Locus System

```
#pragma @Locus loop = matmul
  for (i=0; i<M; i++)
    for (j=0; j<N; j++)
      for (k=0; k<K; k++)
      C[i][j] = beta*C[i][j]
                + alpha*A[i][k]*B[k][j];
```

Annotated Source Code

```
CodeReg matmul {
  tiledim = 4;
  tiletype = Tiling2D() OR Tiling3D();
  printstatus(tiletype);
  if (tiletype == "2D") {
    RoseLocus.Unroll(loop=innermost, factor=tiledim);
  }
}
```

Locus Program

- Optimizations are target-specific and region-specific
- Separated from the application's code

Locus System

Annotated Source Code

```
#pragma @Locus loop = matmul
  for (i=0; i<M; i++)
    for (j=0; j<N; j++)
      for (k=0; k<K; k++)
      C[i][j] = beta*C[i][j]
                + alpha*A[i][k]*B[k][j];
```

Locus Program

```
CodeReg matmul {
  tiledim = 4;
  tiletype = Tiling2D() OR Tiling3D();
  printstatus(tiletype);
  if (tiletype == "2D") {
    RoseLocus.Unroll(loop=innermost, factor=tiledim);
  }
}
```

- Optimizations are target-specific and region-specific
- Separated from the application's code

Locus Optimization Language