VCC: Contract-based Modular Verification of Concurrent C

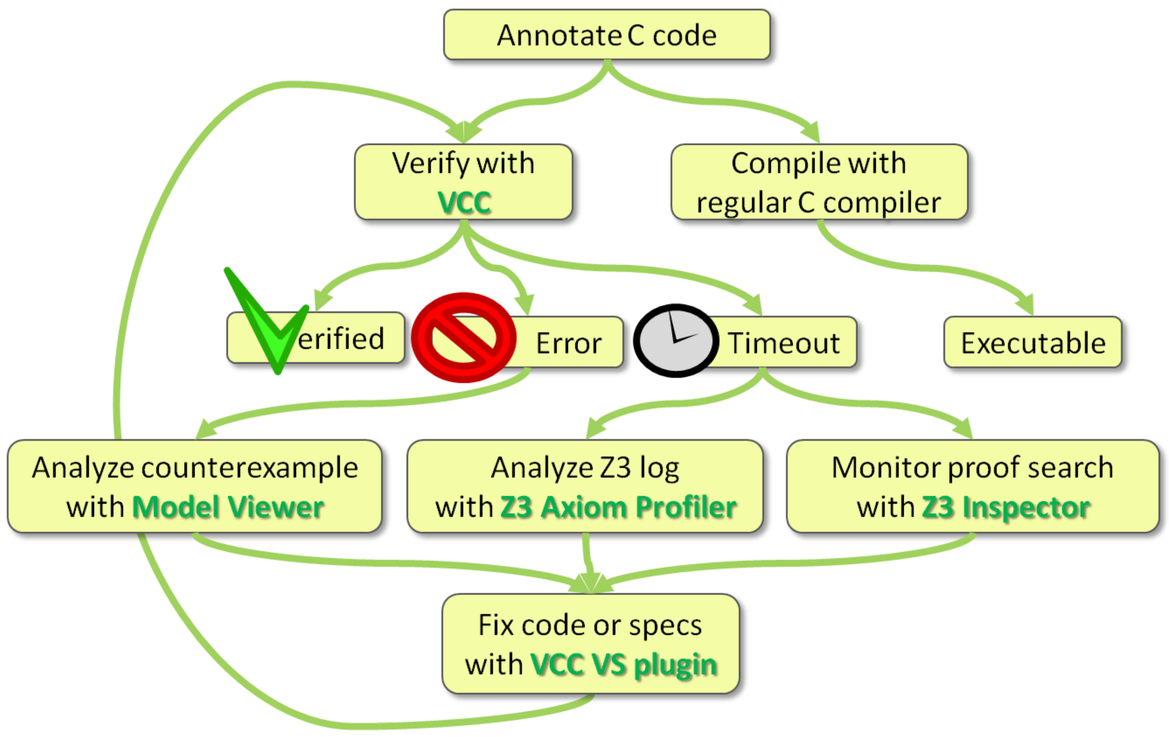
Markus Dahlweid, Michał Moskal, Thomas Santen, Stephan Tobies
European Microsoft Innovation Center
Aachen, Germany

Wolfram Schulte
Microsoft Research, RiSE
Redmond, WA, USA

Features:

- **Correctness**: Verified programs never go wrong (i.e., violate their specification)
- **Concurrency**: Operating systems stopped being single-threaded 20 years ago
- **Modularity**: Functions and data structures as natural abstraction boundaries
- **Low-level C**: Bit fields, unions, machine arithmetic, lock-free algorithms, ...
- **Inline contracts**: Specifications live and evolve along their code
- **Tool integration**: Plug into existing Microsoft developer tools
- **Application**: Used to verify Microsoft **Hyper-V**'s (virtualization) kernel

Workflow:



Verification Condition-based Verification with Boogie and Z3:

```
#include <vcc2.h>

typedef struct _BITMAP {
  UINT32 Size;      // Number of bits …
  PUINT32 Buffer;   // Memory to store …

  // private invariants
  invariant(Size > 0 && Size <= …
  …
```

Annotated C

VCC

```
$ref_cnt      (old($s), #p) == $ref_cnt
($s, #p) &&  $ite.bool($set_in(#p, $owns
(old($s), owner)),
   $ite.bool($set_in(#p, owns),
   $st_eq(old($s), $s, #p),
   $wrapped($s, #p, $typ(#p)) &&
   $timestamp_is_now($s, #p)),
$ite.bool($set_in(#p, owns),
$owner($s, #p) == owner && $closed($s,
```

Boogie

Boogie

Verification Condition Generator
Minimal imperative control flow and types on top of first order logic
Microsoft Research (R. Leino)

Chalice
Spec#
HAVOC
Simplify
Isabelle/HOL
SMTLIB

```
(FORALL (v    lv x lxv w a b)
 (QID bv:e:c4)
 (PATS
   ($bv_extract
    ($bv_concat
    ($bv_extract v lv x lv) lxv w x)
   lv a b))
    (IMPLIES
      (AND
```

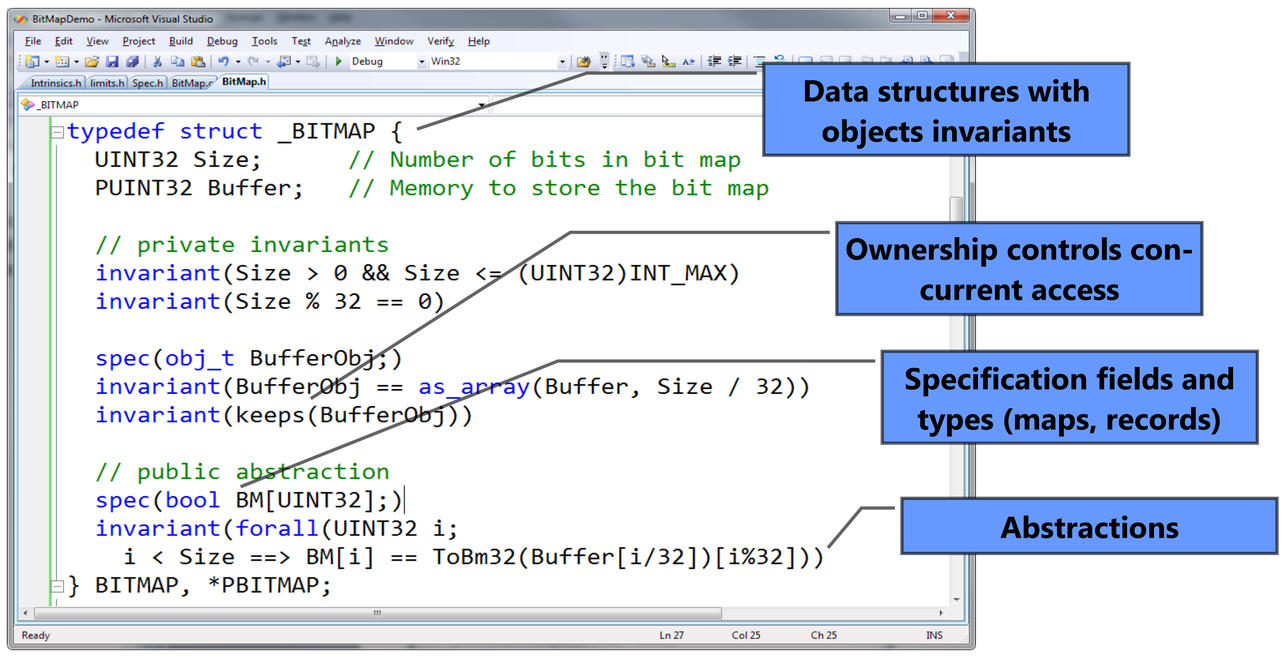
FOL

Z3

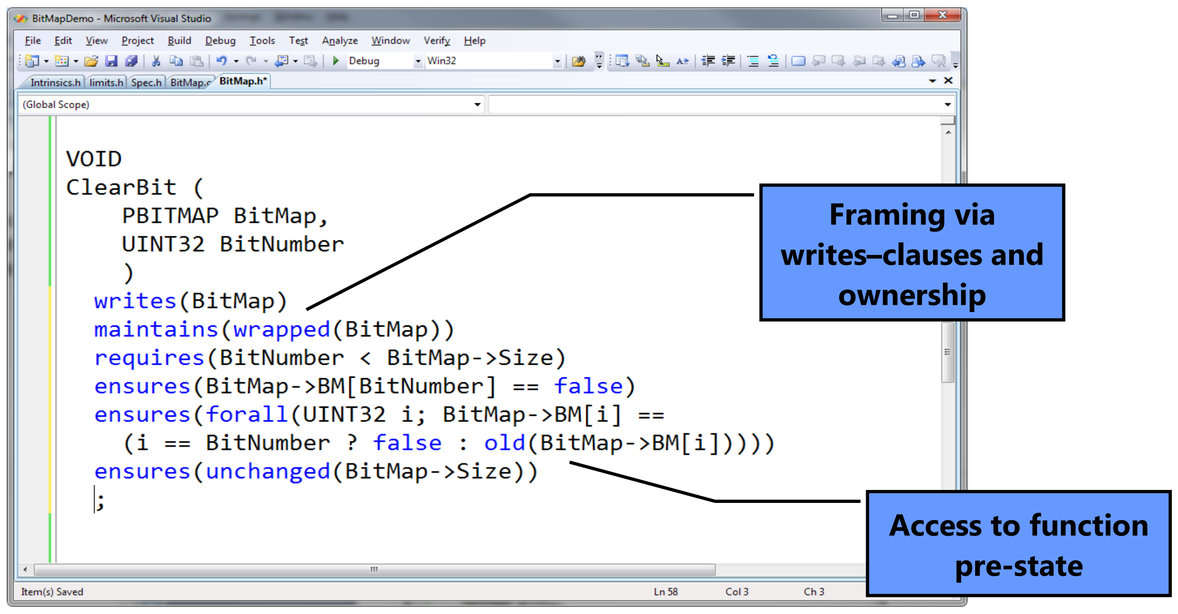
State-of-the-art automatic SMT Solver
SMT (Satisfiability Modulo Theories):
- Integer and fixed-length bitvector arithmetic
- Arrays, algebraic data types

Microsoft Research (L. de Moura, N. Bjørner)
E.g., also used by MSR's Pex tool

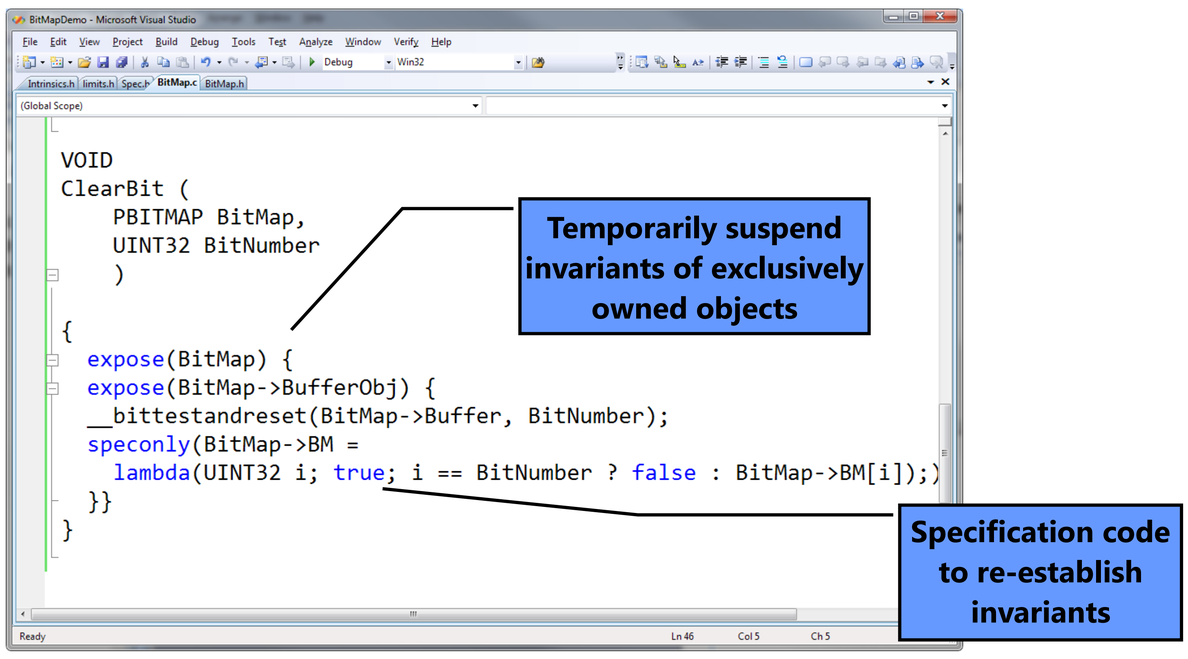
Data Structure Invariants:



Function Contracts with Pre– and Postconditions:



Specification Code to (Re-)Establish Invariants:



Integrated Error Reporting and Analysis: