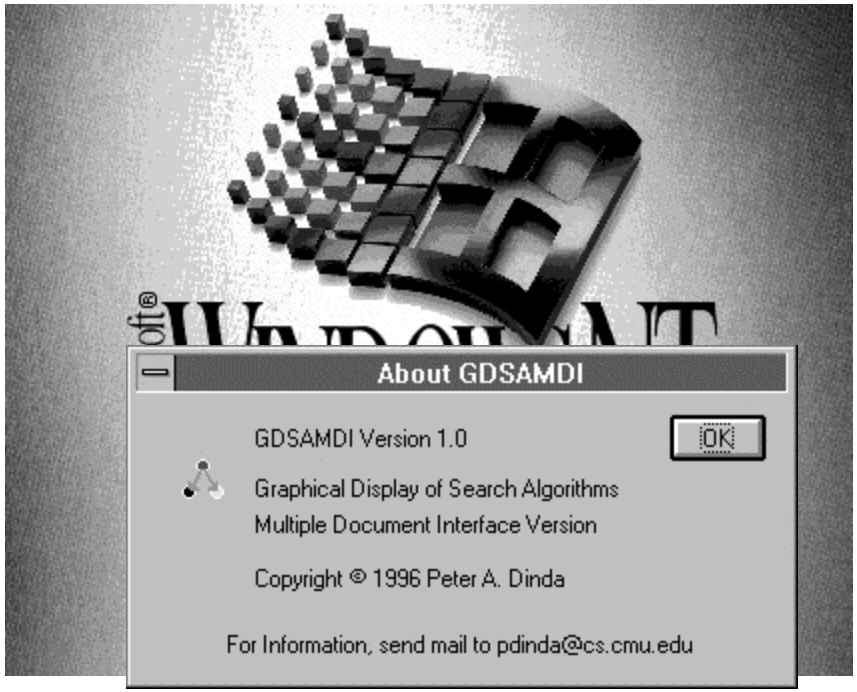
A Framework for the Graphical Display
of Search Algorithms in Microsoft Windows

GDSAMDI Framework Overview

Peter A. Dinda

CMU SCS



GDSAMDI?!? What a Lousy Name!

Yes, it is.

Graphical Display of Search Algorithms,
Multiple Document Interface

Please suggest a better one

Other horrendous, unrelated names I have chosen recently: AABLK, AAPAIR, DMRLE, DMRLEC, ART. I am a hopelessly inept namer.

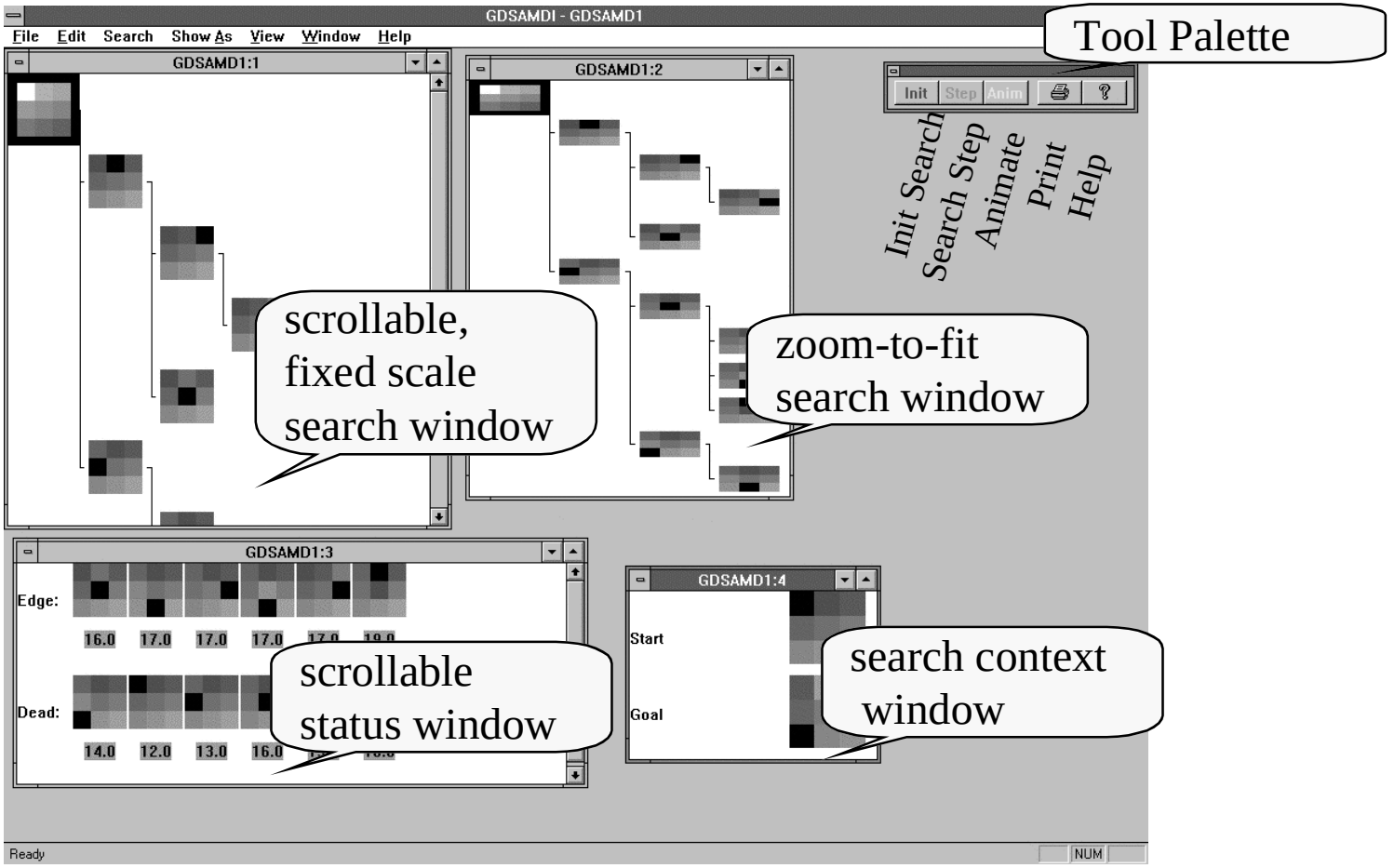
Overview of Talk

- **GDSAMDI in one slide**
- **The user interface**
- **Video taped demo**
- **Building on the framework**
- **Implementation**

GDSAMDI in One Slide

- **Cool Animated Searches**
  - Theoretically educational too...
- **An orthogonally extensible framework for animating the application of search algorithms to different state spaces and with different search contexts.**

The User Interface



Building on the Framework

- **Orthogonal extensibility along 2.5 dimensions via C++ inheritance**
- **New State Spaces**
  - Derived from class State
- **New Search Algorithms**
  - Derived from class SearchAlgorithm
- **New Search Contexts**
  - Derived from class Context

Adding New State Spaces

Derivation

```
class NPuzzleState : public State {
private:
   // NPuzzleState-specific Stuff
public:
   // NPuzzleState-specific Stuff
   CompareResult Compare(State *s);
   void GenerateChildren();
   void DrawSelf(CDC *dc, CRect *rect);
};
```

Required Overrides

Optional Override

Adding New Search Contexts

Derivation

```
class NPuzzleContext : public Context {
private:
   // NPuzzleContext specific stuff
public:
   // NPuzzleContext specific
   State *GetStartState();
   int    IsGoalState(State *s);
   void DrawContext(CDC *cdc);
   void GetContextRect(CRect *rect);
   double Heuristic(State *s);
```

Required Overrides

Optional Overrides

Adding New Search Algorithms

Derivation

```
class ASTAR : public SearchAlgorithm {
private:
   // ASTAR specific stuff
public:
    // ASTAR specific stuff
   void   Initialize(Context *c);
   State  *Step();
   SearchStatus GetStatus();
   void   DrawStatus(CDC *cdc);
   void   GetStatusRect(CRect *rect);
```

Required Overrides

Optional Overrides

Implementation Details

- **Entirely implemented within the MFC 4.0 framework, no Win32isms**
  - Standard MDI document/view application
- **32 bit Windows assumed**
  - Tested on Windows NT Workstation 3.51
  - Should work fine on Windows 95 and Windows 3.1 with Win32s
- **Should be portable to Mac and Motif using third party MFC ports**

Conclusion

Orthogonally extensible framework for animating the application of search algorithms to different state spaces and with different search contexts

A cool program with a lousy name...