

Enhanced Supervisor Agent with Browser Integration

Complete Setup and User Guide

Overview

The Enhanced Supervisor Agent is a sophisticated AI monitoring system that prevents browser-based AI agents from derailing or switching tasks due to contextual keywords. It specifically addresses problems like:

Problem Example:

- **User:** "I'm building a social media app for this hackathon"
- **Agent (Wrong):** "Let me help you plan the hackathon event..."
- **Supervisor:** 🚨 "Agent is focusing on 'hackathon' instead of 'social media app'"

Key Features

✅ Task Coherence Protection

- **Context Keyword Detection:** Identifies when agents focus on contextual mentions instead of main goals
- **Real-time Intervention:** Automatically corrects agents when they drift from tasks
- **Intelligent Analysis:** Distinguishes between main objectives and contextual information

✓ Browser Integration

- **Universal Compatibility:** Works with MiniMax, ChatGPT, Claude, Gemini, and other browser-based AI platforms
- **Real-time Monitoring:** Captures user inputs and agent responses in real-time
- **Seamless Installation:** Browser extension with one-click setup

✓ Advanced Supervision

- **Multi-Agent Support:** Monitors both traditional MCP agents and browser-based agents
- **Unified Dashboard:** Single interface for all supervision activities
- **Comprehensive Reporting:** Detailed analytics and intervention logs

Quick Start Guide

1. Server Setup

```
# Clone the repository
git clone <repository-url>
cd supervisor-mcp-agent

# Install dependencies
uv sync
# OR
pip install -r requirements.txt

# Start the enhanced supervisor server
python supervisor_mcp_server/server_main.py
```

Server will start at:

- Browser WebSocket: `ws://localhost:8765`
- MCP Server: `localhost:8766`

2. Browser Extension Installation

1. Load Extension in Chrome/Edge:

- Open `chrome://extensions/`
- Enable "Developer mode"
- Click "Load unpacked"
- Select the `browser_extension` folder

2. Configure Connection:

- Click the extension icon
- Verify WebSocket connection to `ws://localhost:8765`
- Set your task coherence threshold (default: 0.6)

3. Usage Example

1. Set Your Task Context:

User: "I'm building a social media app for this hackathon"


2. Monitor for Drift:


If the AI agent responds with:

Agent: "Let me help you organize this hackathon event..."

3. Automatic Intervention:

\\ \

 Supervisor Alert: Agent is focusing too much on 'hackathon' instead of your main task: 'social media app'

 Suggested correction: "Please refocus on the social media app development. I mentioned 'hackathon' as context, not as the main focus."

\\ \

Detailed Configuration

Server Configuration

Create `supervisor_config.json`:

```
{
  "browser": {
    "websocket_host": "localhost",
    "websocket_port": 8765,
    "enable_browser_monitoring": true,
    "task_coherence_threshold": 0.6,
    "auto_intervention": true,
    "max_interventions_per_session": 10,
    "enable_user_notifications": true,
    "enable_proactive_suggestions": true
  },
  "mcp": {
    "port": 8766,
    "enable_server": true
  },
  "logging": {
    "level": "INFO"
  }
}
```

Browser Extension Settings

Intervention Sensitivity:

- **Low (0.1-0.3):** Only intervene on severe drift
- **Moderate (0.4-0.7):** Balance between accuracy and intervention
- **High (0.8-1.0):** Intervene on minor drift (may have false positives)

Advanced Options:

- **Auto-suggest corrections:** Provides ready-to-use correction prompts

- **Real-time notifications:** Shows drift alerts as they happen
- **Session logging:** Saves all interactions for analysis

API Documentation

WebSocket API

Authentication

```
{  
  "type": "AUTH_REQUEST",  
  "extension_id": "your-extension-id",  
  "challenge": "random-string",  
  "signature": "hmac-signature"  
}
```

User Input Analysis

```
{  
  "type": "USER_INPUT_ANALYSIS",  
  "tab_id": "tab-123",  
  "data": {  
    "input": "I'm building a social media app for this hackathon",  
    "url": "https://chat.example.com",  
    "timestamp": 1701234567890  
  }  
}
```

Agent Message Analysis

```
{
  "type": "AGENT_MESSAGE_ANALYSIS",
  "tab_id": "tab-123",
  "data": {
    "content": "Agent response text...",
    "platform": "minimax",
    "user_input": "Previous user input",
    "timestamp": 1701234567890
  }
}
```

Response Format

```
{
  "status": "success",
  "coherence_analysis": {
    "final_score": 0.3,
    "needs_intervention": true,
    "severity": "CRITICAL",
    "issues": [
      "Keyword hijacking detected: hackathon",
      "No mention of primary goal: social media"
    ]
  },
  "intervention": {
    "type": "CRITICAL_CORRECTION",
    "message": "🚨 Agent is focusing too much on 'hackathon'...",
    "suggested_prompt": "Please refocus on the social media app development..."
  }
}
```

REST API Endpoints

Get Session Statistics

```
GET /api/sessions/{tab_id}/stats
```

Response:

```
{
  "tab_id": "tab-123",
  "platform": "minimax",
  "message_count": 15,
  "intervention_count": 2,
  "duration_minutes": 25,
  "task_context": {
    "primary_goal": "social media app",
    "domain": "software_development"
  }
}
```

Get Comprehensive Report

```
GET /api/reports/comprehensive
```

Troubleshooting

Common Issues

1. Extension Not Connecting

Symptom: "Connecting to supervisor server..." never completes

Solution:

- Verify server is running on localhost:8765
- Check firewall settings
- Try reloading the extension

2. No AI Agent Detected

Symptom: "No AI agent detected" in popup

Solution:

- Refresh the AI chat page
- Verify the AI platform is supported
- Check browser console for errors

3. Interventions Not Working

Symptom: Agent keeps derailing without intervention

Solution:

- Lower the coherence threshold (try 0.4)
- Enable "Auto-suggest corrections"
- Check if task context was properly set

4. Too Many False Positives

Symptom: Constant intervention alerts

Solution:

- Raise the coherence threshold (try 0.8)
- Check task context accuracy
- Review intervention logs

Debug Mode

Enable debug logging:

```
python supervisor_mcp_server/server_main.py --log-level DEBUG
```

Browser extension debug:

1. Open Chrome DevTools
2. Go to Extensions tab
3. Find "AI Agent Supervisor"
4. Click "background page" or "service worker"
5. Check console logs

Performance Optimization

For High-Volume Usage:

```
{
  "browser": {
    "max_interventions_per_session": 5,
    "enable_proactive_suggestions": false,
    "task_coherence_threshold": 0.7
  }
}
```

For Sensitive Tasks:

```
{
  "browser": {
    "task_coherence_threshold": 0.4,
    "auto_intervention": true,
    "enable_user_notifications": true
  }
}
```

Advanced Features

Custom Task Context

Manually set task context via popup:

1. Click extension icon
2. Click "Set Task Context"
3. Enter your main goal: "Build a social media app"
4. Click "Save"

Intervention History

View intervention logs:

1. Click extension icon
2. Click "View Activity Log"
3. Review recent interventions and their effectiveness

Integration with Existing Systems

```
# Python integration example
from supervisor_mcp_server.src.enhanced_supervisor_agent import
EnhancedSupervisorAgent

async def integrate_with_existing_system():
    config = {
        'browser': {
            'websocket_host': 'localhost',
            'websocket_port': 8765,
            'enable_browser_monitoring': True
        }
    }

    supervisor = EnhancedSupervisorAgent(config)
    await supervisor.start_browser_monitoring()

    # Monitor browser agent
    report = await supervisor.supervise_agent('tab-123',
agent_type='browser')
    print(f"Supervision report: {report}")

    # Get comprehensive report
    full_report = await supervisor.get_comprehensive_report()
    print(f"Active sessions: {full_report['browser_monitoring']
['active_sessions']}")
```

Security Considerations

Data Privacy

- **Local Processing:** All analysis happens locally on your machine

- **No Data Transmission:** Conversations are not sent to external servers
- **Secure WebSocket:** Uses secure WebSocket connections with HMAC authentication

Network Security

- **Localhost Only:** Default configuration only accepts local connections
- **Rate Limiting:** Built-in protection against abuse
- **Authentication:** Extension must authenticate before sending data

Production Deployment

For production environments:

```
{
  "browser": {
    "websocket_host": "0.0.0.0",
    "websocket_port": 8765,
    "enable_rate_limiting": true,
    "max_connections_per_ip": 10,
    "require_authentication": true
  }
}
```

Support and Development

Getting Help

- **Documentation:** Check this guide and API documentation
- **Logs:** Enable debug mode and check logs
- **Issues:** Report bugs with detailed reproduction steps

Contributing

1. Fork the repository
2. Create a feature branch
3. Add tests for new functionality
4. Submit a pull request

Extension Development

To modify the browser extension:

1. Edit files in `browser_extension/`
2. Reload extension in Chrome
3. Test with different AI platforms
4. Update manifest version

Adding New AI Platform Support

To add support for a new AI platform:

```
// In content.js, add new platform detection
const aiPlatforms = [
  // ... existing platforms
  {
    name: 'NewAI',
    selector: '.new-ai-input, [data-ai="input"]'
  }
];
```

License and Acknowledgments

This Enhanced Supervisor Agent builds upon the original Supervisor Agent architecture and extends it with browser integration and task coherence protection capabilities.

Ready to get started? Follow the Quick Start Guide above and begin protecting your AI interactions from task derailment!