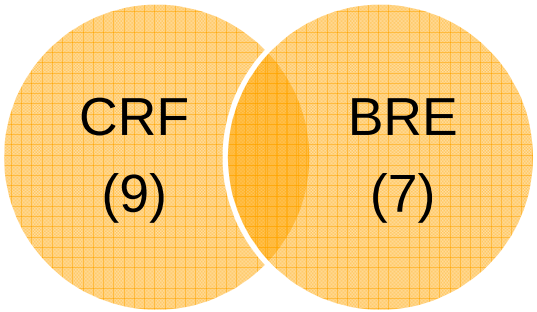


NO	Ignition Key	Engine State	Engine out Temperature	If/else	TC
1	=	=	=	else	Y
	=	=	+e	else	Y
	=	=	-e	if	Y
2	=	+e	=	else	
	=	+e	+e	else	
	=	+e	-e	else	Y
3	=	-e	=	else	
	=	-e	+e	else	
	=	-e	-e	else	Y
4	+e	=	=	else	
	+e	=	+e	else	
	+e	=	-e	else	Y
5	+e	+e	=	else	
	+e	+e	+e	else	
	+e	+e	-e	else	
6	+e	-e	=	else	
	+e	-e	+e	else	
	+e	-e	-e	else	
7	-e	=	=	else	
	-e	=	+e	else	
	-e	=	-e	else	Y
8	-e	+e	=	else	
	-e	+e	+e	else	
	-e	+e	-e	else	
9	-e	-e	=	else	
	-e	-e	+e	else	
	-e	-e	-e	else	

If a predicate contains *n*-Boolean operator, then the maximum size of the BRE adequate test set should be 2*n*+3

7 test cases generated by predicate testing is an efficient solution for the test optimization

Comparison of CRF and BRE



BRE

NO	Ignition Key	Engine State	Engine out Temperature	If/else	TC
1	=	=	=	else	Y
2	=	=	+e	else	Y
3	=	=	-e	if	Y
4	=	+e	-e	else	Y
5	=	-e	-e	else	Y
6	+e	=	-e	else	Y
7	-e	=	-e	else	Y

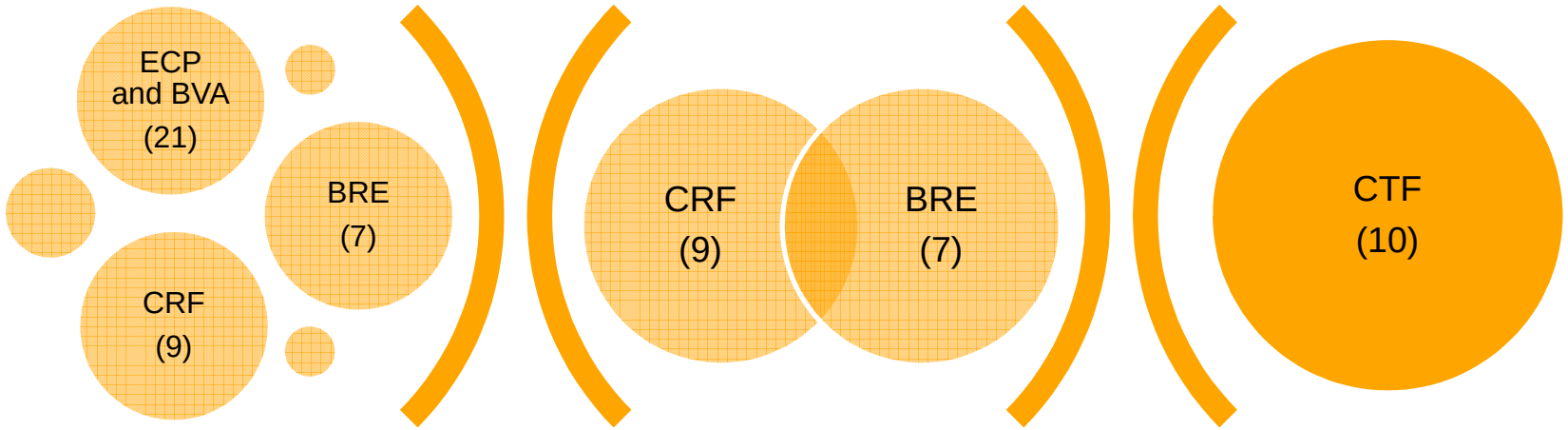
	Ignition key	Engine state	Engine out temperature
1	1	STOP	<20
2	1	STOP	20… 650
3	1	STOP	>650
4	1	START	<20
5	1	START	20… 650
6	1	START	>650
7	1	RUNNING	<20
8	1	RUNNING	20… 650
9	1	RUNNING	>650

CRF

NO	Ignition Key	Engine state	Engine out Temperature	If/else	TC
1	=	-e	-e	else	Y
2	=	-e	=	else	Y
3	=	-e	+e	if	Y
4	=	=	-e	else	Y
5	=	=	=	else	Y
6	+e	=	+e	else	Y
7	-e	+e	-e	else	Y
8	=	+e	=	else	Y
9	=	+e	+e	else	Y

By Comparing CRF (9) and BRE(7) we eliminate duplicate test cases and consider the required test cases. Total 10 cases generated using CTF

Union of CRF with BRE

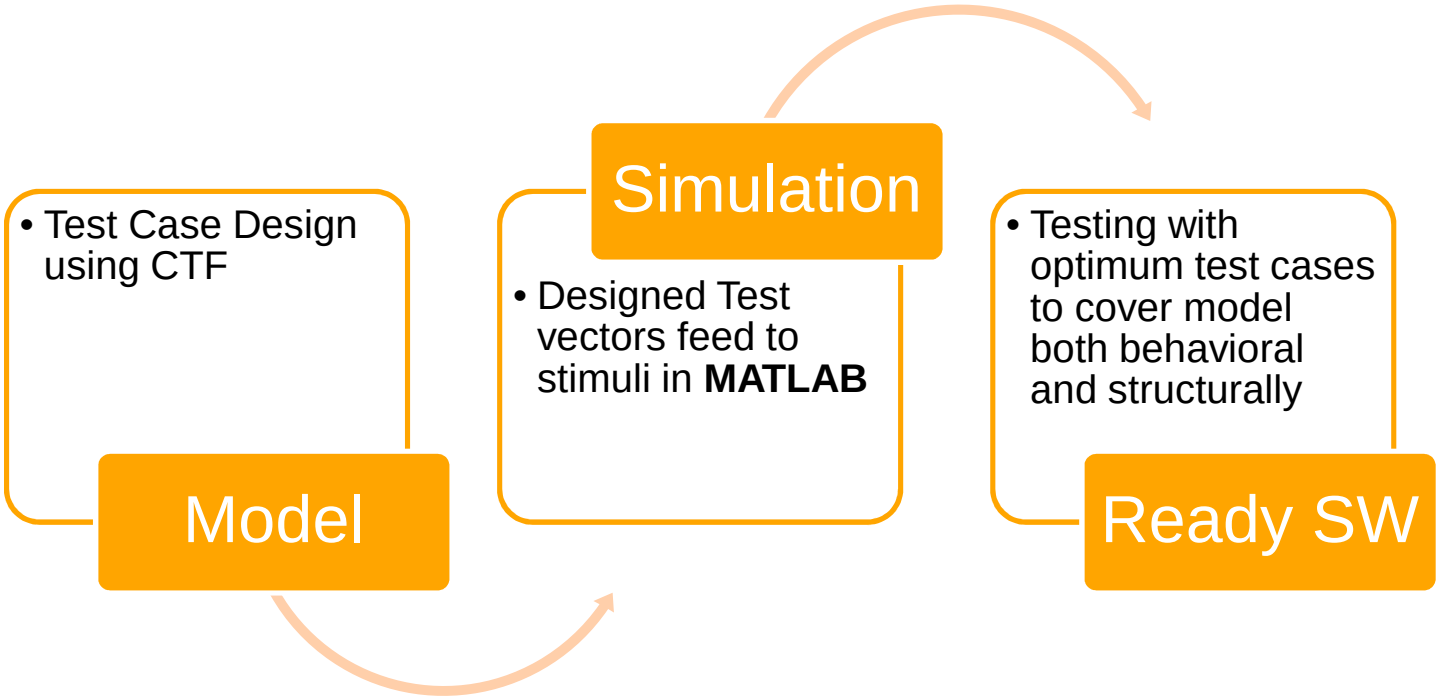


1. 21 Test cases generated based on choice relation table without considering requirements
2. The number of generated test cases is reduced to 9 with CRF
3. 7 test cases generated by BRE

1. Comparing the generated test cases from predicate testing and choice relation framework, duplicated test cases are eliminated

1. The final total number of test cases by Combination of Test Framework is 10
2. 52% of test cases reduction is achieved with increased testing efficiency

Implementation



Results

Sample	Inputs	ECP/BVA (Choice)	Test Cases				Reduction Rate(%)
			CRF		BRE	CTF	
			w/0 Req	w Req			
Torque Limitation	2	6	9	2	5	6	33%
Temperature Determination	3	8	21	9	7	10	52%
Speed Limitation	4	13	61	17	9	21	65%

Restriction of Input Domain

- Our recommendation for reasonable construction of choice relation table is to limit its maximum number of choice under 20.

Conclusion

The effective test cases generated to cover both structural and behavioral test scenarios

Test cases reduction of around 40% to 60% is achieved with increased testing efficiency

It is seen that efficiency of test case creation increases as we move towards the optimum test cases with the framework

This approach is based upon standardization of test cases as these generated by Predicate testing and Choice Relation framework

Acknowledgement

- We thank the management of Continental Technical centre India for the encouragement. Authors would like to especially thank Mr Girish Ramaswamy, General manager Engine System Technical Centre India.
- Authors would like to thank Mr Ravi Ponnusamy, Head of EPC.
- Authors would like to thank Mr Venkatasubramaniyan Kuppusamy, Group Leader P ES SE EPC E and SCR Controls.
- Authors would like to thank Mr Vivek Venkobarao and Mr Rohith Kamath for the guidance and support.

Reference

Kuppusamy, V., B M, Aparna., and Venkobarao, V., "A Modified Framework for Reliable Testing of Model Based System in Automotive Industry," SAE Technical Paper 2011-28-0138, 2011, doi:10.4271/2011-28-0138.

Ruizhi Gao, Shou-yu Lee., “Case Study of Requirements-based Test Case Generation on an Automotive Domain”, IEEE International Conference on Software Quality and Reliability and Security, 2015, doi: 10.1109/QRS-C.2015.34

Ruizhi Gao,Jeong Soo Eo W.Eric Wong, Xinwei Gao and Shou-Yu Lee. “An Empirical Study of Requirement-based test generation on an Automobile Control System”.in proceedings of the SAC2014 Conference, Gyeongju, Korea, March 24-28,2014pp.1094-1099.

T.Y. Chen, P.L. Poon, and T.H. Tse, “A choice relation framework for supporting category-partition test case generation.” IEEE Transactions on Software Engineering, vol. 29, no.7, pp. 577−593, July 2003.

Thank you
for your attention!!