Healthcare Patient Care Coordination System

Complete Microsoft Copilot Studio Case Study

Overview

This case study demonstrates building a comprehensive healthcare patient care coordination system using Microsoft Copilot Studio. The system includes multiple specialized agents working together to streamline patient intake, medical record management, appointment scheduling, and care coordination.

Business Scenario

Mercy General Hospital needs to modernize their patient care coordination process. Currently, patients experience long wait times, fragmented communication between departments, and delayed access to medical records. The hospital wants to create an intelligent system that can handle patient inquiries 24/7, coordinate care between departments, and provide seamless access to medical information while maintaining HIPAA compliance.

Architecture Overview

Multi-Agent System Design

- **Primary Care Agent**: Main patient interaction point
- **Scheduling Agent**: Appointment management and coordination
- **Medical Records Agent**: Secure access to patient records
- **Billing Agent**: Insurance and billing inquiries
- **Emergency Triage Agent**: Critical patient assessment

Prerequisites and Setup

Licensing Requirements

- Microsoft Copilot Studio tenant license
- Copilot Studio user licenses for administrators
- Power Platform environment (Production + Development)
- Microsoft 365 Copilot licenses (optional for enhanced features)
- Azure subscription for external integrations

Technical Prerequisites

- Power Platform Admin Center access
- Azure Active Directory tenant
- SharePoint Online for knowledge bases
- Microsoft Dataverse for patient data storage
- Power Automate Premium for complex workflows

Data Sources Setup

1. **Patient Management System (EHR) Integration**
   - Custom connector to Epic/Cerner systems
   - FHIR API endpoints for patient data
   - HL7 message processing
2. **Appointment System Integration**
   - Scheduling database connection
   - Calendar synchronization with Outlook
   - Provider availability systems
3. **Billing System Integration**
   - Insurance verification APIs
   - Claims processing databases
   - Payment processing systems

Detailed Implementation Guide

Phase 1: Primary Care Agent Creation

Agent Configuration

```
Agent Name: Mercy Healthcare Assistant
Description: Comprehensive patient care coordination agent providing 24/7 support for pat

Instructions:
You are a healthcare assistant for Mercy General Hospital. You help patients with:
- General health inquiries and symptom assessment
- Appointment scheduling and management
- Medical record access and explanations
- Insurance and billing questions
- Emergency triage and routing

Always maintain a professional, empathetic tone. Follow HIPAA guidelines strictly. Never

Conversation Style: Professional, empathetic, healthcare-focused
```

Core Topics Development

Topic 1: Patient Registration and Intake

- **Trigger Phrases**: "register as new patient", "first time here", "new patient intake"
- **Flow Description**: Collects patient demographics, insurance information, medical history
- **Nodes**:
  - Welcome message with HIPAA disclosure
  - Patient information collection (Name, DOB, Address, Phone)
  - Insurance verification
  - Medical history questionnaire
  - Primary care provider assignment
  - Registration confirmation

Topic 2: Appointment Scheduling

- **Trigger Phrases**: "schedule appointment", "book visit", "see doctor"
- **Flow Description**: Intelligent appointment booking with provider matching
- **Nodes**:
  - Appointment type identification
  - Provider specialty matching
  - Date/time preference collection
  - Insurance verification
  - Appointment confirmation
  - Calendar integration

Topic 3: Symptom Assessment and Triage

- **Trigger Phrases**: "not feeling well", "symptoms", "medical concern"
- **Flow Description**: Basic symptom collection and urgency assessment
- **Nodes**:
  - Symptom description collection
  - Severity assessment questionnaire
  - Urgency scoring algorithm
  - Care pathway recommendation
  - Emergency routing (if critical)

Generative AI Configuration

Enable **Generative Orchestration** to allow the agent to:

- Dynamically select appropriate topics based on patient needs
- Fill in missing information through conversational prompts
- Chain multiple actions for complex patient requests
- Provide personalized responses based on patient history

Phase 2: Agent Flows for Complex Workflows

Agent Flow 1: Patient Record Retrieval

```
Flow Name: Secure Patient Record Access
Trigger: When an agent calls the flow
Inputs:
- Patient ID (Text)
- Healthcare Provider ID (Text)
- Record Type Requested (Choice)

Actions:
1. Authenticate Healthcare Provider
2. Verify Patient Consent
3. Retrieve Medical Records from EHR
4. Apply Data Filtering (HIPAA Compliance)
5. Format Response for Agent Consumption

Outputs:
- Patient Summary (Text)
- Recent Visits (Text)
- Current Medications (Text)
- Allergies and Conditions (Text)
```

Agent Flow 2: Multi-Department Appointment Coordination

```
Flow Name: Complex Appointment Scheduling
Trigger: When an agent calls the flow
Inputs:
- Patient ID (Text)
- Required Specialties (Text Array)
- Preferred Date Range (Date)
- Insurance Type (Text)

Actions:
1. Query Provider Availability
2. Check Insurance Coverage
3. Coordinate Multiple Appointments
4. Send Confirmation Notifications
5. Update Patient Calendar
6. Generate Appointment Packets
```

```
Outputs:
- Appointment Confirmations (Text)
- Total Cost Estimate (Number)
- Preparation Instructions (Text)
```

Phase 3: Tools and Connectors

Custom Connector: EHR Integration

```
Connector Name: Epic EHR Connector
Base URL: https://api.epic.example.com/fhir/
Authentication: OAuth 2.0

Actions:
1. Get Patient Demographics
   - Endpoint: /Patient/{id}
   - Method: GET
   - Returns: Patient demographic information

2. Get Patient Observations
   - Endpoint: /Observation?patient={id}
   - Method: GET
   - Returns: Lab results, vital signs

3. Create Appointment
   - Endpoint: /Appointment
   - Method: POST
   - Body: Appointment details JSON

4. Send Secure Message
   - Endpoint: /Communication
   - Method: POST
   - Body: Message content with patient reference
```

Power Automate Tools Integration

- **Email Notifications**: Appointment confirmations and reminders
- **SMS Alerts**: Critical appointment changes and health alerts
- **Calendar Sync**: Provider and patient calendar integration
- **Document Generation**: Consent forms and patient instructions

Phase 4: Multi-Agent Orchestration

Agent Hierarchy Design

```
Master Agent: Mercy Healthcare Assistant
├── Scheduling Specialist Agent
│   ├── Provider availability checking
│   ├── Insurance verification
│   └── Appointment confirmation
├── Medical Records Agent
│   ├── Secure record retrieval
│   ├── Record interpretation
│   └── Privacy compliance
├── Billing Support Agent
│   ├── Insurance claims processing
│   ├── Payment plan setup
│   └── Financial assistance programs
└── Emergency Triage Agent
    ├── Symptom severity assessment
    ├── Emergency routing
    └── Critical alert system
```

Cross-Agent Communication

- **Shared Variables**: Patient context maintained across agents
- **Agent Handoffs**: Seamless transitions between specialized agents
- **Context Preservation**: Medical history and current session data

Phase 5: Model Context Protocol (MCP) Integration

MCP Server Setup

```
MCP Server: Mercy Medical Knowledge Server
Capabilities:
- Medical knowledge base access
- Drug interaction checking
- Clinical decision support
- Medical coding assistance

Tools Exposed:
1. Medical Knowledge Query
   - Input: Symptom description
   - Output: Relevant medical information
   - Usage: Help agents provide accurate health information

2. Drug Interaction Checker
   - Input: List of medications
   - Output: Interaction warnings
   - Usage: Patient safety during medication review

3. ICD-10 Code Lookup
   - Input: Diagnosis description
   - Output: Medical codes
   - Usage: Billing and documentation support
```

Phase 6: Autonomous Agent Capabilities

Autonomous Monitoring Agent

```
Agent Name: Patient Care Monitor
Purpose: Proactively monitor patient health metrics and care gaps

Autonomous Triggers:
1. Lab Result Alerts
   - Trigger: New critical lab results
   - Action: Notify care team and patient
   - Escalation: Page on-call physician if critical

2. Appointment Reminders
   - Trigger: 24 hours before appointment
   - Action: Send reminder with prep instructions
   - Follow-up: Reschedule if no confirmation

3. Medication Refill Reminders
   - Trigger: 7 days before medication expires
   - Action: Contact pharmacy for refill
   - Alert: Notify patient of refill status

4. Care Gap Analysis
   - Trigger: Monthly review cycle
   - Action: Identify overdue preventive care
   - Notification: Alert primary care provider
```

Sample Data Setup

Patient Test Data (Dataverse Tables)

```
-- Patients Table
PatientID | FirstName | LastName | DateOfBirth | Phone | Email | InsuranceID
PAT001 | John | Smith | 1980-05-15 | 555-0101 | john.smith@email.com | INS001
PAT002 | Sarah | Johnson | 1975-08-22 | 555-0102 | sarah.j@email.com | INS002
PAT003 | Michael | Brown | 1990-12-03 | 555-0103 | m.brown@email.com | INS003

-- Appointments Table
AppointmentID | PatientID | ProviderID | AppointmentDate | Type | Status
APT001 | PAT001 | DOC001 | 2025-08-15 10:00 | Routine Checkup | Scheduled
APT002 | PAT002 | DOC002 | 2025-08-16 14:30 | Specialist | Confirmed
APT003 | PAT003 | DOC001 | 2025-08-17 09:00 | Follow-up | Pending

-- Providers Table
ProviderID | Name | Specialty | Phone | Email | Schedule
DOC001 | Dr. Emily Chen | Family Medicine | 555-0201 | dr.chen@mercy.com | Mon-Fri 8-5
```

```
DOC002 | Dr. Robert Wilson | Cardiology | 555-0202 | dr.wilson@mercy.com | Mon-Wed-Fri 9-
DOC003 | Dr. Lisa Rodriguez | Pediatrics | 555-0203 | dr.rodriguez@mercy.com | Tue-Thu 8-
```

Knowledge Base Content (SharePoint)

Create the following document libraries in SharePoint:

- **Medical Procedures**: Step-by-step procedure guides
- **Patient Education**: Health information and care instructions
- **Insurance Policies**: Coverage information and claims processes
- **Emergency Protocols**: Critical care pathways and escalation procedures

Testing and Validation

Test Scenarios

Scenario 1: New Patient Registration

1. User: "I'm a new patient and want to schedule an appointment"
2. Expected Flow: Registration → Insurance verification → Provider matching → Appointment scheduling
3. Validation Points:
   - Patient data collected and stored securely
   - Insurance verified through API
   - Appropriate provider assigned
   - Confirmation sent via multiple channels

Scenario 2: Emergency Triage

1. User: "I'm having chest pain and shortness of breath"
2. Expected Flow: Immediate triage → Emergency routing → Provider notification
3. Validation Points:
   - High-priority symptoms recognized
   - Emergency protocols activated
   - Healthcare providers alerted
   - Patient given immediate guidance

Scenario 3: Multi-Department Coordination

1. User: "I need to see a cardiologist and nutritionist for my diabetes"
2. Expected Flow: Multiple agent coordination → Appointment scheduling → Care plan creation
3. Validation Points:
    - Multiple specialists coordinated
    - Appointments scheduled efficiently
    - Care team communication established
    - Patient receives comprehensive care plan

Performance Metrics

- Response Time: < 2 seconds for standard queries
- Accuracy Rate: > 95% for information retrieval
- Patient Satisfaction: > 4.5/5 rating
- HIPAA Compliance: 100% adherence to privacy standards

Security and Compliance

HIPAA Compliance Measures

- **Data Encryption**: All patient data encrypted at rest and in transit
- **Access Controls**: Role-based permissions for healthcare staff
- **Audit Logging**: Complete audit trail of all patient data access
- **Business Associate Agreements**: Proper agreements with all vendors

Security Implementation

- **Authentication**: Multi-factor authentication for all users
- **Authorization**: Granular permissions based on job role
- **Data Loss Prevention**: DLP policies preventing unauthorized data sharing
- **Network Security**: VPN access for external integrations

Deployment Guide

Environment Setup

1. **Development Environment**
    - Agent development and testing
    - Sandbox data for safe testing

   - Development team access

2. **Test Environment**
   - User acceptance testing
   - Integration testing
   - Performance validation
3. **Production Environment**
   - Live patient interactions
   - Real medical data
   - High availability setup

Go-Live Checklist

- [ ] All agents tested and validated
- [ ] Security and compliance review completed
- [ ] Healthcare staff training completed
- [ ] Monitoring and alerting configured
- [ ] Backup and disaster recovery tested
- [ ] User documentation provided
- [ ] Support procedures established

Success Metrics and ROI

Key Performance Indicators

- **Patient Satisfaction**: 40% improvement in patient satisfaction scores
- **Response Time**: 60% reduction in average response time
- **Staff Efficiency**: 35% reduction in administrative tasks
- **Appointment No-Shows**: 25% reduction through better reminders
- **Cost Savings**: $500,000 annual savings in administrative costs

Return on Investment

- **Implementation Cost**: $150,000
- **Annual Savings**: $500,000
- **ROI**: 233% first-year return
- **Break-even**: 3.6 months

This healthcare case study demonstrates the full power of Microsoft Copilot Studio in creating a comprehensive, compliant, and efficient patient care coordination system that improves patient outcomes while reducing operational costs.